

# Map the ip with Domain and SSL Encryption

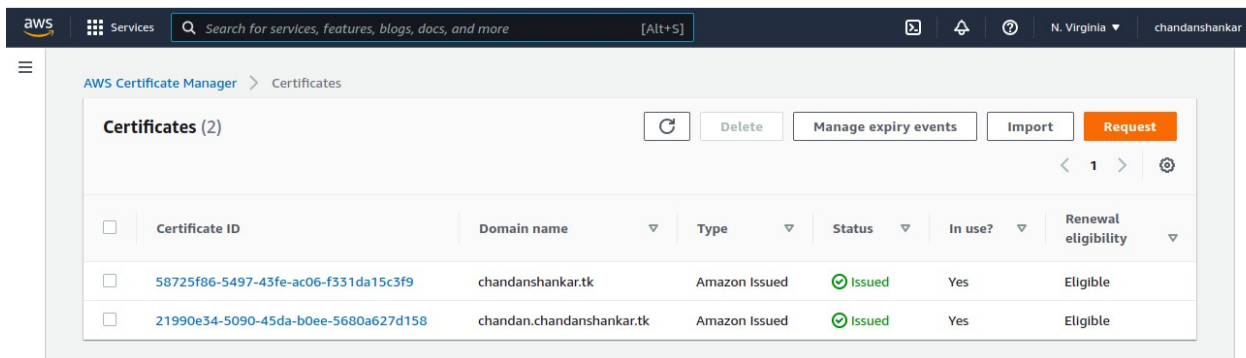
## Task1. Encrypton using AWS Certification

### Step 1

- Using Route53 just map the server names and A records
- and we can access in domain name

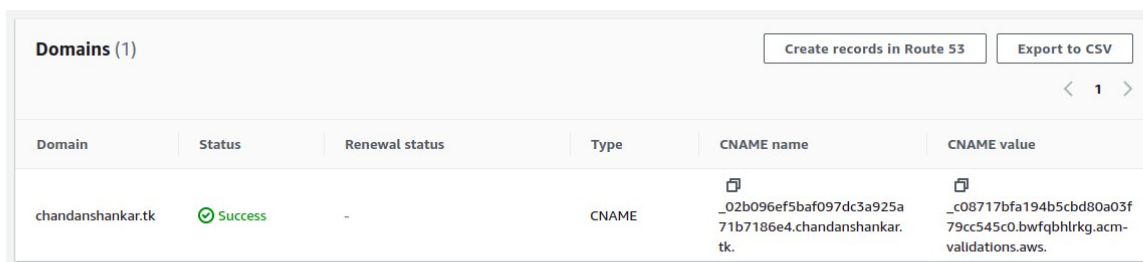
### Step 2

- Create certificate in AWS certificate manager for your domain name
- And inside certification we need to create CNAME for that Domain name



The screenshot shows the AWS Certificate Manager console. At the top, there's a search bar and navigation links. Below, the 'Certificates (2)' section is active. It contains a table with two certificates, both in 'Issued' status. The first certificate is for 'chandanshankar.tk' and the second is for 'chandani.chandanshankar.tk'. Both are Amazon Issued and eligible for renewal.

Certificate ID	Domain name	Type	Status	In use?	Renewal eligibility
58725f86-5497-43fe-ac06-f331da15c3f9	chandanshankar.tk	Amazon Issued	Issued	Yes	Eligible
21990e34-5090-45da-b0ee-5680a627d158	chandani.chandanshankar.tk	Amazon Issued	Issued	Yes	Eligible



The screenshot shows the AWS Route 53 console. The 'Domains (1)' section is active. It contains a table with one domain, 'chandanshankar.tk', which has a 'Success' status. The table also shows the CNAME name and CNAME value for the domain.

Domain	Status	Renewal status	Type	CNAME name	CNAME value
chandanshankar.tk	Success	-	CNAME	_02b096ef5baf097dc3a925a71b7186e4.chandanshankar.tk.	_c08717bfa194b5cbd80a03f79cc545c0.bwfbqhlrkg.acm-validations.aws.

- and Wait until certificates status get issued
- with this step we are sucessfully get certificates in AWS but we need integrtae with project

### Step 3

- To attach SSL to our webpage we are create application load balancer to instance
- in network mapping select vpc and availibilty region

- then choose security group
- and listeners and routings (port 80)
- along with we need to choose the valid certification for the page
- we can add optional content like add-on service and tags
- finally we get summary to launch load balancer before launch
- and finally create load balancers

EC2 > Load balancers > Create Application Load Balancer

## Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

### Basic configuration

**Load balancer name**  
Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** [Info](#)  
Scheme cannot be changed after the load balancer is created.

☒ **Internet-facing**  
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ **Internal**  
An internal load balancer routes requests from clients to targets using private IP addresses.

aws Services  N. Virginia chandan

### Security groups

Select up to 5 security groups

[Create new security group](#)

default sg-0ea213637a33ce6cd   
VPC: vpc-0dfffbc11b95f1d0

### Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTP:80

Protocol: HTTP Port: 80 Default action: Forward to

1-65535

- summary part of load balancer

**Tags - optional**

Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

**Summary**

Review and confirm your configurations. [Estimate cost](#)

**Basic configuration** [Edit](#)

Load balancer name not defined

- Internet-facing
- IPv4

**Security groups** [Edit](#)

- default [sg-0ea213637a33ce6cd](#)

**Network mapping** [Edit](#)

VPC [vpc-0dfffbc11b95f1d0](#)

Subnet not defined

**Listeners and routing** [Edit](#)

- HTTP:80 defaults to [Target group not defined](#)

**Add-on services** [Edit](#)

None

**Tags** [Edit](#)

None

**Attributes**

[i](#) Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

[Cancel](#) [Create load balancer](#)

- After create load balancer wait some time for load balancer active state

## Step 4

- then we can access the page using load balancer DNS name

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers:sort=loadBalancerName

[Create Load Balancer](#) [Actions](#)

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones
domainnmaelb	domainnmaelb-1524601104.us-east-1.elb.amazonaws.com	Active	vpc-0dfffbc11b95f1d0	us-east-1d, us-east-1e
new	new-258016103.us-east-1.elb.amazonaws.com	Active	vpc-0dfffbc11b95f1d0	us-east-1d, us-east-1e

**Basic Configuration**

**Name** domainnmaelb

**ARN** [arn:aws:elasticloadbalancing:us-east-1:359956665393:loadbalancer/app/domainnmaelb/5ced6d016ce6cbeb](#)

**DNS name** [domainnmaelb-1524601104.us-east-1.elb.amazonaws.com](#) (A Record)

**State** Active

**Type** application

**Scheme** internet-facing

**IP address type** ipv4 [Edit IP address type](#)

**VPC** [vpc-0dfffbc11b95f1d0](#)

- And finally we can Add the A record alias of load balancer for this DNS name
- AND we can access this page with ssl certificate

The screenshot shows a web browser at the URL `chandanshankar.tk`. The page has a red header with the text "React With NodeJS". Below the header, there is a "Create User" section with input fields for "First Name", "Last Name", and "Email", and a "Create" button. To the right of the form is a green box labeled "Users Created" with the number "12" in red and a "Get all Users" button. Below these elements is a table titled "Users" with columns "User Id", "Firstname", "Lastname", and "Email". The table contains three rows of data.

User Id	Firstname	Lastname	Email
1	first1	last1	abc@gmail.com
2	first2	last2	abc@gmail.com
3	first3	last3	abc@gmail.com

- And this page we can see tha secure SSL to chandanshankar.tk Domain

THANK YOU