Lab Certificate Manager

Date	@September 1, 2025
Multi-select	Certificate Manager, EC2, Load Balancer, NAT Gateways, Route 53, Route Table, Security Group, Subnet, Target Group, VPC
Status	Done

Networking

- Create a VPC with CIDR (`10.0.0.0/16`)
- Create 4 Subnets
  - Public Subnet 1 (`10.0.0.0/24`) availability zone `1a`
  - Public Subnet 2 (`10.0.1.0/24`) availability zone `1b`
  - Private Subnet 1 (`10.0.2.0/24`) availability zone `1a`
  - Private Subnet 2 (`10.0.3.0/24`) availability zone `1b`
- Create Internet Gateway (`dev-ig`)
- Create NAT Gateway (`dev-nat`)
- Create a Security Group (`dev-sg`) with rules to allow `SSH` , `HTTP` , and `HTTPS` .

- Create Route Table
  - **Public RT** (associated with `dev-ig`, `Public Subnet 1`, and `Public Subnet 2`)
  - **Private RT** (associated with `dev-nat`, `Private Subnet 1`, and `Private Subnet 2`)

Instance Creation

- Launch Instance `Bastion Host` in `Public Subnet 1` and enable a `Auto-assign public IP`
  - Connect to this `Bastion Host` and insert the private key inside the instance:

```
vi key.pem
chmod 400 key.pem
```

- Launch another instance `Private-EC2` in `Private Subnet 1` and disable `Auto-assign public IP`.
  - Connect to the `Private-EC2` using the private key and `private-ip`:

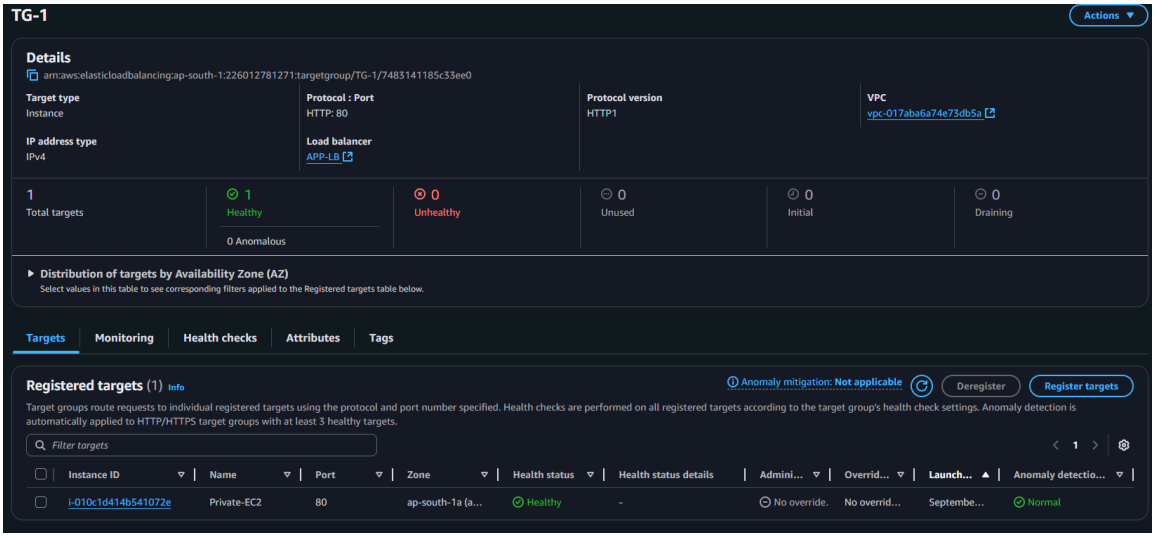
```
ssh -i private-key ec2-user private-ip
```

  - Install `nginx` server on this private server:

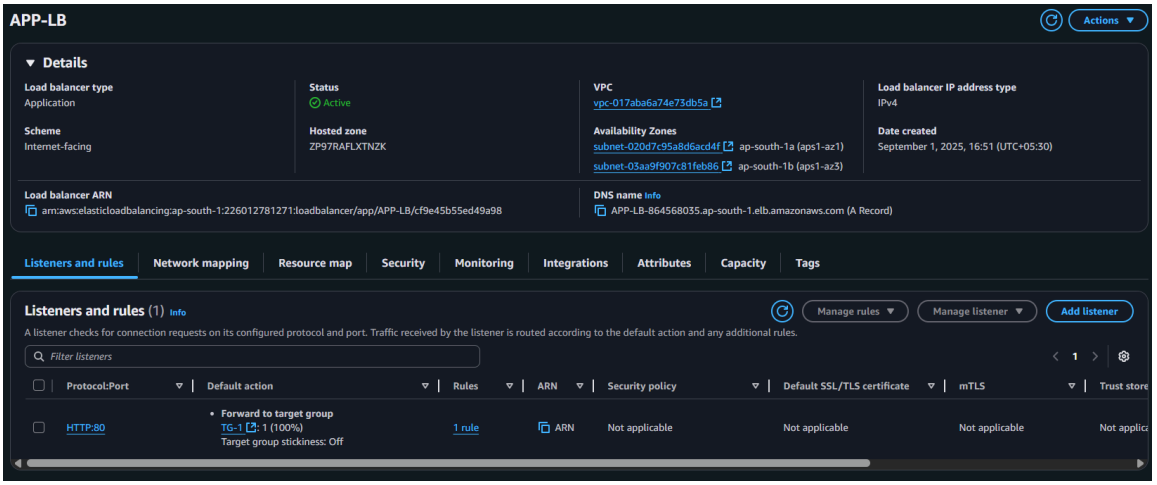
```
sudo su -
yum install nginx -y
systemctl start nginx
systemctl status nginx
```

Load Balancer

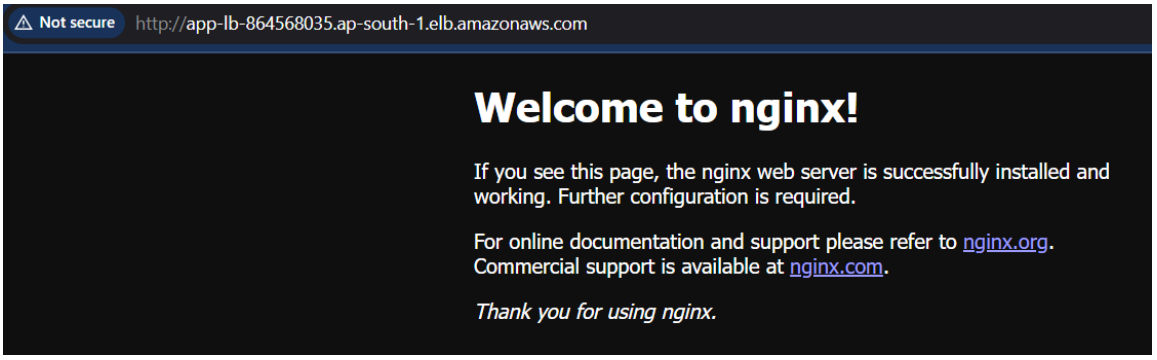
- Create Target Group `TG-1`



- Create Load Balancer



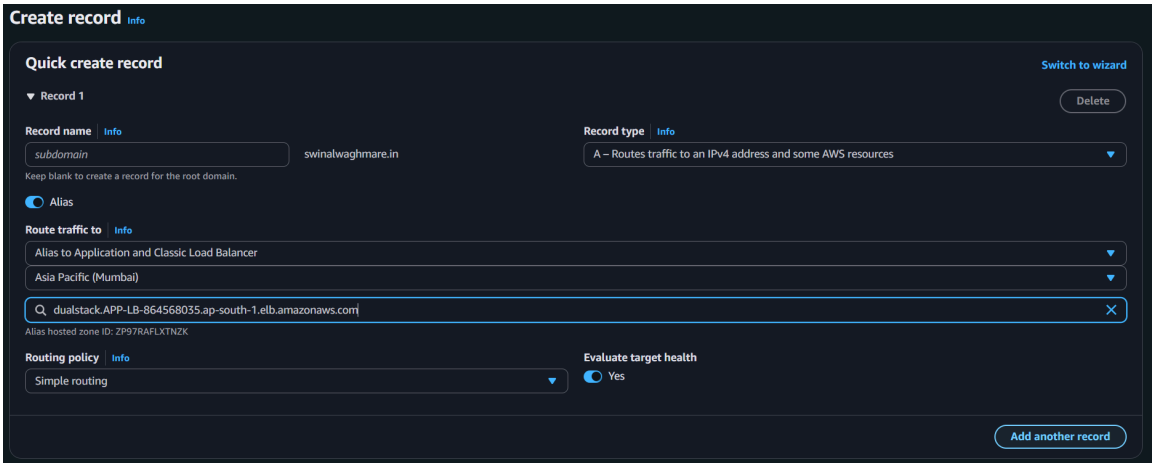
- Access the Load Balancer DNS using `http`.



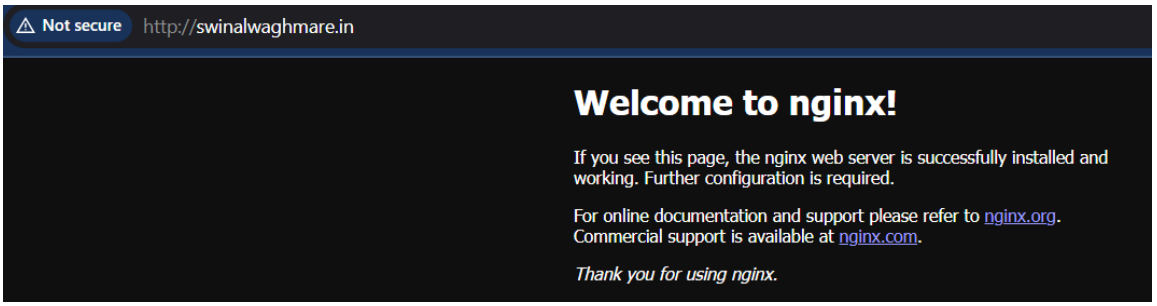
Accessing the Load Balancer Using Domain

- Create an `A` Record (alias):

- Route traffic to **Application and Classic Load Balancer**
- Region: **Asia Pacific (Mumbai)**
- Endpoint: Select the Load Balancer
- Routing Policy: **Simple routing**

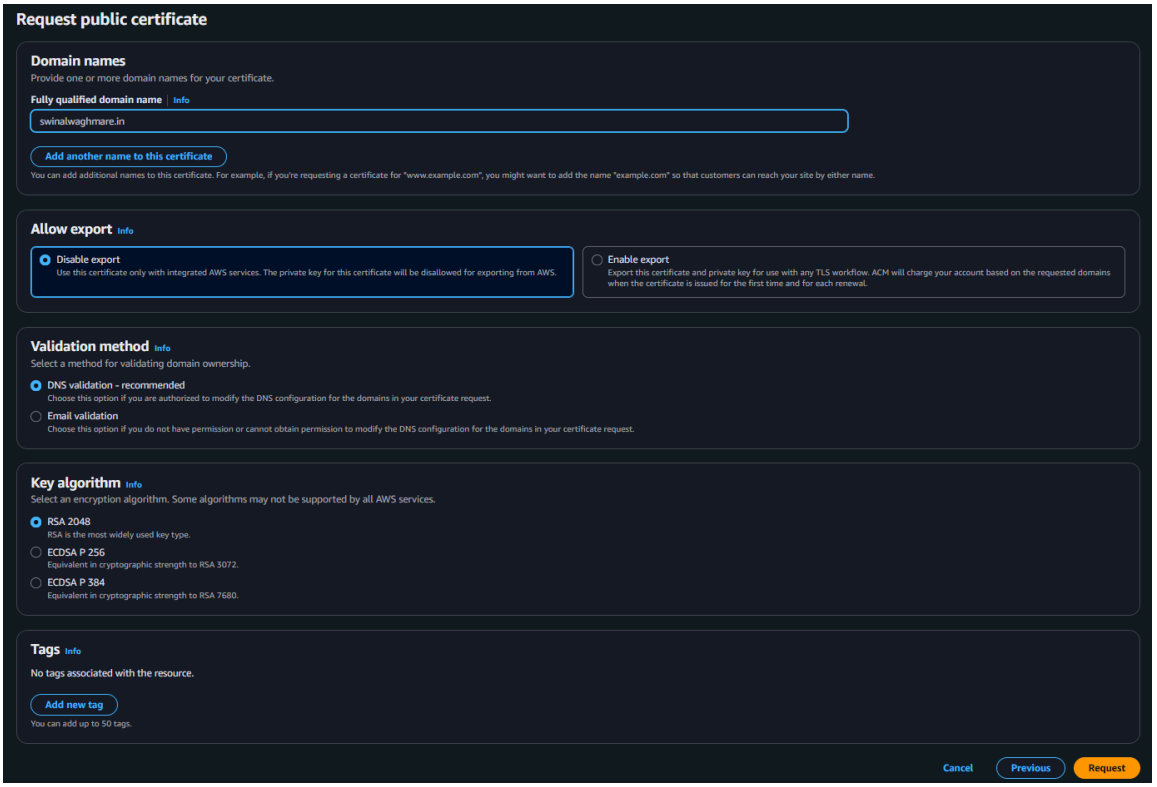


- Access the Load Balancer using the domain name `http://swinalwaghmare.in`.

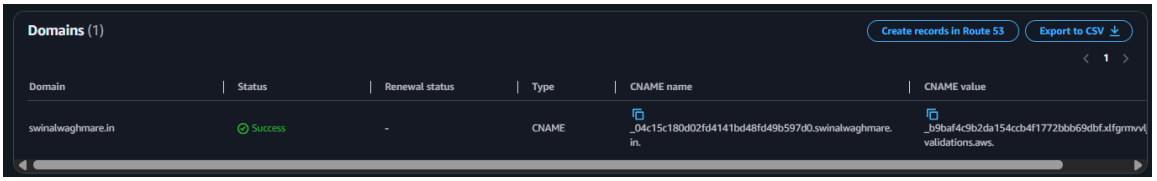


Creation of Certificate

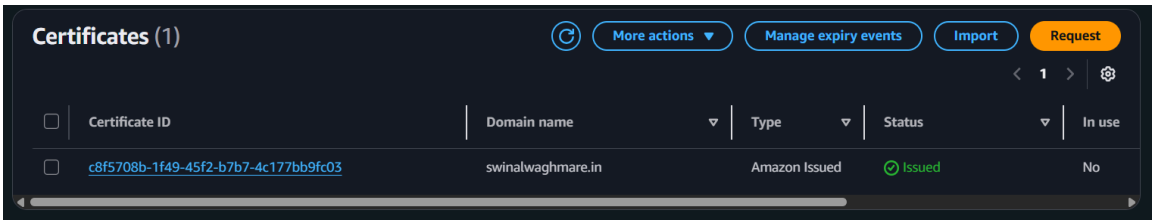
- Request a certificate:
  - Type: `Request a public certificate`
  - Domain name (e.g., `swinalwaghmare.in`)
  - Validation Method: `DNS Validation`
  - Key Algorithm: `RSA 2048`



- Create the required `CNAME record` in Route 53 for Certificate Manager.

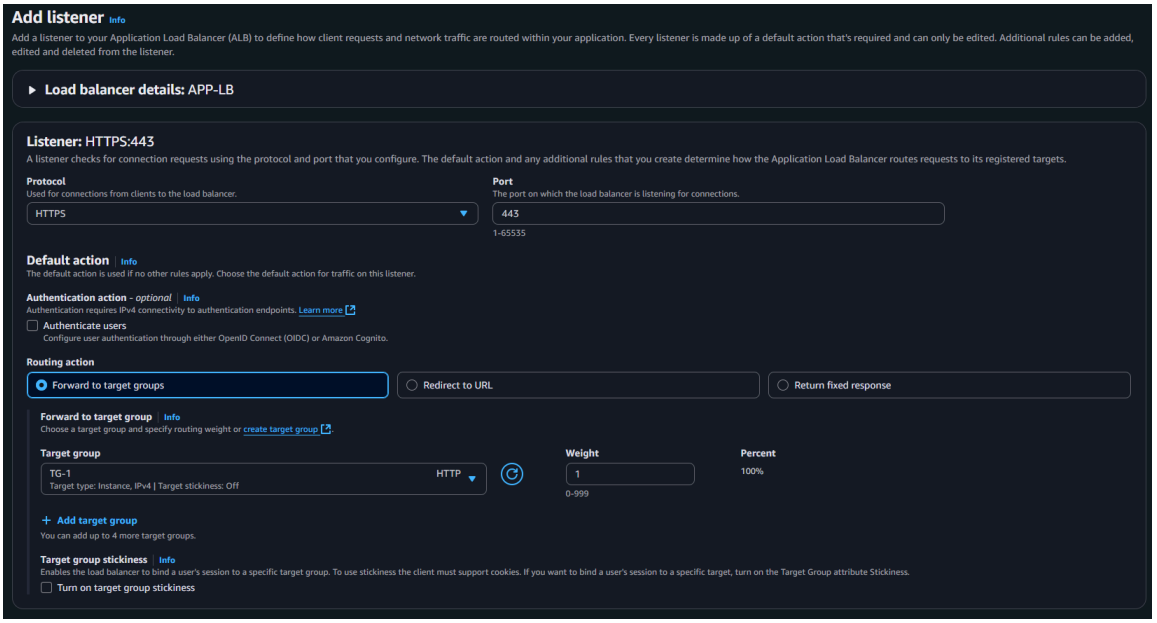


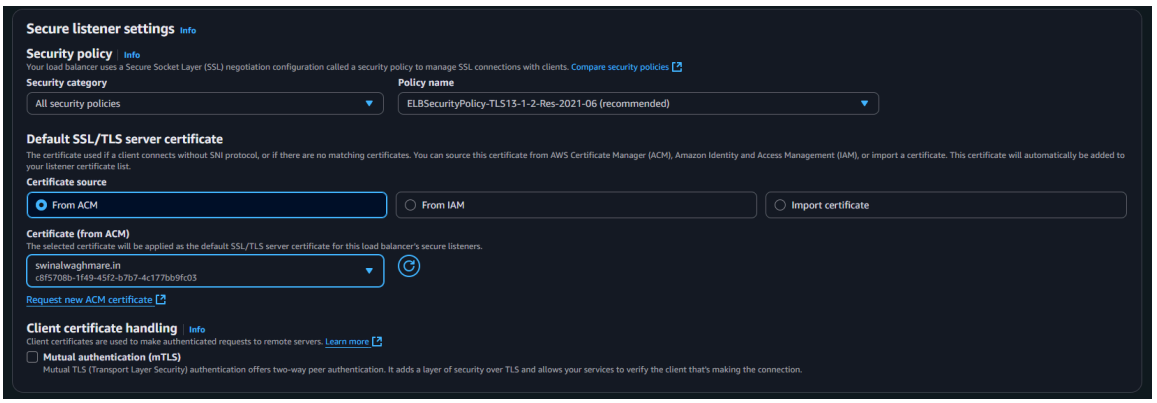
- Once validated, the certificate will be issued.



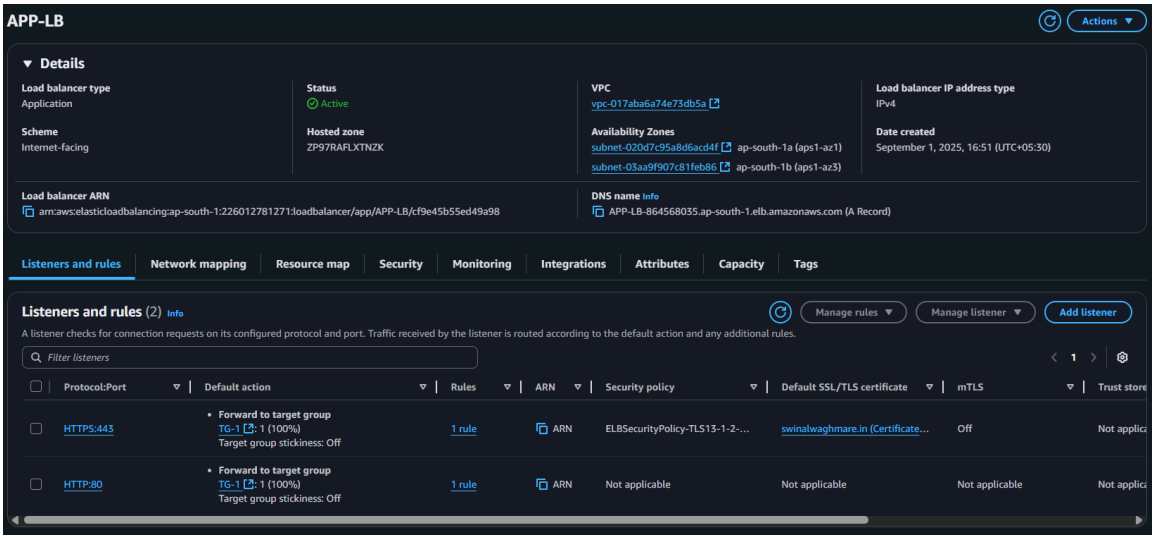
Adding an HTTPS Listener to the Load Balancer

- Add a listener for `HTTPS` and attach the certificate.

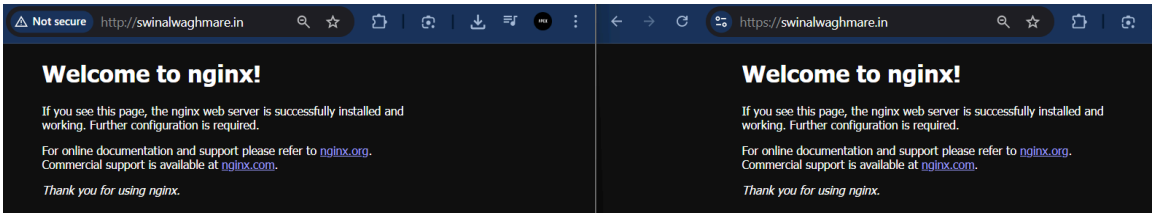




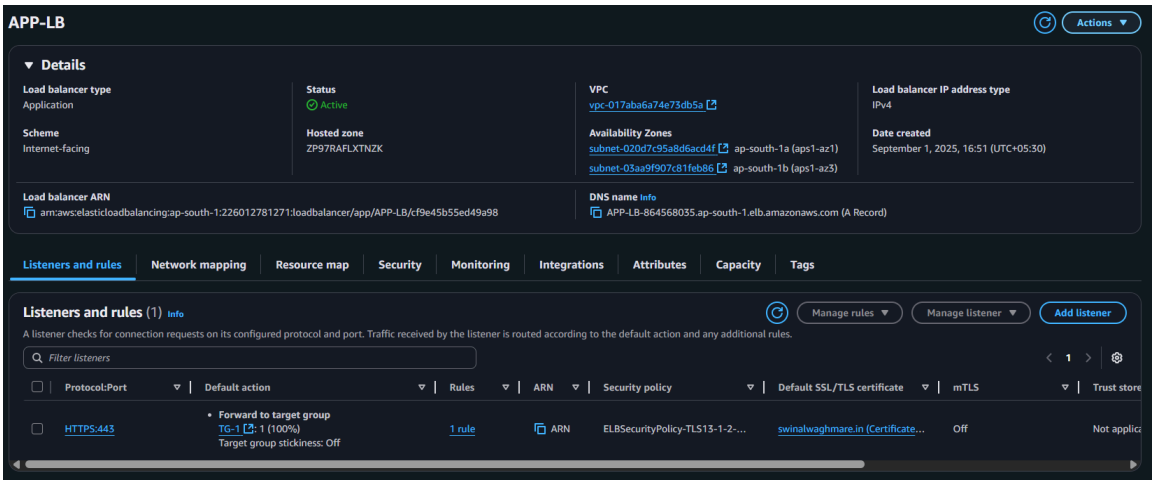
- Now the Load Balancer has two listeners: `HTTP` and `HTTPS`.

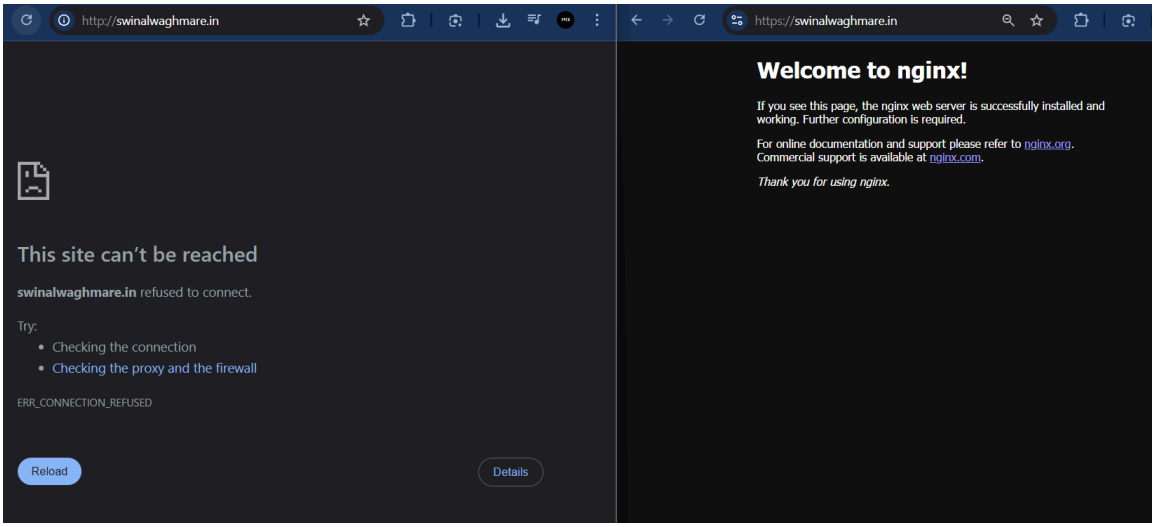


- Since we already have a record in Route 53 for the domain, we can access the domain using both `http` and `https`.



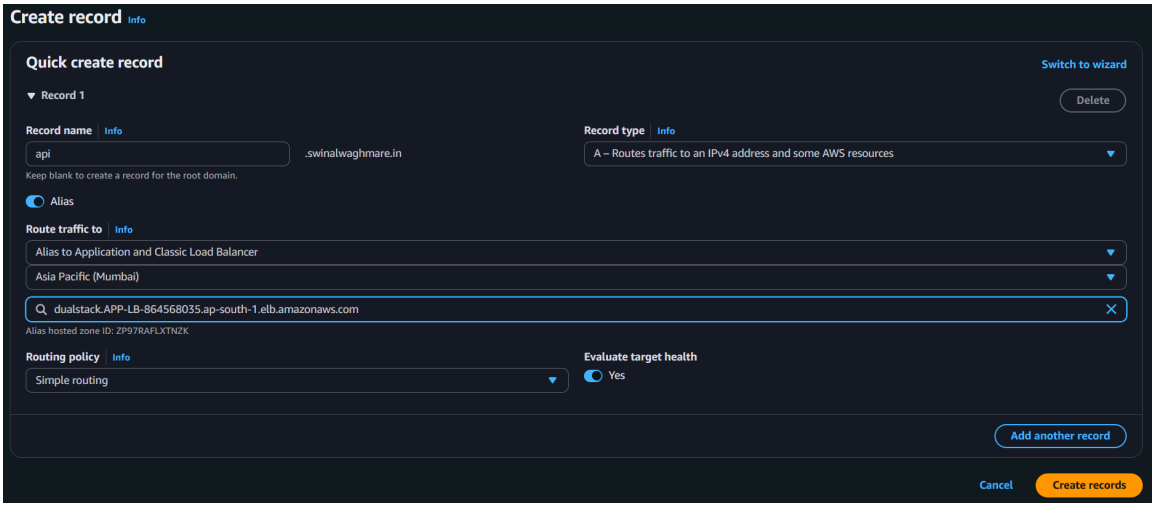
- If we delete one of the listeners (e.g., `HTTP`), the domain will only be accessible via the remaining protocol (`HTTPS`).



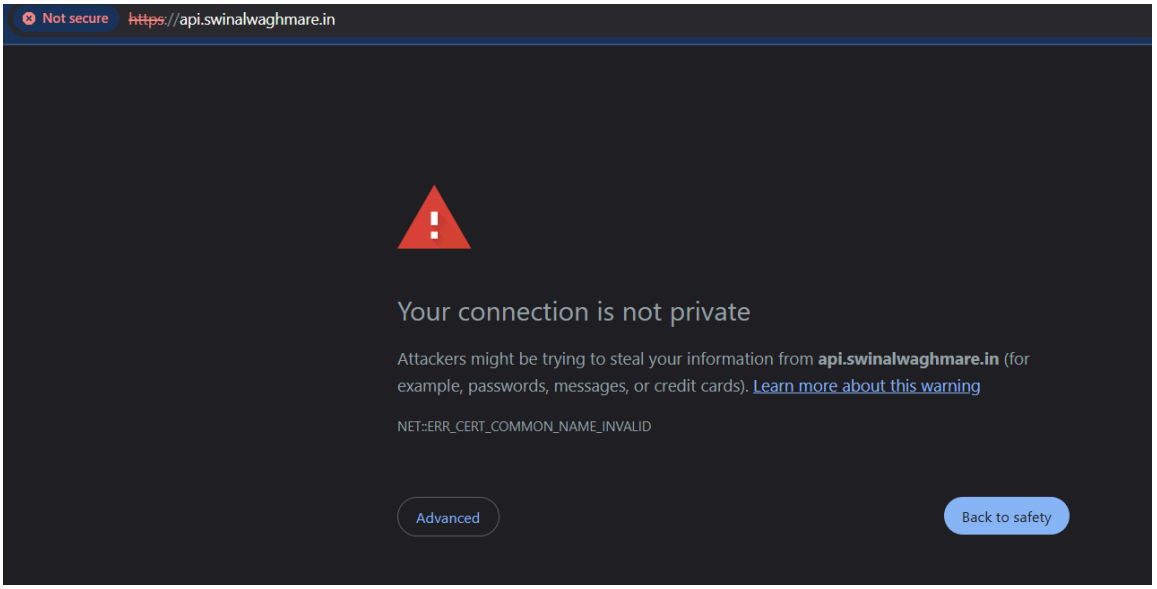


Accessing Through Sub Domain

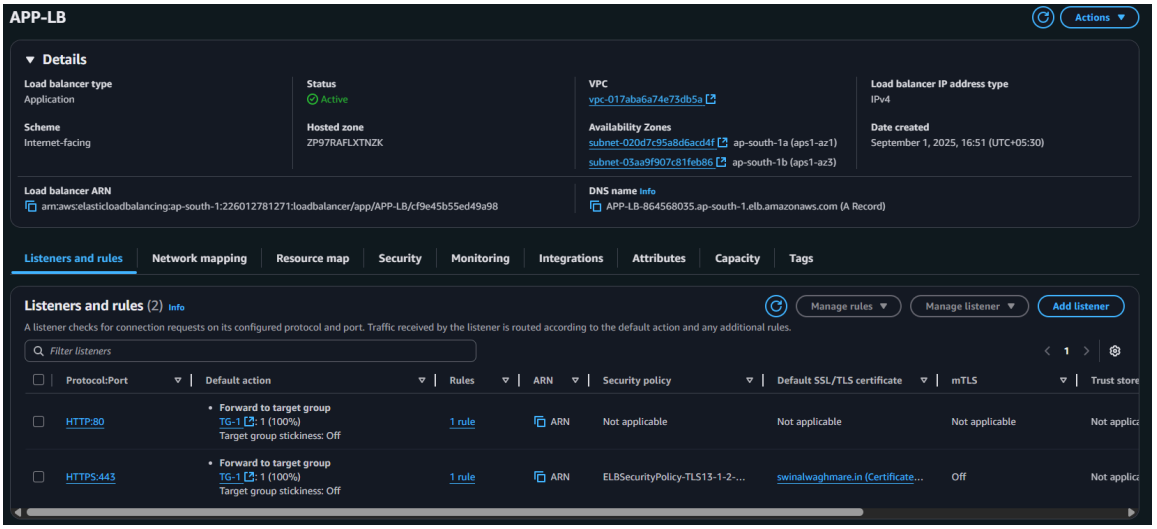
- Create a subdomain record: `api.swinalwaghmare.in`.



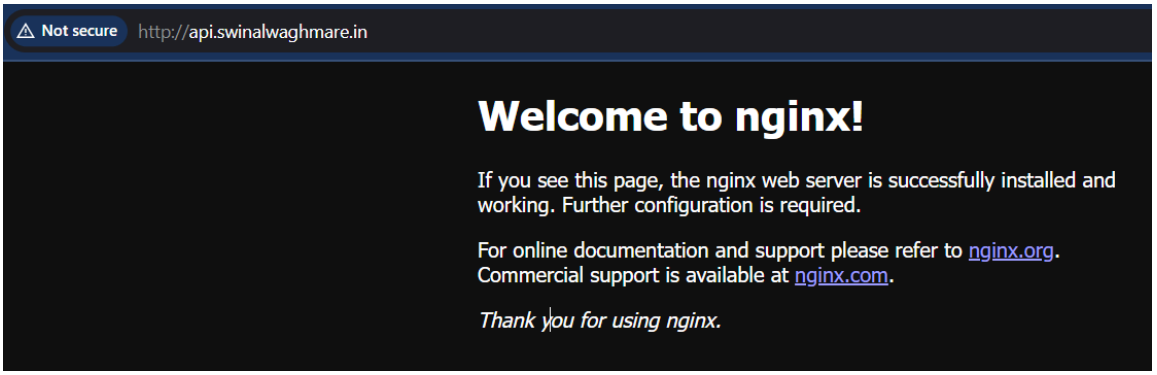
- Since we only have an `HTTPS` listener, this subdomain cannot be accessed via `https` because the certificate is mapped only to `swinalwaghmare.in`.



- To allow access via `http`, add an `HTTP listener` to the Load Balancer.

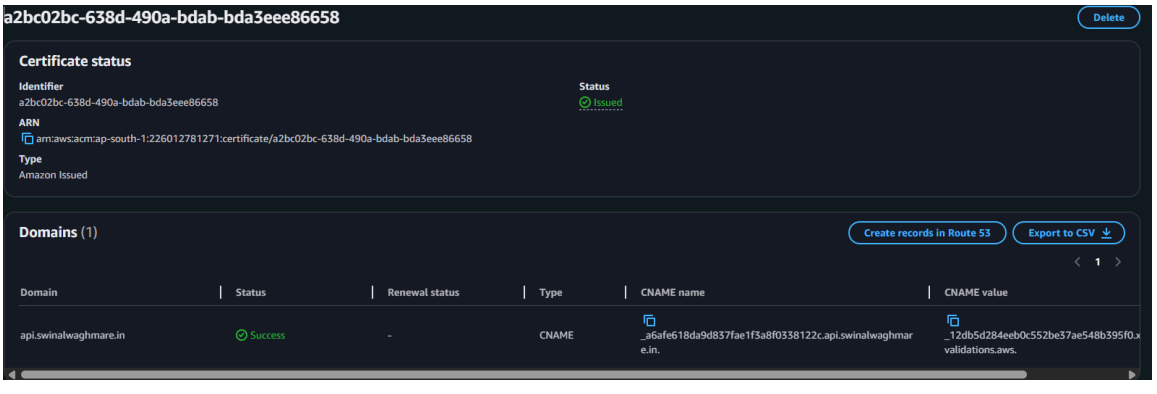


- After adding the listener, the subdomain can be accessed using `http`.

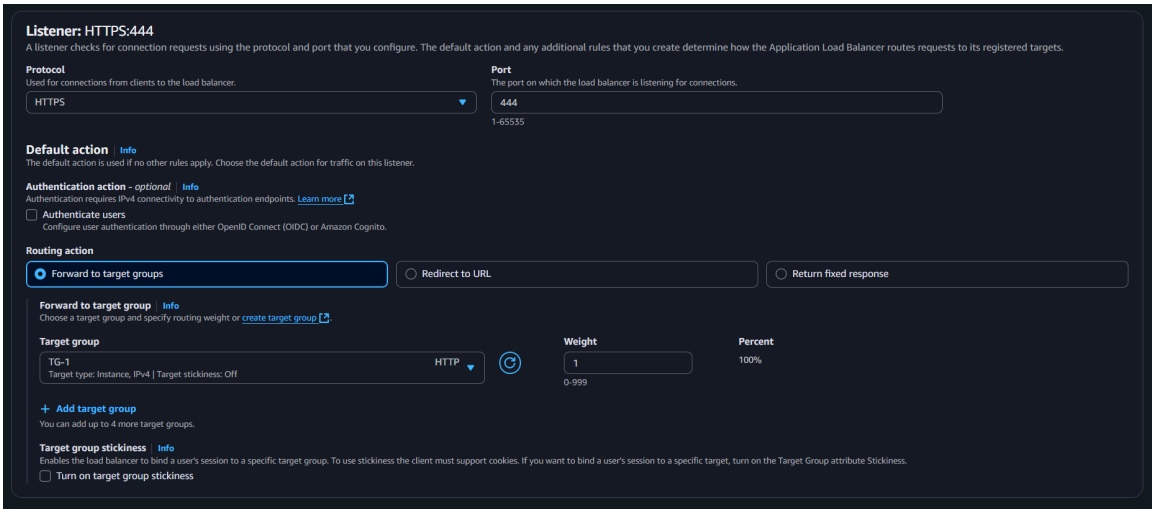


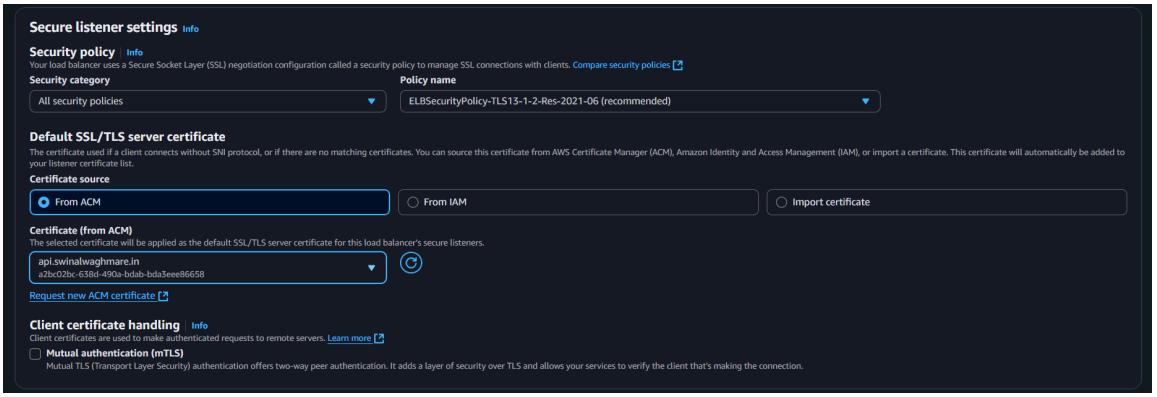
Accessing Subdomain via HTTPS

- Create a certificate for the subdomain.

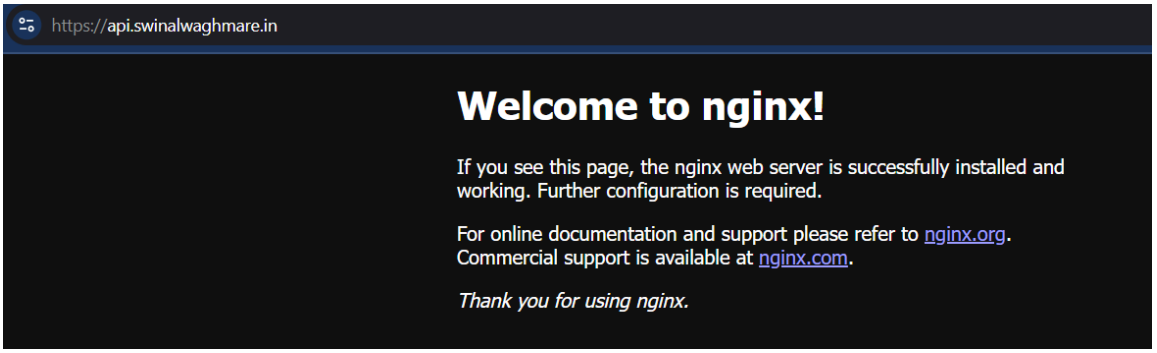


- Add an `HTTPS` listener for the subdomain in the Load Balancer.



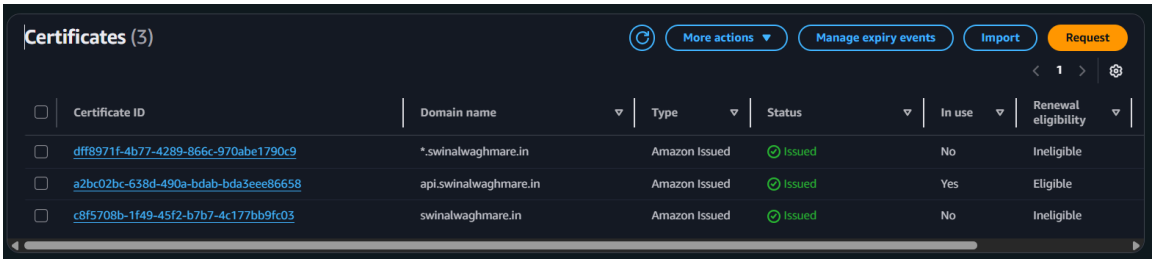


- Now the subdomain `api.swinalwaghmare.in` can be accessed securely via `https`.

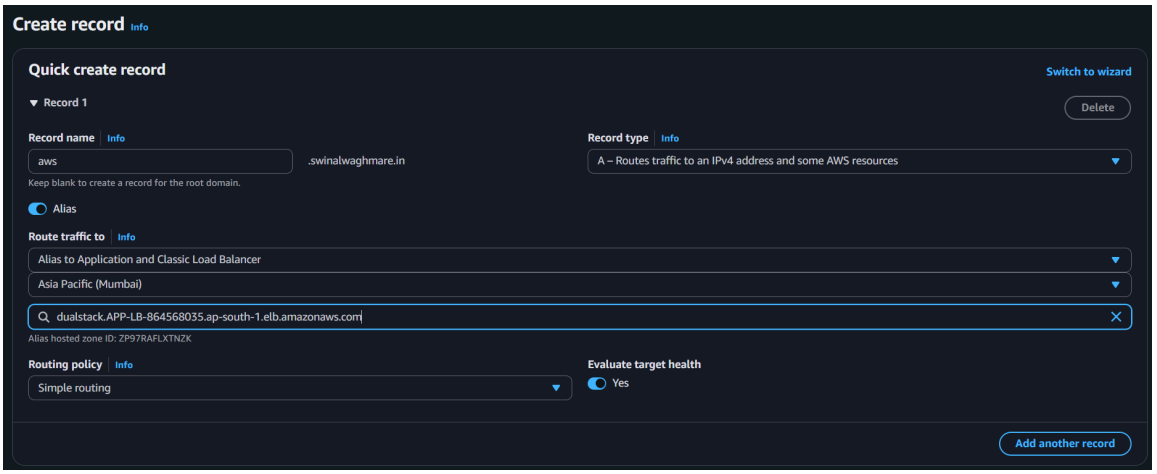


Creating a Wildcard Certificate for All Subdomains

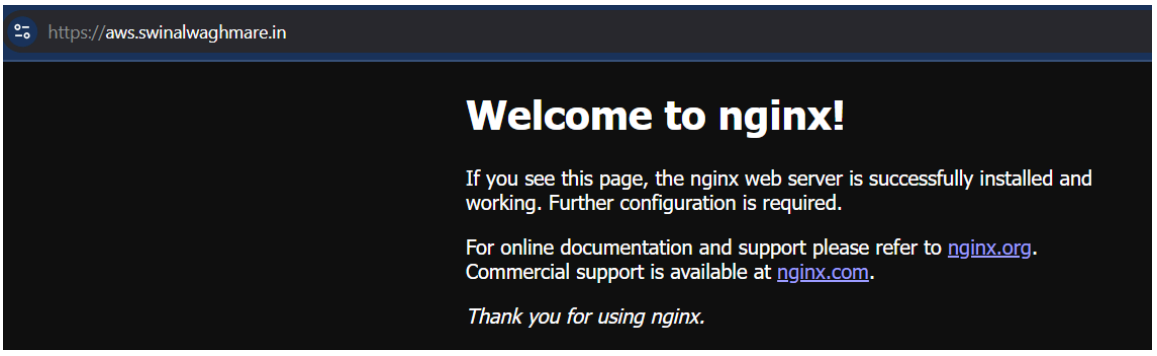
- Create a wildcard certificate `.swinalwaghmare.in`.



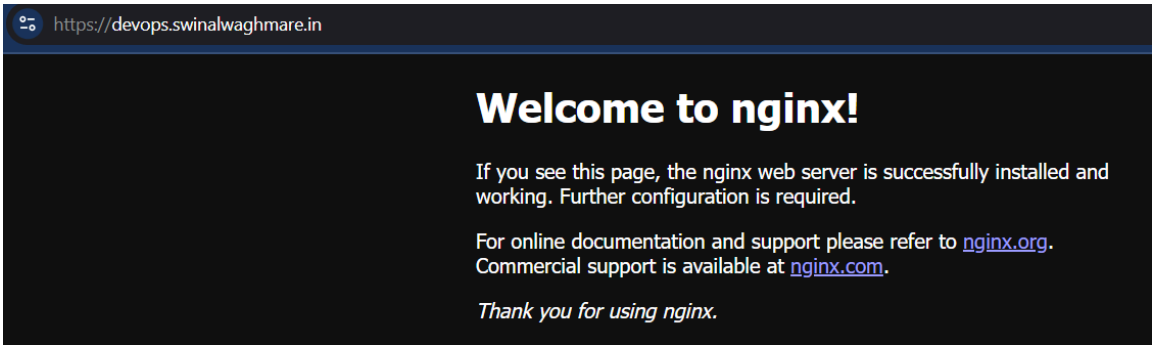
- Create a subdomain record: `aws.swinalwaghmare.in`.



- Add an `HTTPS` listener to the Load Balancer for this subdomain.



- Create another subdomain record for testing: `devops.swinalwaghmare.in`.



- This subdomain can also be accessed via `https` using the same wildcard certificate.