

CloudFront & Route 53-Task

1. Configure VPC peering in cross regions.

- VPC Peering allows **two VPCs to privately communicate** using AWS backbone network(not public internet).It works like a **private tunnel between two VPCs**.
- **Example:** - VPC-A → 10.0.0.0/16, VPC-B → 192.168.0.0/16
- We want EC2 in VPC-A to talk to EC2 in VPC-B.

Pre-Requisites:

- CIDR blocks must **NOT overlap**
- VPCs can be same or different AWS accounts
- Same region or different regions allowed

Step 1:

Go to AWS console and search for VPC

Click on create VPC

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with options like 'Virtual private cloud' (selected), 'Your VPCs', 'Subnets', 'Route tables', etc. In the center, a green banner says 'You successfully created vpc-063deb11e5d2636a0 / Vpc-peerA'. Below it, the main panel shows the details for 'vpc-063deb11e5d2636a0 / Vpc-peerA'. The 'Details' section includes fields for VPC ID (vpc-063deb11e5d2636a0), State (Available), Block Public Access (Off), DNS resolution (Enabled), Main network ACL (acl-0ecd55a9634d9883b), Default VPC (No), IPv4 CIDR (10.0.0.0/16), Network Address Usage metrics (Disabled), Encryption control ID (None), and Route 53 Resolver DNS Firewall rule groups (None). At the bottom, there are tabs for 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'.

- Click on subnets and create a subnet with CIDR range.

CloudFront & Route 53-Task

The screenshot shows the AWS VPC Subnets console. A green success message at the top states: "You have successfully created 1 subnet: subnet-0cd21b94ac78a9e66". The main table lists one subnet: "Sub_pub" with Subnet ID "subnet-0cd21b94ac78a9e66", State "Available", and VPC "vpc-08aced393ce2cf636 | Vpc-peerOhio". The sidebar on the left shows the "Subnets" section under "Virtual private cloud". The bottom navigation bar includes links for CloudShell, Feedback, and Console Mobile App.

- Click route tables
- Edit routes
- Add subnet associations to route table

The screenshot shows the AWS VPC Route Tables console for route table "rtb-0e3959c98b0837f11 / RT_ohio_peer". A green success message at the top states: "You have successfully updated subnet associations for rtb-0e3959c98b0837f11 / RT_ohio_peer". The "Details" tab shows the Route table ID "rtb-0e3959c98b0837f11", Main status (unchecked), Owner ID "814588432081", and Explicit subnet associations "subnet-0cd21b94ac78a9e66 / Sub_pub". The "Routes" tab shows one route entry: Destination "0.0.0.0/0", Target "rtb-0e3959c98b0837f11", Status "Active", Propagated, and Route Origin "Peer". The bottom navigation bar includes links for CloudShell, Feedback, and Console Mobile App.

The screenshot shows the AWS VPC Route Tables console for route table "rtb-0f6881c2166b91ae0 / RT_peer_virginia". A green success message at the top states: "You have successfully updated subnet associations for rtb-0f6881c2166b91ae0 / RT_peer_virginia". The "Details" tab shows the Route table ID "rtb-0f6881c2166b91ae0", Main status (unchecked), Owner ID "814588432081", and Explicit subnet associations "subnet-0c627c14fe18fa4cf / sub_prt". The "Routes" tab shows one route entry: Destination "0.0.0.0/0", Target "rtb-0f6881c2166b91ae0", Status "Active", Propagated, and Route Origin "Peer". The bottom navigation bar includes links for CloudShell, Feedback, and Console Mobile App.

CloudFront & Route 53-Task

Step 2: Create VPC peering connection

Go to:

VPC Console → Peering Connections → Create Peering Connection→

Action send request to other region(ohio).

The screenshot shows the AWS VPC Peering Connections page. A green banner at the top indicates a new peering connection request: "A VPC peering connection pcx-0da60d4421e30a4f7 / peer ohio-virginia has been requested. Remember to change your region to us-east-2 to accept the peering connection." Below this, the peering connection details are listed:

pcx-0da60d4421e30a4f7 / peer ohio-virginia	
Details	Info
Requester owner ID 814588432081	Acceptor owner ID 814588432081
Peering connection ID pcx-0da60d4421e30a4f7	Requester VPC vpc-063deb11e5d2636a0 / Vpc-peerA
Status Initiating Request to 814588432081	Requester CIDRs 10.0.0.0/16
Expiration time Tuesday, January 27, 2026 at 15:55:43 GMT+5:30	Requester Region N. Virginia (us-east-1)
	VPC Peering connection ARN arn:aws:ec2:us-east-1:814588432081:vpc-peering-connection/pcx-0da60d4421e30a4f7
	Acceptor VPC vpc-08aced393ce2cf636
	Acceptor CIDRs -
	Acceptor Region Ohio (us-east-2)

- Go to ohio region and click on peering connection and in action accept the request .

The screenshot shows the AWS VPC Peering Connections page in the Ohio region. A green banner indicates the connection is established: "Your VPC peering connection (pcx-0da60d4421e30a4f7 / ohio-virginia) has been established. To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables." Below this, the peering connections table shows the established connection:

Peerings					
Name	Peering connection ID	Status	Requester VPC	Acceptor	Actions
pcx-0da60d4421e30a4f7 / ohio-virginia	pcx-0da60d4421e30a4f7	Established	vpc-063deb11e5d2636a0 / Vpc-peerA	vpc-08aced393ce2cf636	Actions Create peering connection

CloudFront & Route 53-Task

- In route tables edit routes and add peering connection ID of opposite region in both regions route tables should update .

The screenshot shows the AWS VPC Route Tables page. At the top, a green success message says "Updated routes for rtb-0f6881c2166b91ae0 / RT_peer_virginia successfully". Below this, the "Routes" tab is selected, showing two routes:

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table
11.0.0.0/16	pcx-0da60d4421e30a4f7	Active	No	Create Route

- Go to EC2 click on instance
- And launch one instance as shown below

The screenshot shows the AWS EC2 Instances page. A green success message at the top says "Successfully initiated launch of instance (i-0ff0868e060d60658)". Below this, there's a "Next Steps" section with a search bar and a numbered list of actions:

- 1. Create billing usage alerts
- 2. Connect to your instance
- 3. Connect an RDS database
- 4. Create EBS snapshot policy

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App.

CloudFront & Route 53-Task

- Go to ohio region and launch instance with public ip

The screenshot shows the AWS EC2 Instances page. The instance summary for i-0f195e7edb77402b6 (ec2-peer0Ohio) is displayed. Key details include:

- Instance ID: i-0f195e7edb77402b6
- Public IPv4 address: 18.224.23.20
- Private IP4 address: 11.0.229.104
- Instance state: Pending
- Public DNS: ec2-18-224-23-20.us-east-2.compute.amazonaws.com
- Private IP DNS name (IPv4 only): ip-11-0-229-104.us-east-2.compute.internal
- Instance type: t2.micro
- VPC ID: vpc-0917073257345678

- Check connectivity using ssh -i ec2-user@ip

```
root@ip-11-0-229-104:~#
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "ohio_keypair.pem" ec2-user@ec2-18-224-23-20.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-224-23-20.us-east-2.compute.amazonaws.com (18.224.23.20)' can't be established.
ED25519 key fingerprint is SHA256:MF4uX9ghkmTEAtHjCTZDQCrcQlB1rYDTa/vIEN9brw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-224-23-20.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
,
#_
~\_\_ #####_ Amazon Linux 2023
~~ \_\_#####\
~~ \_\#\#\| https://aws.amazon.com/linux/amazon-linux-2023
~~ \_\#/ \_\_ V~' '->
~~ \_\_\_. /_
~~ \_\_ /_/
~~ \_\_m/' [ec2-user@ip-11-0-229-104 ~]$ sudo su -
[root@ip-11-0-229-104 ~]# |
```

CloudFront & Route 53-Task

Error: After connecting with public we have to connect with private ip here ping is not coming to EC2 private ip .

- So that first we have to switch user to root for avoid error (extra privileges)
- And in SG we should add ICMP protocol.

```
root@ip-11-0-229-104:~  
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)  
$ ssh -i "ohio_keypair.pem" ec2-user@ec2-18-224-23-20.us-east-2.compute.amazonaws.com  
The authenticity of host 'ec2-18-224-23-20.us-east-2.compute.amazonaws.com (18.224.23.20)' can't be established.  
ED25519 key fingerprint is SHA256:MF4uX9ghkmTEAtHjCTZDQCrcQ1B1rYDTa/vIEN9brw.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-18-224-23-20.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.  
 , #  
 ~\_\_ ##### Amazon Linux 2023  
 ~\_\_ #####\|  
 ~\_\_ \|#|  
 ~\_\_ \|#/ , __ https://aws.amazon.com/linux/amazon-linux-2023  
 ~\_\_ \|/ , '->  
 ~\_\_ \|/ , /  
 ~\_\_ \|/ , /  
 ~\_\_ \|/ , /  
 [ec2-user@ip-11-0-229-104 ~]$ sudo su -  
[root@ip-11-0-229-104 ~]# ping 10.0.219.45  
PING 10.0.219.45 (10.0.219.45) 56(84) bytes of data.  
|
```

```
[root@ip-11-0-229-104 ~]# ping 10.0.219.45  
PING 10.0.219.45 (10.0.219.45) 56(84) bytes of data.  
64 bytes from 10.0.219.45: icmp_seq=160 ttl=127 time=12.3 ms  
64 bytes from 10.0.219.45: icmp_seq=161 ttl=127 time=12.1 ms  
64 bytes from 10.0.219.45: icmp_seq=162 ttl=127 time=12.3 ms  
64 bytes from 10.0.219.45: icmp_seq=163 ttl=127 time=12.2 ms  
64 bytes from 10.0.219.45: icmp_seq=164 ttl=127 time=12.1 ms  
64 bytes from 10.0.219.45: icmp_seq=165 ttl=127 time=12.7 ms  
64 bytes from 10.0.219.45: icmp_seq=166 ttl=127 time=12.2 ms  
64 bytes from 10.0.219.45: icmp_seq=167 ttl=127 time=12.1 ms  
64 bytes from 10.0.219.45: icmp_seq=168 ttl=127 time=12.2 ms  
64 bytes from 10.0.219.45: icmp_seq=169 ttl=127 time=12.5 ms  
64 bytes from 10.0.219.45: icmp_seq=170 ttl=127 time=12.8 ms  
64 bytes from 10.0.219.45: icmp_seq=171 ttl=127 time=12.1 ms  
64 bytes from 10.0.219.45: icmp_seq=172 ttl=127 time=12.2 ms  
64 bytes from 10.0.219.45: icmp_seq=173 ttl=127 time=12.8 ms  
64 bytes from 10.0.219.45: icmp_seq=174 ttl=127 time=12.5 ms  
64 bytes from 10.0.219.45: icmp_seq=175 ttl=127 time=12.5 ms  
64 bytes from 10.0.219.45: icmp_seq=176 ttl=127 time=12.3 ms  
^C  
--- 10.0.219.45 ping statistics ---  
176 packets transmitted, 17 received, 90.3409% packet loss, time 181385ms  
rtt min/avg/max/mdev = 12.093/12.355/12.828/0.234 ms  
[root@ip-11-0-229-104 ~]# |
```

- Conclusion: The above image shows ec2 in both regions are connected with private ip's

CloudFront & Route 53-Task

2. Purchase one domain from GoDaddy.

➤ Step-1: Open GoDaddy

Go to  <https://www.godaddy.com>

Click **Sign In** → Login or Create account

➤ Step-2: Search for Domain

- In search box type your domain name

Example:

- neelimaranidevops.online
- Click **Search**

It will show:

-  Available
-  Taken
-  Suggested alternatives

➤ Step-3: Add to Cart

Click **Add to Cart**

Click **Continue to Cart**

 Skip all extras:

- Domain protection
- Website builder
- Email
(you don't need them for AWS)

Click **Continue to Checkout**

➤ Step-4: Payment

Choose:

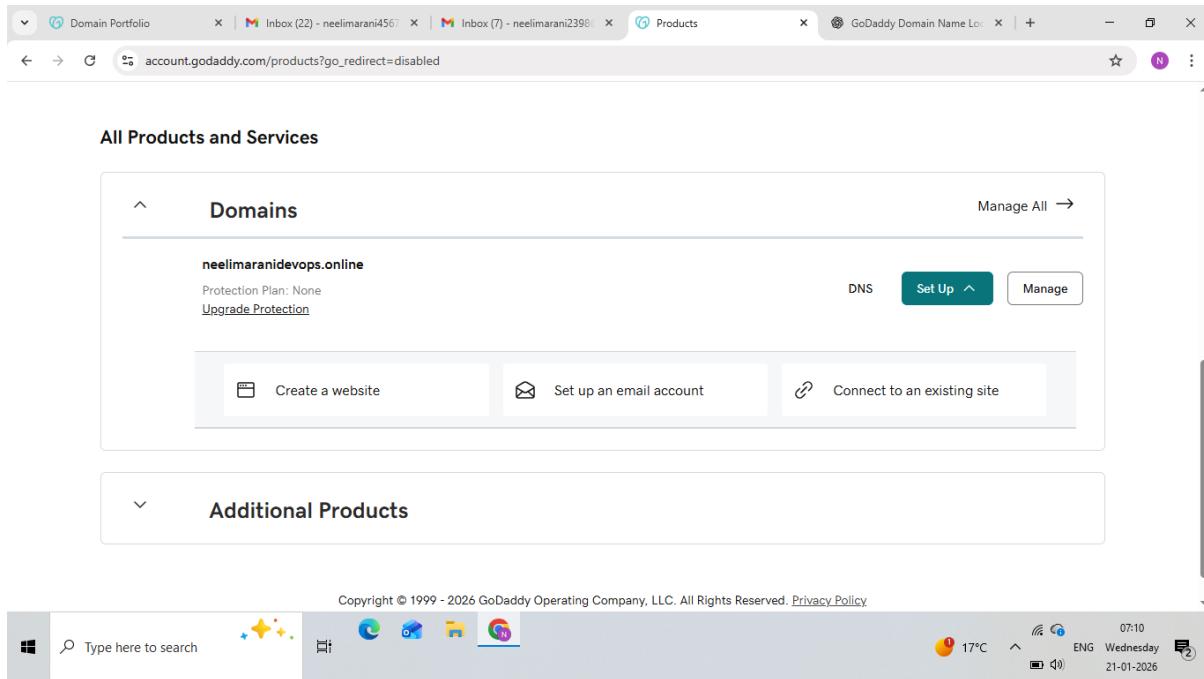
- 1 year (enough)
- Payment method (UPI / Card)

Click **Complete Purchase**

➤ Step-5: Verify Domain

- GoDaddy will send verification email
Open mail → Click **Verify Domain**
- Without this, DNS will not work 

CloudFront & Route 53-Task

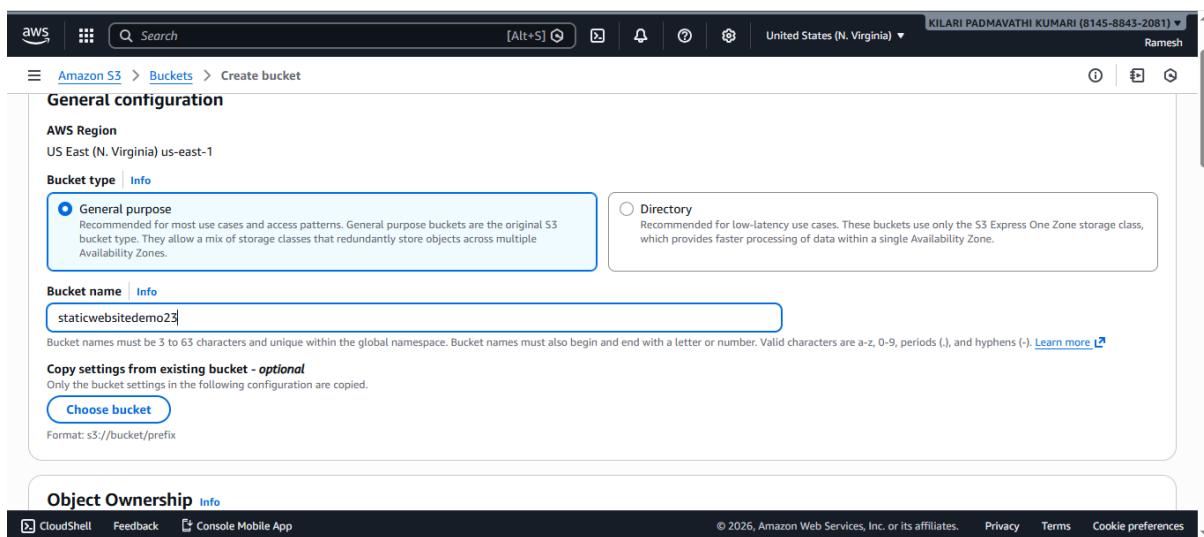


The above image shows Domain in Godaddy website in my products.

3. Deploy static website in S3.

Step-1: Create S3 Bucket

- Go to:
AWS → S3 → Create bucket
- Bucket name: staticwebsitedemo23(bucket name should be unique)



CloudFront & Route 53-Task

The screenshot shows the AWS S3 'Create bucket' interface. In the 'Block Public Access settings for this bucket' section, there is a checkbox labeled 'Block all public access'. Below it, four other options are listed under 'Block public access to buckets and objects granted through new access control lists (ACLS)':

- 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.

A warning message at the bottom states: '⚠️ Turning off block all public access might result in this bucket and the objects within becoming public'. The AWS footer includes links for CloudShell, Feedback, Console Mobile App, Privacy, Terms, and Cookie preferences.

- Block Public Access settings for this bucket uncheck it.

The screenshot shows the AWS S3 'Create bucket' interface. In the 'Bucket Versioning' section, there is a radio button labeled 'Enable' selected. A note below says: 'Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.' A 'Tags - optional' section follows, with a note: 'You can use bucket tags to analyze, manage and specify permissions for a bucket.' A callout box provides information about using s3:ListTagsForResource, s3:TagResource, and s3:UntagResource APIs. The AWS footer includes links for CloudShell, Feedback, Console Mobile App, Privacy, Terms, and Cookie preferences.

- Bucket versioning must be enable

The screenshot shows the AWS S3 'Create bucket' interface. In the 'Encryption type' section, there is a radio button labeled 'Server-side encryption with Amazon S3 managed keys (SSE-S3)' selected. A note below says: 'Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).' In the 'Bucket Key' section, there is a radio button labeled 'Enable' selected. A note below says: 'Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS.' A 'Advanced settings' section is shown with a note: 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.' The AWS footer includes links for CloudShell, Feedback, Console Mobile App, Privacy, Terms, and Cookie preferences.

- Click on create bucket.

CloudFront & Route 53-Task

The screenshot shows the AWS S3 console with the following details:

- Breadcrumbs:** Amazon S3 > Buckets
- Status Bar:** Successfully created bucket "staticwebsitedemo2398". To upload files and folders, or to configure additional bucket settings, choose View details.
- General purpose buckets (13) Info:**
 - Create bucket** button
 - Buckets are containers for data stored in S3.**
 - Find buckets by name** search bar
 - Table Headers:** Name, AWS Region, Creation date
 - Items:**
 - aws-athena-query-results-814588432081-us-east-2-0pf98ayv (US East (Ohio) us-east-2, October 15, 2025, 11:15:47 (UTC+05:30))
 - aws-cloudtrail-logs-814588432081-0a7db287 (US East (N. Virginia) us-east-1, September 10, 2025, 23:55:50 (UTC+05:30))
- Account snapshot** section: Updated daily, View dashboard, Storage Lens provides visibility into storage usage and activity trends.
- External access summary** section: Updated daily, External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.

- The above image shows bucket created successfully.

The screenshot shows the AWS S3 console with the following details:

- Breadcrumbs:** Amazon S3 > Buckets > staticwebsitedemo2398 > Upload
- Upload** section:
 - Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. [Learn more](#)
 - Drag and drop files and folders you want to upload here, or choose Add files or Add folder.**
- Files and folders (2 total, 389.0 B)**
 - All files and folders in this table will be uploaded.
 - Find by name** search bar
 - Table Headers:** Name, Folder, Type, Size
 - Items:**
 - login.html (text/html, 196.0 B)
 - error.html (text/html, 193.0 B)
- Destination** section

- Search for bucket we created and click on it
- Click on upload and add files to bucket.

CloudFront & Route 53-Task

The screenshot shows the AWS S3 console interface. At the top, there's a green success message: "Upload succeeded. For more information, see the Files and folders table." Below this, the file list shows two files: "login.html" and "error.html". Both files are listed under the "s3://staticwebsitedemo2398" bucket. The "Files and folders" tab is selected. The table has columns for Name, Folder, Type, Size, Status, and Error. Both files have a size of 196.0 B and 195.0 B respectively, and both are marked as "Succeeded".

- Click on upload, we can see files uploaded into bucket.

The screenshot shows a browser window displaying an XML error response. The URL in the address bar is "staticwebsitedemo2398.s3.us-east-1.amazonaws.com/error.html". The content of the page is an XML document with the following structure:

```
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>HB6040G0VATW8E1Y</RequestId>
  <HostId>cAbPKXt52upFcqP5RQvOucnpJki7YHy9PBnf0kL0b06vohZwb8B+DkX3z1+3B8e570yDPVNYkuM=</HostId>
</Error>
```

- When we are trying to access bucket by object url it showing error so to rectify it we show follow some steps:-

Step-1: Turn OFF Block Public Access

Go to

S3 → Your bucket → Permissions

- Click **Block public access (Edit)**
- Uncheck **ALL** boxes
- Check "**I acknowledge...**"
- Click **Save**

CloudFront & Route 53-Task

The screenshot shows the 'Edit Block public access (bucket settings)' page. It includes a heading, a note about public access being granted through various mechanisms, and a list of checkboxes for different access control options. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with standard AWS footer links.

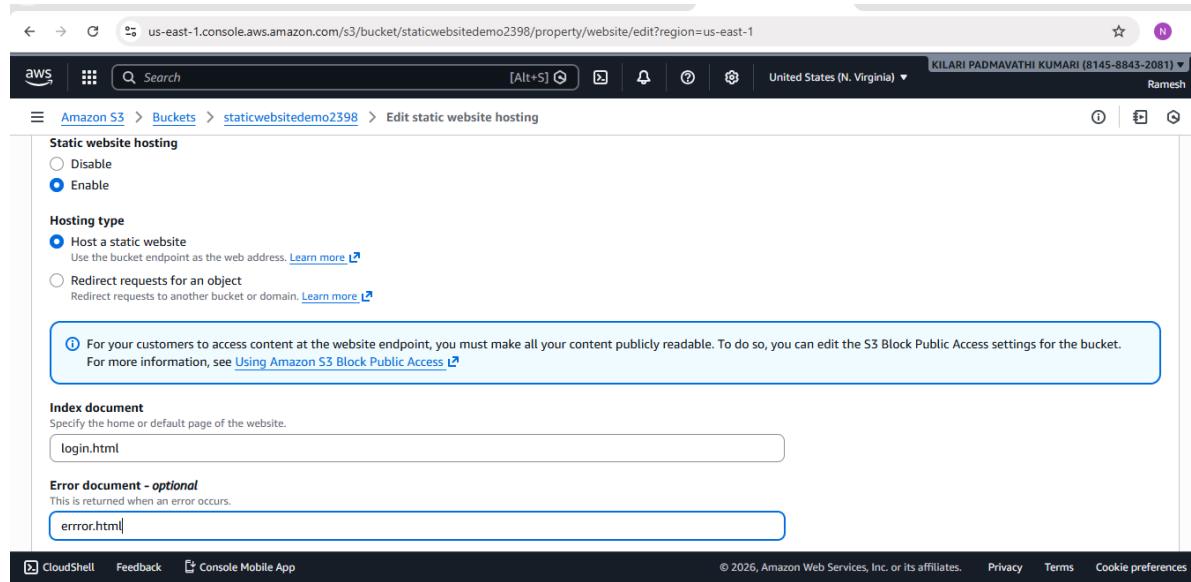
The screenshot shows the 'Bucket policy' editor. It displays a JSON policy document with several statements. To the right, there's a panel for editing a statement, with options to 'Edit statement', 'Select a statement', and 'Add new statement'. The bottom of the screen shows the usual AWS navigation and footer links.

- Edit bucket policy and add policy to it.

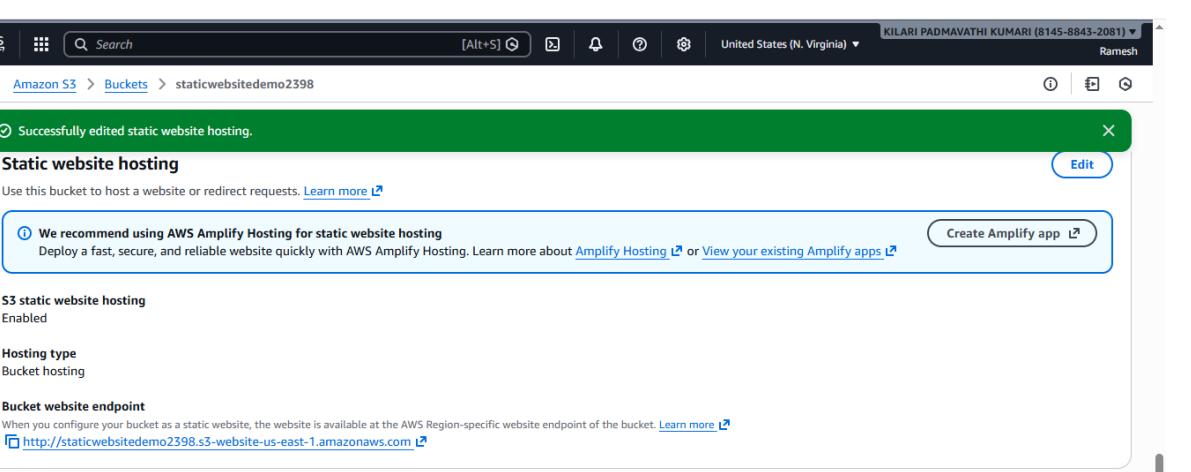
Step-3: Confirm Static Website Hosting

- Go to Properties → Static website hosting
- Specify the default page of the website

CloudFront & Route 53-Task



The screenshot shows the 'Edit static website hosting' page for a bucket named 'staticwebsitedemo2398'. Under 'Static website hosting', 'Enable' is selected. Under 'Hosting type', 'Host a static website' is selected. A note states: '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.' The 'Index document' field contains 'login.html' and the 'Error document - optional' field contains 'error.html'.



The screenshot shows a confirmation message: 'Successfully edited static website hosting.' It includes a note: '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.' Below this, it shows the 'S3 static website hosting' status as 'Enabled' and the 'Hosting type' as 'Bucket hosting'. The 'Bucket website endpoint' is listed as <http://staticwebsitedemo2398.s3-website-us-east-1.amazonaws.com>.

- It shows successfully edited static website hosting
- It provided bucket website endpoint
- By clicking on URL we access our static website.

CloudFront & Route 53-Task



Conclusion: The above image shows static website page which is present in s3 bucket.

CloudFront & Route 53-Task

4. Create a CDN and attach one SSL certificate.

- CDN and Attach one SSL certificate, for making your S3 website secure and global

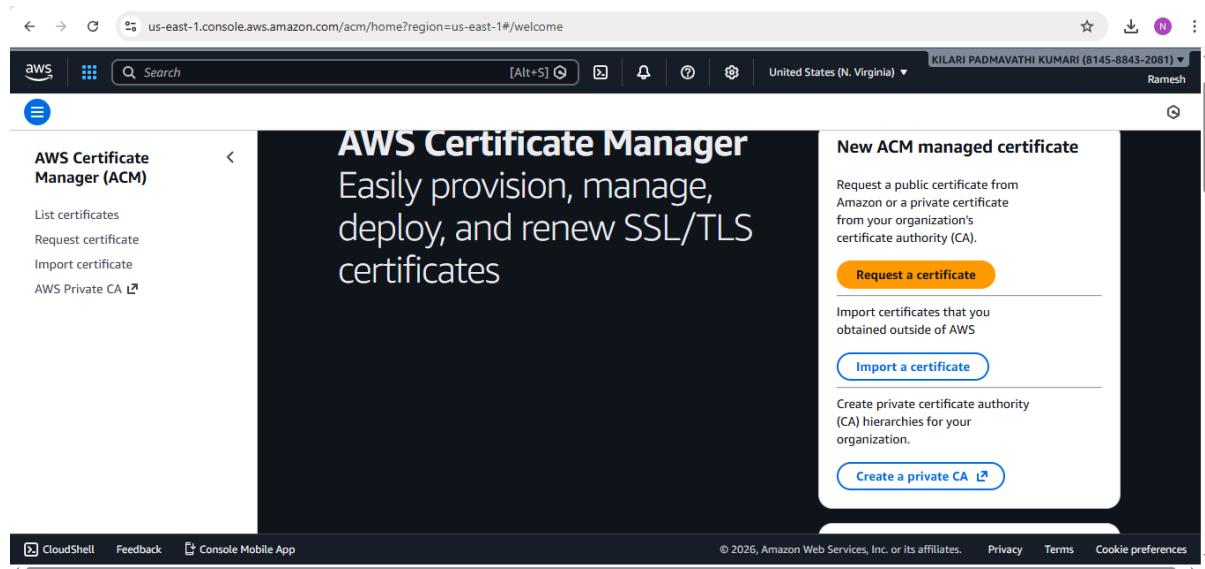
Requirements:

- Domain Purchased(neelimaranidevops.online)
- Subdomain Planned(www.neelimaranidevops.online)
- S3 Static Website Ready
- Bucket Name Must Match Domain

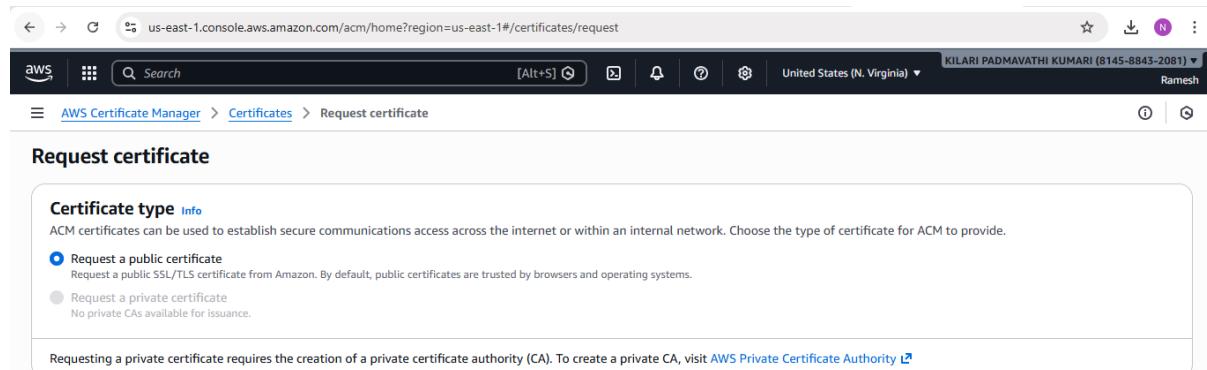
STEP 1— Request SSL Certificate (ACM)

CloudFront needs SSL in us-east-1

1. AWS → ACM
2. Change region → N. Virginia (us-east-1)
3. Click **Request certificate**
4. Enter:
neelimaranidevops.online
www.neelimaranidevops.online



CloudFront & Route 53-Task



Certificate type [Info](#)
ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

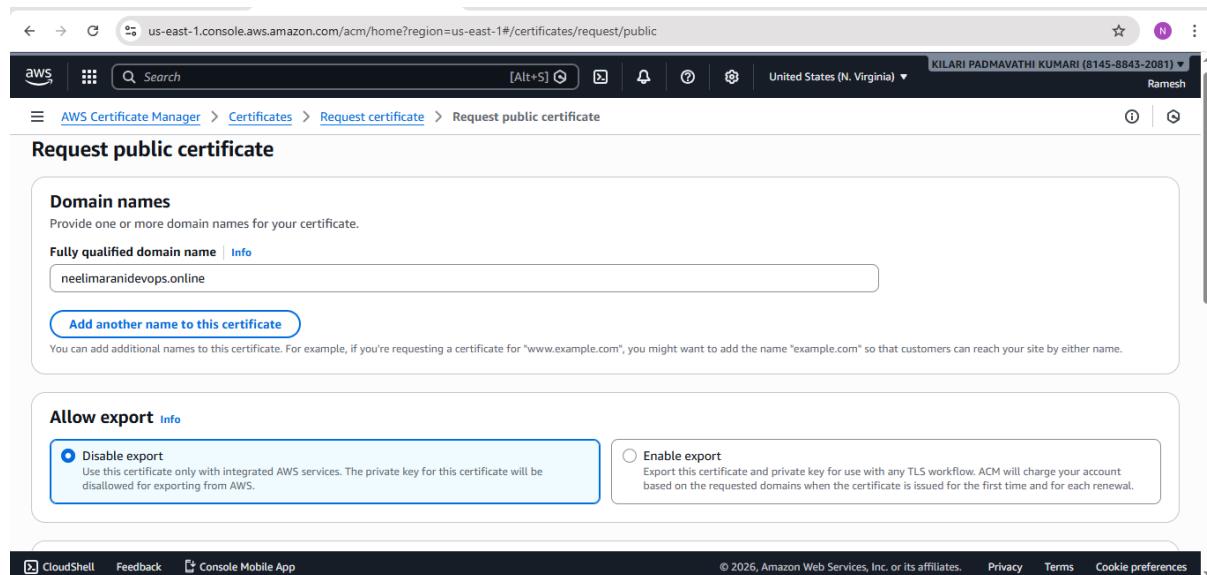
Request a private certificate
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit [AWS Private Certificate Authority](#).

[Cancel](#) [Next](#)



- Click on Request a public certificate.



Domain names
Provide one or more domain names for your certificate.

Fully qualified domain name [Info](#)

[Add another name to this certificate](#)
You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

Allow export [Info](#)

Disable export
Use this certificate only with integrated AWS services. The private key for this certificate will be disallowed for exporting from AWS.

Enable export
Export this certificate and private key for use with any TLS workflow. ACM will charge your account based on the requested domains when the certificate is issued for the first time and for each renewal.

[CloudShell](#) [Feedback](#) [Console Mobile App](#) © 2026, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- Add domain name :neelimaranidevops.online.

CloudFront & Route 53-Task

The screenshot shows the 'Request public certificate' step in the AWS Certificate Manager. It includes sections for 'Allow export', 'Validation method', and 'Key algorithm'. The 'DNS validation - recommended' option is selected. The status bar at the bottom indicates the certificate is pending validation.

- Validation method: click on DNS validations
- Click on create certificate.

The screenshot shows the 'Certificates' page after a certificate has been requested. A success message states 'Successfully requested certificate with ID ddadbdbb-0999-4d02-b8b6-f376cce8588d'. The certificate details show it is pending validation. One domain, 'ddadbdbb-0999-4d02-b8b6-f376cce8588d', is listed.

- The above image shows successfully request certificate with ID XXXx.

CloudFront & Route 53-Task

The screenshot shows the AWS Certificate Manager (ACM) console. A success message at the top states: "Successfully requested certificate with ID ddadbdbb-0999-4d02-b8b6-f376cce8588d. A certificate request with a status of pending validation has been created. Further action is needed to complete the validation and approval of the certificate." Below this, the certificate details are shown: Identifier (ddadbdbb-0999-4d02-b8b6-f376cce8588d), ARN (arn:aws:acm:us-east-1:814588432081:certificate/ddadbdbb-0999-4d02-b8b6-f376cce8588d), and Type (Amazon Issued). The status is listed as "Pending validation". At the bottom, there is a "Domains (1)" section with a "Create records in Route 53" button and an "Export to CSV" button.

- Scroll to Domains → neelimaranidevops.online
You will see CNAME Name and CNAME Value (DNS validation records).
- It will look like:

Name	Value
_a1b2c3.neelimaranidevops.online	_x9y8z.acm-validations.aws

The screenshot shows the GoDaddy domain management interface. On the left, there is a sidebar with options: Dashboard, Domain, Website, Email, Store, Appointments, and Marketing. The main area is titled "Add Organization Name" with a sub-instruction: "Missing an organization name in your domain contact info? If you're managing for an organization, adding its name can assert domain ownership and aid in disputes. Make sure it matches official documents for easy recovery. Ready to add it?". Below this, there are tabs for Overview, DNS (which is selected), Registration Settings, Products, and Activity Log. Under the DNS tab, there are sections for "DNS Records", "Forwarding", "Nameservers", "Hostnames", and "DNSSEC". The "DNS Records" section contains a "Add a new record" form with a "Add New Record" button. Other sections include "Easily verify domain ownership" with a "Verify Domain Ownership" button and "Create MX records" with a "Quickly create MX records to connect your domain" sub-section.

- GO to Godaddy website and click on Domain →DNS
- Add a new record in it.

CloudFront & Route 53-Task

New Records

CNAME records are a type of subdomain, or alias, that points to another domain name.

Type *	Name *	Value *	TTL
CNAME	_695c1f0328ecb084a24870d614612c	_8ac06868c0ee9a1fac275d91fc77fe3	1/2 Hour

Add More Records Save Cancel

Filters Actions

Add TYPE: CNAME ,NAME,VALUE,TTL and click on save.

The screenshot shows the GoDaddy control panel for a portfolio site. A modal window titled 'Create Now' displays a 'SUCCESS' message: 'Your DNS record has been updated successfully. Most DNS updates take effect within an hour, but could take up to 48 hours to update globally.' The main interface shows a table with columns: Type, Name, Data, and TTL. One row is visible with Type 'A', Name '@', Data 'WebsiteBuilder Site', and TTL '1 Hour'. The sidebar on the left lists 'Neelima Rani DevOps', 'Dashboard', 'Domain', 'Website', and 'Email'.

The screenshot shows the AWS Certificate Manager (ACM) console. It displays a certificate named 'ddadbdbb-0999-4d02-b8b6-f376cce8588d'. The 'Certificate status' section shows the identifier 'ddadbdbb-0999-4d02-b8b6-f376cce8588d', ARN 'arn:aws:acm:us-east-1:814588432081:certificate/ddadbdbb-0999-4d02-b8b6-f376cce8588d', and Type 'Amazon Issued'. The status is 'Issued'. Below this, the 'Domains (1)' section shows a single domain entry: 'Domain' (with value 'WebsiteBuilder Site'), 'Status' (with value 'Issued'), 'Renewal status' (with value 'Not due'), 'Type' (with value 'A'), and 'CNAME name' (with value '').

- And certificate status will be shown as issued.

STEP 4 — Add CloudFront Records in Route 53

Go back to:

Route 53 → Hosted Zone → neelimaranidevops.online → Create record

CloudFront & Route 53-Task

Create:

The screenshot shows the AWS Route 53 console. On the left, the navigation menu includes 'Route 53' (selected), 'Dashboard', 'Hosted zones' (selected), 'Health checks', 'Profiles', 'Global Resolver', 'VPC Resolver', and 'Domains'. The main content area displays 'Hosted zones (1/1)' with a table showing one entry: 'neelimarandevops.online' (Type: Public, Created by: Route 53). A 'Create hosted zone' button is visible. To the right, the 'Hosted zone details' pane shows the hosted zone name, ID, description, query log, type (Public hosted zone), and record count (2).

This screenshot shows the 'Hosted zone details' page for the 'neelimarandevops.online' zone. It lists 'Records (2)'. The first record is an NS record for 'neelimara...' pointing to 'ns-1821.awsdns-ns-499.awsdns-6-ns-1153.awsdns-ns-722.awsdns-2.'. The second record is an SOA record for 'neelimara...' pointing to 'ns-1821.awsdns-ns-499.awsdns-6-ns-1153.awsdns-ns-722.awsdns-2.'. Both records are of type 'Simple'.

This screenshot shows the 'Create record' page for a new subdomain. The 'subdomain' field is set to 'neelimarandevops.online'. The 'Route traffic to' section is configured for an 'Alias' to a CloudFront distribution in 'US East (N. Virginia)'. The 'Evaluate target health' option is set to 'No'. The 'Create records' button is highlighted at the bottom right.

CloudFront & Route 53-Task

- We required cloud front distribution.

The screenshot shows the 'Choose a plan' step in the CloudFront distribution creation wizard. The left sidebar lists steps: Step 1 (Choose a plan), Step 2 (Get started), Step 3 (Specify origin), Step 4 (Enable security), Step 5 (Get TLS certificate), Step 6 (Review and create). Step 1 is selected. The main area shows a 'Free' plan with '\$0/month' cost, intended for hobbyists, learners, and developers. A callout box highlights features: Always-on DDoS protection, Protect against common web threats with AWS WAF, IP-based rate limiting, Geographic traffic blocking, Serverless edge compute, Global CDN, DNS, Free TLS certificate, Tiered caching, Default caching rules, and Fast cache invalidations. A 'Learn more' button is also present.

The screenshot shows the 'Get started' step in the CloudFront distribution creation wizard. The left sidebar lists steps: Step 1 (Choose a plan), Step 2 (Get started), Step 3 (Specify origin), Step 4 (Enable security), Step 5 (Get TLS certificate), Step 6 (Review and create). Step 2 is selected. The main area shows 'Distribution options'. It includes fields for 'Distribution name' (set to 'create_first_distribution'), 'Description - optional' (empty), and 'Distribution type'. Two options are shown: 'Single website or app' (selected) and 'Multi-tenant architecture - New'. The 'Single website or app' option is described as choosing if each website or application will have a unique configuration.

- Go to cloud front → distribution → create distribution.

CloudFront & Route 53-Task

The screenshot shows the 'Create distribution' wizard in the AWS CloudFront console. The first step, 'Distribution type', is completed. The 'Single website or app' option is selected, with a note explaining it's chosen for each website or application. The 'Multi-tenant architecture - New' option is also available for multiple domains sharing configurations. The next step, 'Domain Info', is shown below, where the domain 'neelimarandidevops.online' is entered and a 'Check domain' button is present. A 'Tags - optional' section is also visible. Navigation buttons 'Cancel', 'Previous', and 'Next' are at the bottom.

- Check domain and click next

The screenshot shows the 'Create distribution' wizard in the AWS CloudFront console, moving to the 'Origin' step. An 'S3 origin' is selected, with the URL 'neelimarandidevops.online.s3.us-east-1.amazonaws.com' entered. A 'Browse S3' button is available. The 'Origin path - optional' field contains '/path'. In the 'Settings' section, the 'Allow private S3 bucket access to CloudFront' checkbox is checked. The 'Origin settings' section is partially visible at the bottom. Navigation buttons 'Cancel', 'Previous', and 'Next' are at the bottom.

- Specify origin → s3 origin browse s3
- Settings → Allow private s3 bucket access to CloudFront

CloudFront & Route 53-Task

The screenshot shows the AWS CloudFront 'Create distribution' wizard at Step 5: 'Get TLS certificate'. On the left, a vertical navigation bar lists steps from 1 to 6: Step 1 (Choose a plan), Step 2 (Get started), Step 3 (Specify origin), Step 4 (Enable security), Step 5 (Get TLS certificate), and Step 6 (Review and create). Step 5 is highlighted with a blue circle. The main content area is titled 'Get TLS certificate' and contains a 'TLS certificate' section with a 'Info' link. It states: 'Transport layer security (TLS) encrypts communication to and from your domain. You must have a TLS certificate with AWS Certificate Manager (ACM) to use CloudFront.' Below this, there are two options: 'Available certificates' (selected, showing 'neelimaranidevops.online') and 'Create a new certificate'. A 'View in AWS Certificate Manager' button is also present. At the bottom, it shows the ARN: arn:aws:acm:us-east-1:814588432081:certificate/ddadbd8-0999-4d02-b8b6-f376cce8588d and the 'Covered domains' neelimaranidevops.online.

- Get TLS certificate
- Select Available certificate
- Click on create distribution.

The screenshot shows the AWS CloudFront distribution details page for 'create_first_distribution'. At the top, a green success message says 'Successfully created new distribution.' Below it, the distribution name 'create_first_distribution' is shown with a 'Free plan' badge. The 'General' tab is selected. In the 'Details' section, it shows the 'Distribution domain name' as dvtxxwhudan77.cloudfront.net, 'Billing' as 'Free plan (\$0/month)', and 'ARN' as arn:aws:cloudfront:814588432081:distribution/E1UMDHG2Q47GFK. The 'Last modified' status is 'Deploying'. Other tabs include 'Security', 'Origins', 'Behaviors', 'Error pages', 'Invalidations', 'Logging', and 'Tags'. The 'Settings' section shows the 'Name' as 'create_first_distribution' and 'Alternate domain names' as 'neelimaranidevops.online'. A note indicates 'Standard logging Available with the Pro plan' and a 'Edit' button. The bottom of the screen includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.

- The above image shows distribution created successfully.

CloudFront & Route 53-Task

The screenshot shows the AWS Route 53 'Create record' interface. A new alias record is being created for the subdomain 'dvtxxwhudan77.cloudfront.net'. The 'Route traffic to' dropdown shows 'Alias to CloudFront distribution' and 'US East (N. Virginia)'. The 'Use:' field contains 'dvtxxwhudan77.cloudfront.net'. The 'Create records' button is visible at the bottom right.

- Go to Route 53 → Hosted zones → neelimaranidevops.online
- Route traffic to Alias to cloudfont distribution.
- Add cloudfont distribution ID
- Click on create records

The screenshot shows the AWS Route 53 'Hosted zones' interface for the domain 'neelimaranidevops.online'. It displays three successfully created records. A success message indicates that the record was created and propagated. The 'Records' tab is selected, showing three entries. The 'Edit hosted zone' button is visible. The sidebar shows navigation options like Route 53, Hosted zones, Global Resolver, VPC Resolver, and Domains.

- The above image shows Records for domain was successfully created.

CloudFront & Route 53-Task

5. Create a Route 53 hosted zone and map the domain with the CDN.

- Go to:

AWS → Route 53 → Hosted Zones → Create hosted zone

The screenshot shows the 'Get started' section of the AWS Route 53 console. It displays six options: 'Register a domain', 'Transfer domain', 'Create hosted zones' (which is selected and highlighted with a blue border), 'Configure health checks', 'Configure traffic flow', and 'Configure resolvers'. Each option has a brief description and a corresponding icon.

The screenshot shows the 'Create hosted zone' configuration page. It includes fields for 'Domain name' (neelimarandevops.online), 'Description - optional' (The hosted zone is used for...), and 'Type' (Public hosted zone). The 'Public hosted zone' option is selected and highlighted with a blue border.

- Add domain name as neelimarandevops.online
- Type: public hosted zone

CloudFront & Route 53-Task

The screenshot shows the AWS Route 53 console. On the left, a navigation menu includes 'Route 53', 'Hosted zones' (which is selected), 'Global Resolver', 'VPC Resolver', and 'Domains'. The main area displays a success message: 'neelimaranidevops.online was successfully created. Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.' Below this, the 'neelimaranidevops.online' hosted zone is listed as 'Public'. It shows 'Records (2)' and buttons for 'Delete zone', 'Test record', and 'Configure query logging'. A 'Hosted zone details' section is also visible.

- Click on create zone.
- The above image shows create records in the hosted zone to specify how you want to route 53 to route traffic for your domain.

The screenshot shows the 'Records' tab for the 'neelimaranidevops.online' hosted zone. It lists two records: one NS record and one SOA record. The NS record has four entries: ns-1821.awsdns-35.co.uk., ns-499.awsdns-62.com., ns-1153.awsdns-16.org., and ns-722.awsdns-26.net. The SOA record has one entry: ns-1821.awsdns-35. The right side of the screen shows 'Record details' for the selected NS record, including fields for Record name, Record type (NS), Value, Alias, and TTL (seconds).

- Go to route→hosted Zones→ neelimaranidevops.online
- Click on record details in that we can see namesevers.

CloudFront & Route 53-Task

The screenshot shows the GoDaddy domain control interface for the domain 'neelimaranidevops.online'. The 'Nameservers' tab is selected in the top navigation bar. On the left, there's a sidebar with options like Dashboard, Domain, Website, Email, Store, Appointments, and Marketing. The main content area displays the current nameservers: 'ns77.domaincontrol.com' and 'ns78.domaincontrol.com'. A button labeled 'Change Nameservers' is visible.

- Now go to godaddy website in that domain → DNS → nameservers
- Click on change nameservers.

The screenshot shows the 'Edit nameservers' dialog box. It asks to choose nameservers for the domain 'neelimaranidevops.online'. There are two options: 'GoDaddy Nameservers (recommended)' (radio button) and 'I'll use my own nameservers' (radio button, which is selected). Below this, four nameserver entries are listed: 'ns-1821.awsdns-35.co.uk', 'ns-499.awsdns-62.com', 'ns-1153.awsdns-16.org', and 'ns-722.awsdns-26.net'. Each entry has a small trash can icon to its right. At the bottom, there's a '+ Add Nameserver' link, and 'Save' and 'Cancel' buttons.

- Add nameservers which are present in route53.

CloudFront & Route 53-Task

```
user@DESKTOP-3KH1LIRE MINGW64 ~ (master)
$ nslookup -type=NS neelimaranidevops.online
Non-authoritative answer:
Server: gpon.net
Address: fe80::1

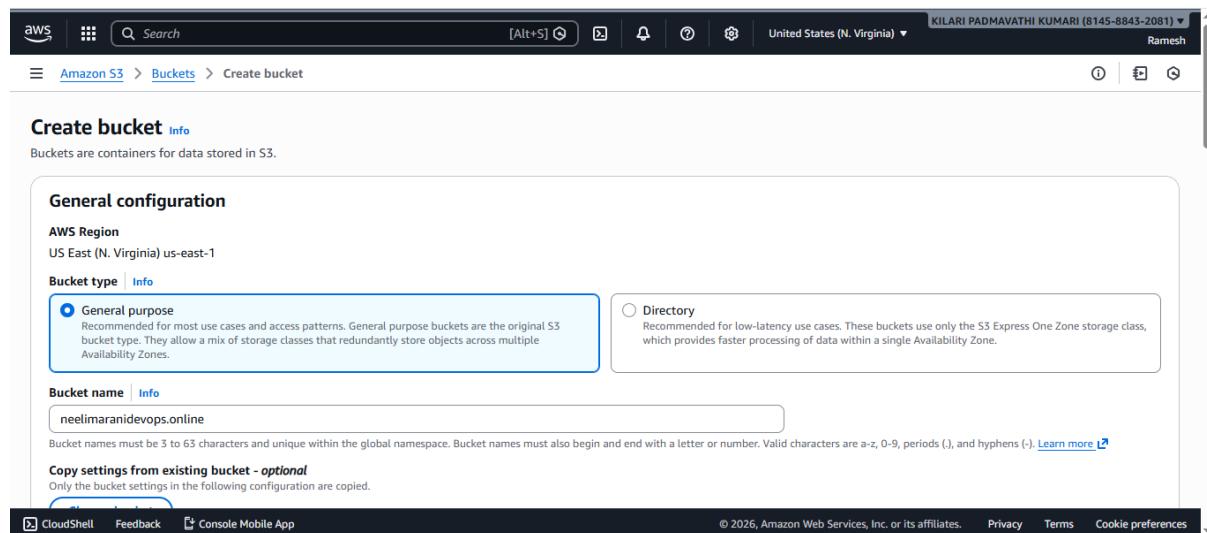
neelimaranidevops.online      nameserver = ns-1153.awsdns-16.org
neelimaranidevops.online      nameserver = ns-499.awsdns-62.com
neelimaranidevops.online      nameserver = ns-722.awsdns-26.net
neelimaranidevops.online      nameserver = ns-1821.awsdns-35.co.uk
```

Conclusion :To verify where they are added or not

Command:- Nslookup -type=NS neelimaranidevops.online

6. Update the index.html in the S3 bucket and ensure the updated file is accessible using the domain name.

- Go to Aws console →S3→create bucket
- Bucket name should be same as domain name.



CloudFront & Route 53-Task

The screenshot shows the AWS S3 'Create bucket' page. In the top navigation bar, the user is identified as 'KILARI PADMAVATHI KUMARI (8145-8843-2081)' and the location is 'United States (N. Virginia)'. The main content area is titled 'BLOCK PUBLIC Access settings for this bucket'. It contains several checkboxes for configuring public access:

- 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.

A warning message in a yellow box states: 'Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.' A checkbox below it says 'I acknowledge that the current settings might result in this bucket and the objects within becoming public.'

At the bottom of the page, there are links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information: '© 2026, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

- Block public Access settings for this bucket
- Click on confirmation.

The screenshot shows the AWS S3 'Create bucket' page. In the top navigation bar, the user is identified as 'KILARI PADMAVATHI KUMARI (8145-8843-2081)' and the location is 'United States (N. Virginia)'. The main content area includes sections for 'Encryption' and 'Bucket Key'.

Encryption: A note says 'Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#)'. Below it are three radio buttons:

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Bucket Key: A note says 'Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)'. Below it are two radio buttons:

- Disable
- Enable

A blue button labeled 'Advanced settings' is visible. A note at the bottom says 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.'

At the bottom right are 'Cancel' and 'Create bucket' buttons. At the very bottom are links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information: '© 2026, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

- Click on create bucket

CloudFront & Route 53-Task

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, search bar, and various links like 'Ask Amazon Q'. The main header says 'neelimaranidevops.online' with a 'Info' link. Below the header, there's a tab bar with 'Objects' (which is selected), 'Metadata', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Under the 'Objects' tab, it says '(0)' and has a large orange 'Upload' button. There's also a 'Find objects by prefix' input field and some sorting options for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message at the bottom states 'No objects' and 'You don't have any objects in this bucket.'.

- Go to Bucket and click on objects
- Upload files into bucket.

The screenshot shows the 'Edit static website hosting' configuration page for the 'neelimaranidevops.online' bucket. At the top, there's a navigation bar with the AWS logo, search bar, and various links like 'Ask Amazon Q'. The main header says 'Edit static website hosting' with a 'Info' link. Below the header, there's a section titled 'Static website hosting' with a note: 'Use this bucket to host a website or redirect requests.' and a 'Learn more' link. It has two radio buttons: 'Disable' (unchecked) and 'Enable' (checked). Below that is a 'Hosting type' section with two radio buttons: 'Host a static website' (checked) and 'Redirect requests for an object' (unchecked). A note below the first radio button says: '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](#)'.

- Static website hosting should be enable

The screenshot shows the 'Edit static website hosting' configuration page for the 'neelimaranidevops.online' bucket. At the top, there's a navigation bar with the AWS logo, search bar, and various links like 'Ask Amazon Q'. The main header says 'Edit static website hosting' with a 'Info' link. Below the header, there's a section titled 'Hosting type' with the same two radio button options as the previous screenshot. A note below the first radio button says: '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](#)'. Below the hosting type section is an 'Index document' section with a note: 'Specify the home or default page of the website.' and a text input field containing 'login.html'. There's also an 'Error document - optional' section with a note: 'This is returned when an error occurs.' and a text input field containing 'error.html'. At the bottom, there's a 'Redirection rules - optional' section with a note: 'Redirection rules, written in JSON, automatically redirect webpage requests for specific content.' and a text input field.

CloudFront & Route 53-Task

- Hosting type host a static website.
- Add Index document.

The screenshot shows the AWS S3 Bucket Policy editor. The policy is defined in JSON:

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "PublicReadGetObject",
6        "Effect": "Allow",
7        "Principal": "*",
8        "Action": [
9          "s3:GetObject"
10        ],
11        "Resource": [
12          "arn:aws:s3:::neelimaranidevops.online/*"
13        ]
14      }
    ]
}
```

The right side of the screen shows the policy structure with an "Edit statement" button, a "Remove" button, and a sidebar for "Add actions" and "Included" services (S3). The bottom of the screen shows standard AWS navigation links and copyright information.

- Edit bucket policy
- Add policy to bucket
- By using bucket URL we can access our static website as shown below.

neelimaranidevops.online.s3-website-us-east-1.amazonaws.com

Welcome to My Static Website!

This is the home page hosted on Amazon S3.

CloudFront & Route 53-Task

- Now Go to Route53 →hosted Zones→neelimaranidevops.online
- Create Record.
- Click on Alias Route traffic to Alias to s3 website endpoint.
- Click on create Records.

The screenshot shows the AWS Route 53 console interface for creating a new record set. The URL is us-east-1.console.aws.amazon.com/route53/v2/hostedzones?region=us-east-1#CreateRecordSet/Z08733002VRDBE2TKF4C1. The user is on the 'Create record' sub-page under 'Hosted zones' for the domain 'neelimaranidevops.online'.
Form fields:

- Record name:** subdomain (highlighted)
- Record type:** A – Routes traffic to an IPv4 address and some AWS resources
- Route traffic to:** Alias (selected)
- Alias:** s3-website.ap-south-1.amazonaws.com (selected from dropdown)
- Evaluate target health:** Yes (checkbox)

Buttons at the bottom: **Add another record**, **Cancel**, and **Create records**.

The screenshot shows the GoDaddy Domain Manager interface for editing nameservers. The user is choosing nameservers for the domain 'neelimaranidevops.online'.
Options:

- GoDaddy Nameservers (recommended) (radio button)
- I'll use my own nameservers (radio button, selected)

Nameservers listed:

- ns-1821.awsdns-35.co.uk.
- ns-499.awsdns-62.com.
- ns-1153.awsdns-16.org.
- ns-722.awsdns-26.net.

Buttons: **Save** and **Cancel**.

- Go to godaddy website and domain→DNS→ add nameservers to it

CloudFront & Route 53-Task

The screenshot shows the AWS Certificate Manager (ACM) console. 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 'Certificates (1)' and shows a single certificate entry:

Certificate ID	Domain name	Type	Status
ddadbdbb-0999-4d02-b8b6-f376cce8588d	neelimaranidevops.online	Amazon Issued	Issued

At the bottom, there are links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information: '© 2026, Amazon Web Services, Inc. or its affiliates.'

- Go to Aws certificate Manager→certificates→request

The screenshot shows the details of a specific certificate identified by the ARN: arn:aws:acm:us-east-1:814588432081:certificate/06cb08b1-c206-4c39-a217-9ee5bfaae13e. The page includes sections for 'Certificate status' and 'Domains (1)'. In the 'Status' section, it shows 'Pending validation' with a link to 'Info'. In the 'Domains' section, there is one entry for 'neelimaranidevops.online'.

Domain	Status	Renewal status	Type	CNAME name
neelimaranidevops.online	Pending validation	Pending validation	Amazon Issued	

At the bottom, there are links for 'CloudShell', 'Feedback', 'Console Mobile App', and copyright information: '© 2026, Amazon Web Services, Inc. or its affiliates.'

- Request a certificate to our domain.
- Click on Record in route53

CloudFront & Route 53-Task

The screenshot shows the AWS Certificate Manager interface. At the top, it displays a success message: "Successfully requested certificate with ID 06cb08b1-c206-4c39-a217-9ee5bfaae13e". Below this, the "Create DNS records in Amazon Route 53" section is shown. It lists one domain: "neelimaranidevops.online" with a validation status of "Pending validation". There are filters for "Validation status = Pending validation" and "Is domain in Route 53? = Yes". At the bottom right are "Cancel" and "Create records" buttons.

- After clicking on create record

The screenshot shows the AWS Route 53 interface under the "Hosted zones" section for the domain "neelimaranidevops.online". On the left, there's a navigation menu with options like "Route 53", "Dashboard", "Hosted zones", "Health checks", "Profiles", "Global Resolver", "VPC Resolver", and "Domains". The "Hosted zones" section is selected. In the main area, the "Records (4) Info" table is displayed. It shows four records:

- A record for "neelimara..." with Type "A", Value "s3-website.ap-south-1.amazonaws.com", and Alias "Yes".
- NS records for "ns-1821.awsdns-35.co", "ns-499.awsdns-62.co", "ns-1153.awsdns-16.o", and "ns-722.awsdns-26.ne".
- SOA record for "neelimara..." with Type "SOA", Value "ns-1821.awsdns-35.co".
- CNAME record for ".695c1f0..." with Type "CNAME", Value ".8ac06868c0ee9a1fa".

Check in Route53 → hosted zones → neelimaranidevops.online → A record is added.

The screenshot shows the AWS Certificate Manager interface for the certificate "06cb08b1-c206-4c39-a217-9ee5bfaae13e". The "Certificate status" section shows the identifier "06cb08b1-c206-4c39-a217-9ee5bfaae13e", ARN "arn:aws:acm:us-east-1:814588432081:certificate/06cb08b1-c206-4c39-a217-9ee5bfaae13e", and Type "Amazon Issued". The status is listed as "Issued". Below this, the "Domains (1)" section shows the domain "neelimaranidevops.online" with a status of "Issued". There are buttons for "Create records in Route 53" and "Export to CSV".

- Then in AWS certificate status shown as issued

CloudFront & Route 53-Task

The screenshot shows the AWS CloudFront 'Create distribution' wizard at Step 3: Specify origin. In the 'Origin type' section, 'Amazon S3' is selected. Below it, other options like 'Elastic Load Balancer', 'VPC origin', and 'Other' are shown. In the 'Origin' section, 'S3 origin' is selected, and the URL 'neelimaranidevops.online.s3-website-us-east-1.amazonaws.com' is entered. A 'Browse S3' button is available. The bottom of the screen shows standard AWS navigation links: CloudShell, Feedback, Console Mobile App, Privacy, Terms, and Cookie preferences.

- Go to cloudfont → distributions → create distributions.
- Origin type as Amazon s3.

The screenshot shows the 'Create distribution' wizard at a later step, focusing on 'Origin settings'. 'Customize origin settings' is selected. Other options like 'Use recommended origin settings' are shown. Below, there's a section for 'Add custom header - optional' with a 'Add header' button. Under 'Protocol', 'HTTP only' is selected. There's also an 'Origin Shield' section (disabled) and an 'HTTP port' field set to 80. The bottom of the screen shows standard AWS navigation links: CloudShell, Feedback, Console Mobile App, Privacy, Terms, and Cookie preferences.

- Origin settings customize origin settings
- Protocol as HTTP only.

CloudFront & Route 53-Task

The screenshot shows the 'Create distribution' configuration page in the AWS CloudFront console. Key settings visible include:

- Cache settings:** 'Customize cache settings' is selected.
- Viewer protocol policy:** 'Redirect HTTP to HTTPS' is selected.
- Allowed HTTP methods:** 'GET, HEAD' is selected.
- Cache policy:** 'CachingOptimized' is selected, described as 'Policy with caching enabled. Supports Gzip and Brotli compression.'

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with standard AWS footer links for Privacy, Terms, and Cookie preferences.

- Cache setting → customize cache settings
- Viewer protocol policy redirect HTTP to HTTPS,
- Allowed HTTP methods GET, HEAD
- Click on Create distribution

The screenshot shows the confirmation page after creating a new distribution. It displays the following information:

- Successfully created new distribution:** EGOSLXMEYNEBY
- create_first_distribution** (Free plan)
- Details:**
 - Distribution domain name: d2gzm49v85yrea.cloudfront.net
 - Billing: Free plan (\$0/month)
 - ARN: arn:aws:cloudfront::814588432081:distribution/EGOSLXMEYNEBY
 - Last modified: Deploying

Below the details, tabs for General, Security, Origins, Behaviors, Error pages, Invalidations, Logging, and Tags are visible. A 'View metrics' button is also present.

The screenshot shows the 'Add domain' configuration page for the distribution EGOSLXMEYNEBY. It includes:

- Step 1:** Configure domains (selected).
- Step 2:** Get TLS certificate (selected).
- Step 3:** Review changes.

Get TLS certificate: A sub-section showing the process to obtain a TLS certificate. It includes a 'TLS certificate' section with a 'Info' link and a 'Refresh certificates' button, and an 'Available certificates' section listing two options:

- neelimarandevops.online (ddadbbbbb-0999-4d02-b8b6-f376cce8588d)
- neelimarandevops.online (06cb08b1-c206-4c39-a217-9ee5bfaae13e)

Certificate details: Shows the ARN of the selected certificate and the covered domains.

CloudFront & Route 53-Task

- Click on Distribution Id → Add domain
- Get TLS certificate
- Choose certificate issued for our domain.

The screenshot shows the AWS CloudFront 'Distributions' page. A green success message at the top says 'Distribution updated successfully'. Below it, the distribution details are shown: Name: 'create_first_distribution', Free plan (\$0/month), ARN: 'arn:aws:cloudfront::814588432081:distribution/EGOSLXMEYNEBY', and Last modified: 'Deploying'. The 'General' tab is selected. In the 'Settings' section, under 'Alternate domain names', the value 'neelimaranidevops.online' is listed. There is an 'Edit' button next to it.

- Add Alternate Domain names → neelimaranidevops.online

The screenshot shows the AWS Route 53 'Hosted zones' page for the domain 'neelimaranidevops.online'. On the left, the navigation menu includes 'Route 53', 'Hosted zones', 'Global Resolver', 'VPC Resolver', and 'Domains'. Under 'Hosted zones', there is a table of records:

Type	Name	Value	Alias
A	neelimara...	s3-website.ap-sou...	Yes
NS	neelimara...	ns-1821.awsdns-ns-499.awsdns-6...	No
SOA	neelimara...	ns-1821.awsdns-ns-1153.awsdns-722.awsdns-2...	No
CNAME	_695c1f0...	_8ac6868c0ee9	No

To the right, a modal window titled 'A - Routes traffic to an ...' is open, showing the configuration for an 'Alias' record pointing to a CloudFront distribution. It includes fields for 'Route traffic to' (set to 'Alias'), 'Choose distribution' (set to 'CloudFront dist...'), and 'US East (N. Virginia)'. There is also a 'Save' button at the bottom.

- Go to route 53
- Click on A record edit record route traffic to Alias to cloud front distribution instead of s3 bucket endpoint.
- Click on save.

CloudFront & Route 53-Task

The screenshot shows the AWS Route 53 console. In the top navigation bar, the user is identified as KILARI PADMAVATHI KUMARI (8145-8843-2081) and Ramesh. The left sidebar menu includes options like Route 53, Dashboard, Hosted zones, Global Resolver, VPC Resolver, and Domains. Under Hosted zones, the domain neelimaranidevops.online is selected. The main content area displays a success message: "neelimaranidevops.online was successfully updated. Route 53 propagates your changes to all of the Route 53 authoritative DNS servers within 60 seconds. Use "View status" button to check propagation status." Below this, a table lists four records: one A record (ns-1821.awsdns-499.awsdns-6.ns-1153.awsdns-722.awsdns-2), one NS record (ns-1821.awsdns-499.awsdns-6.ns-1153.awsdns-722.awsdns-2), and two SOA records (ns-1821.awsdns-499.awsdns-6.ns-1153.awsdns-722.awsdns-2).

- Here we can see update record successfully.

The screenshot shows a web browser window with the URL neelimaranidevops.online. The page content reads "Welcome to My Static Website! This is the home page hosted on Amazon S3."

- Now we can access our website with our domain name.

7. Share the domain name in Slack to test the connectivity.

Domain name: neelimaranidevops.online