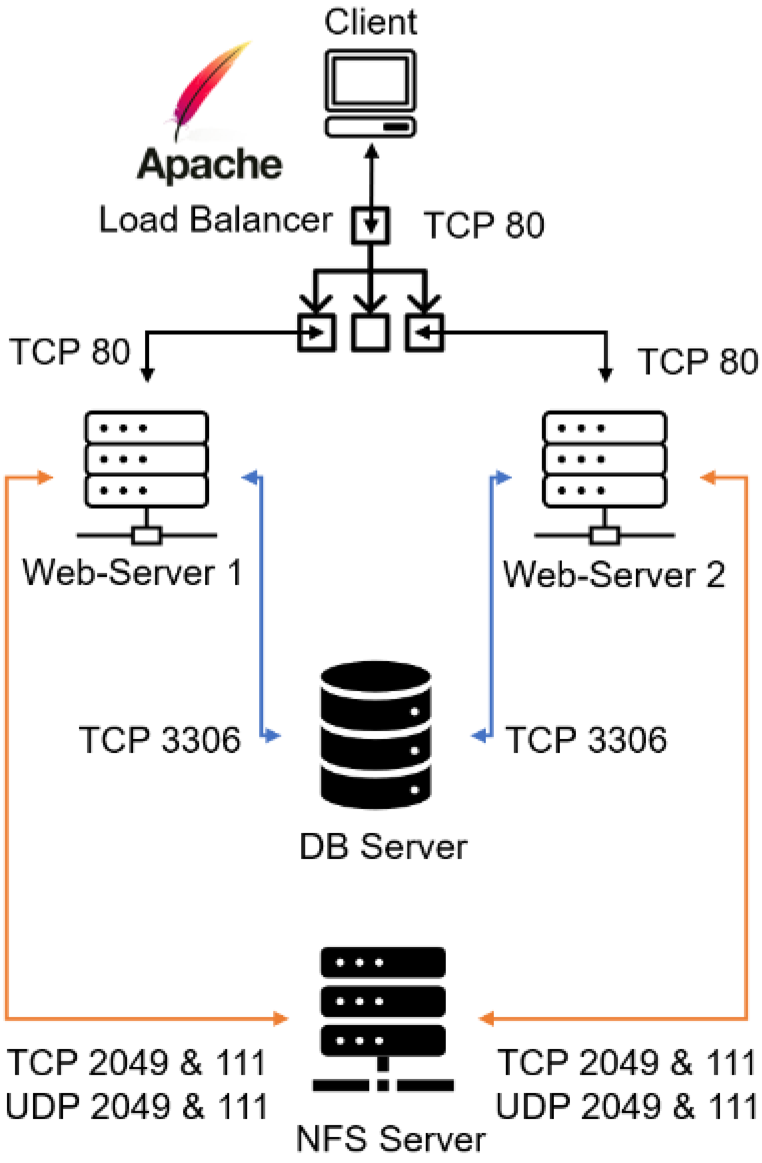
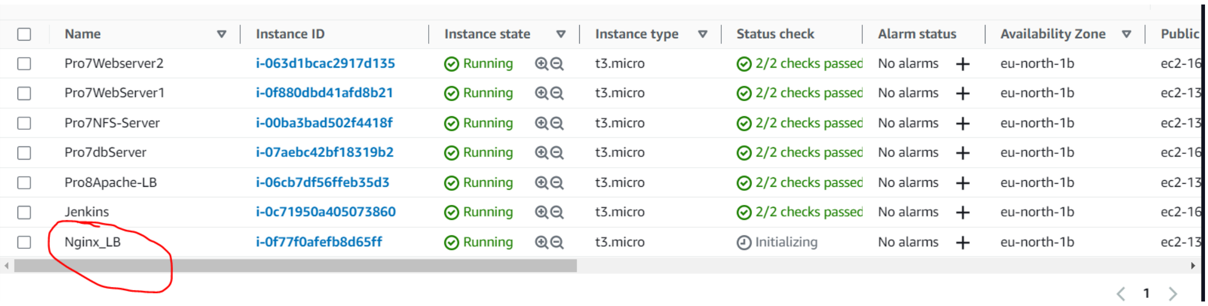
PROJECT 10

LOAD BALANCER SOLUTION WITH NGINX AND SSL/TLS

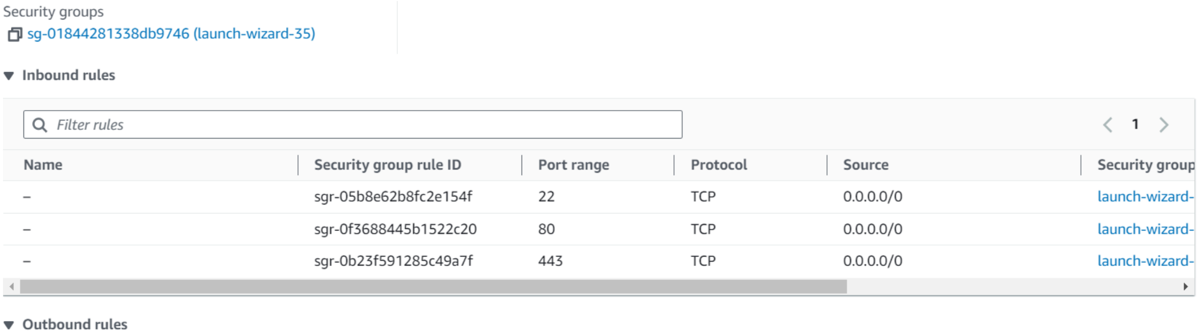


NB: It is also extremely important to ensure that connections to your Web solutions are secure and information is encrypted in transit;

1. Create an EC2 VM based on Ubuntu Server 20.04 LTS and name it `Nginx LB` (do not forget to open TCP port 80 for HTTP connections, also open TCP port **443** – this port is used for secured HTTPS connections)



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
□	Pro7Webserver2	i-063d1bcac2917d135	Running	t3.micro	2/2 checks passed	No alarms +	eu-north-1b	ec2-16
□	Pro7WebServer1	i-0f880dbd41afd8b21	Running	t3.micro	2/2 checks passed	No alarms +	eu-north-1b	ec2-13
□	Pro7NFS-Server	i-00ba3bad502f4418f	Running	t3.micro	2/2 checks passed	No alarms +	eu-north-1b	ec2-13
□	Pro7dbServer	i-07aebc42bf18319b2	Running	t3.micro	2/2 checks passed	No alarms +	eu-north-1b	ec2-16
□	Pro8Apache-LB	i-06cb7df56ffeb35d3	Running	t3.micro	2/2 checks passed	No alarms +	eu-north-1b	ec2-13
□	Jenkins	i-0c71950a405073860	Running	t3.micro	2/2 checks passed	No alarms +	eu-north-1b	ec2-16
□	Nginx_LB	i-0f77f0afefb8d65ff	Running	t3.micro	Initializing	No alarms +	eu-north-1b	ec2-13



Security groups

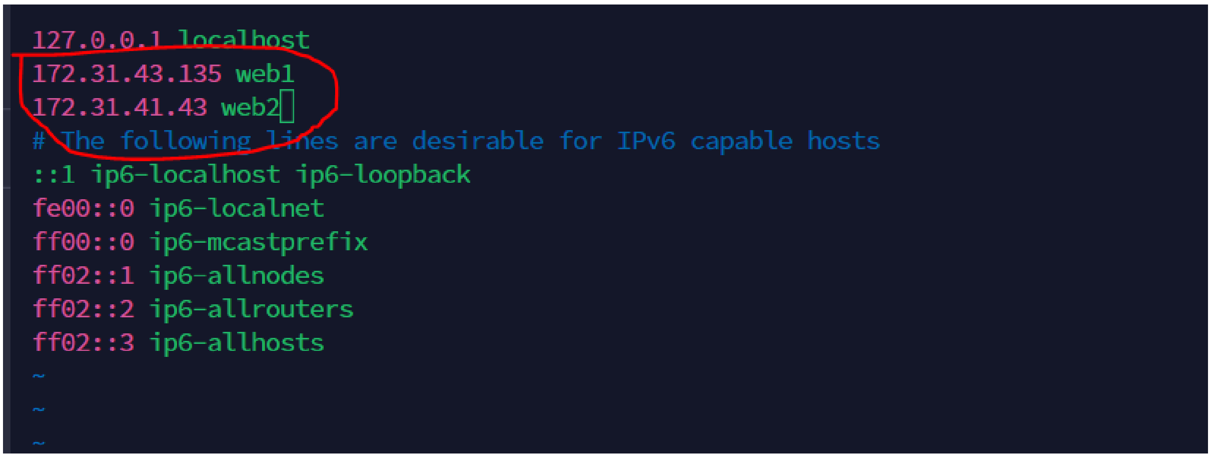
sg-01844281338db9746 (launch-wizard-35)

▼ Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security group
–	sgr-05b8e62b8fc2e154f	22	TCP	0.0.0.0/0	launch-wizard-
–	sgr-0f3688445b1522c20	80	TCP	0.0.0.0/0	launch-wizard-
–	sgr-0b23f591285c49a7f	443	TCP	0.0.0.0/0	launch-wizard-

▼ Outbound rules

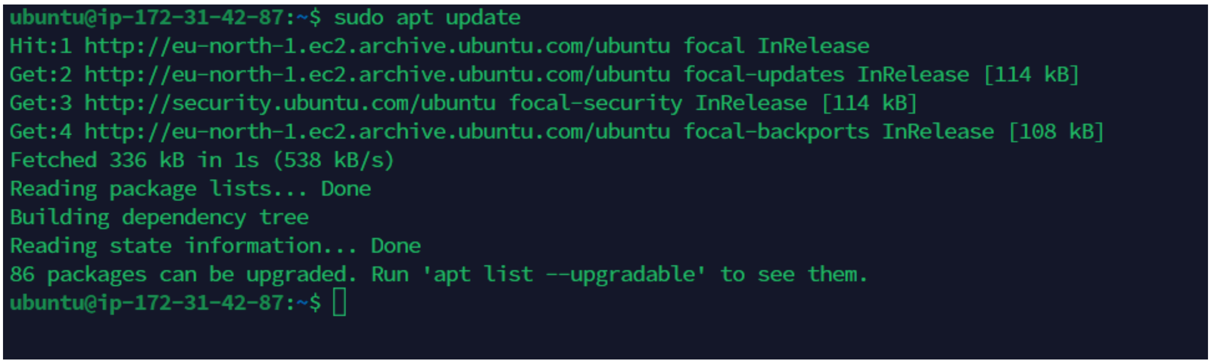
2. Update /etc/hosts file for local DNS with Web Servers' names (e.g. Web1and Web2 and their local IP addresses



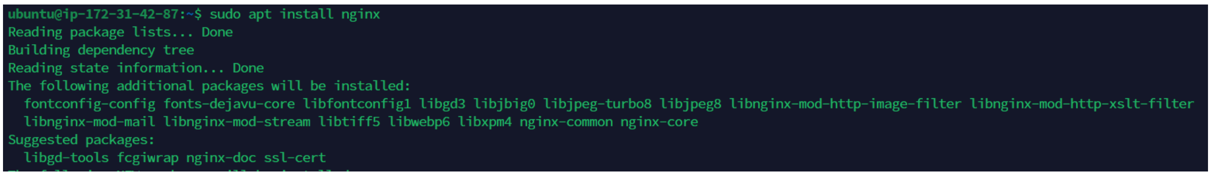
```
127.0.0.1 localhost
172.31.43.135 web1
172.31.41.43 web2
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
~
~
~
```

3. Install and configure Nginx as a load balancer to point traffic to the resolvable DNS names of the web servers

- Update the instance and Install Nginx



```
ubuntu@ip-172-31-42-87:~$ sudo apt update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Fetched 336 kB in 1s (538 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
86 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-42-87:~$
```



```
ubuntu@ip-172-31-42-87:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
```

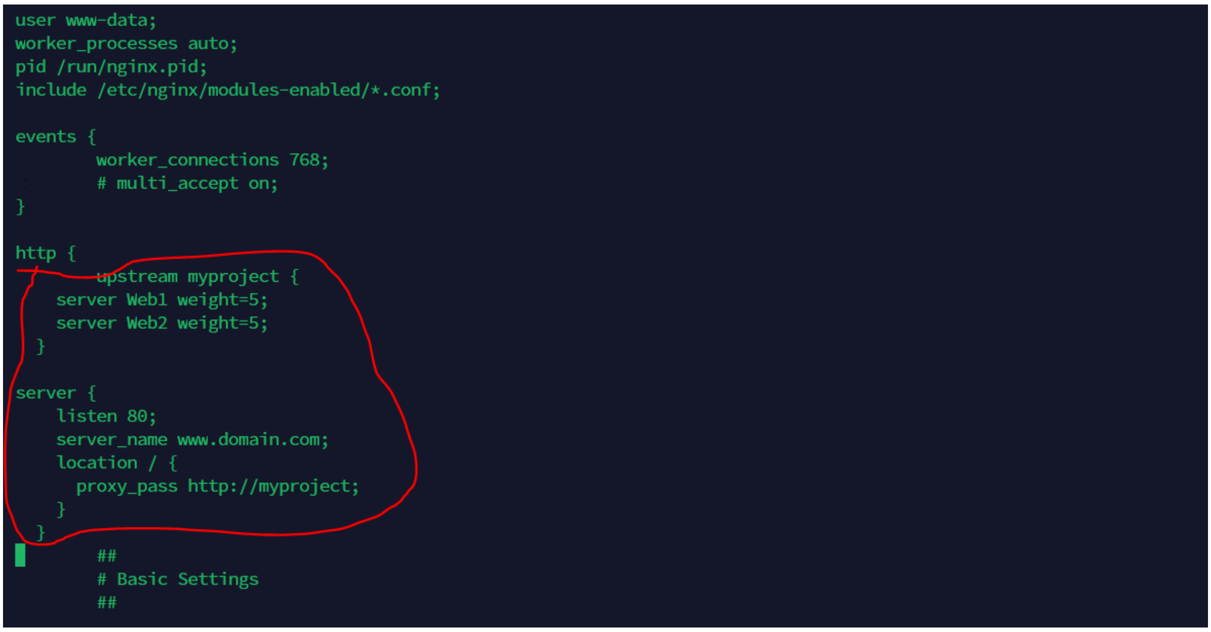
- Configure Nginx LB using Web Servers’ names defined in `/etc/hosts`

Open the default nginx configuration file;

```
sudo vi /etc/nginx/nginx.conf;
```

```
#insert following configuration into http section


 upstream myproject {

    server Web1 weight=5;

    server Web2 weight=5;

  }



server {

    listen 80;

    server_name www.domain.com;

    location / {

      proxy_pass http://myproject;

    }

  }
```

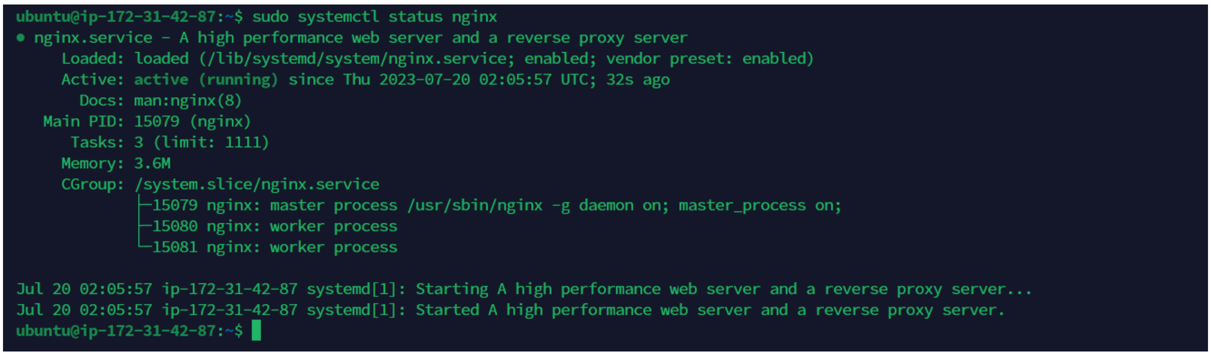
```
#comment out this line

#        include /etc/nginx/sites-enabled/*;
```



```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {
        upstream myproject {
    server Web1 weight=5;
    server Web2 weight=5;
  }

server {
    listen 80;
    server_name www.domain.com;
    location / {
      proxy_pass http://myproject;
    }
  }
        ##
        # Basic Settings
        ##
```

- Restart Nginx and make sure the service is up and running

```
sudo systemctl restart nginx

sudo systemctl status nginx
```



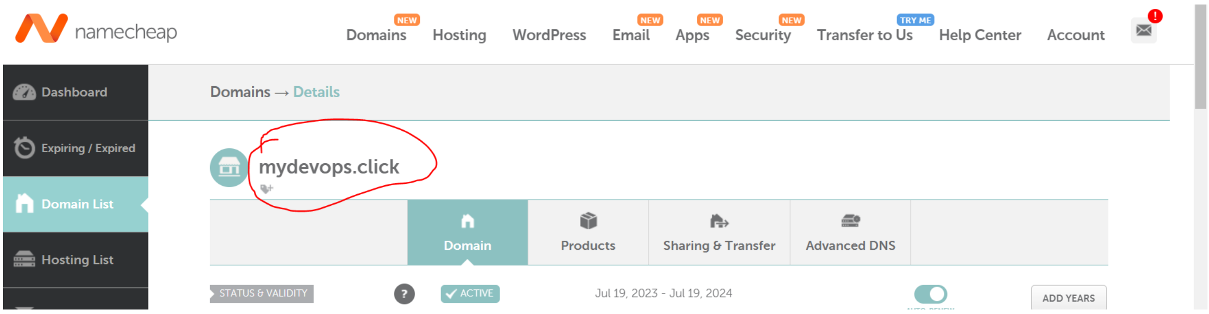
```
ubuntu@ip-172-31-42-87:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-07-20 02:05:57 UTC; 32s ago
       Docs: man:nginx(8)
   Main PID: 15079 (nginx)
      Tasks: 3 (limit: 1111)
     Memory: 3.6M
     CGroup: /system.slice/nginx.service
             ├─15079 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             ├─15080 nginx: worker process
             └─15081 nginx: worker process

Jul 20 02:05:57 ip-172-31-42-87 systemd[1]: Starting A high performance web server and a reverse proxy server...
Jul 20 02:05:57 ip-172-31-42-87 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-42-87:~$
```

REGISTER A NEW DOMAIN NAME AND CONFIGURE SECURED CONNECTION USING SSL/TLS CERTIFICATES

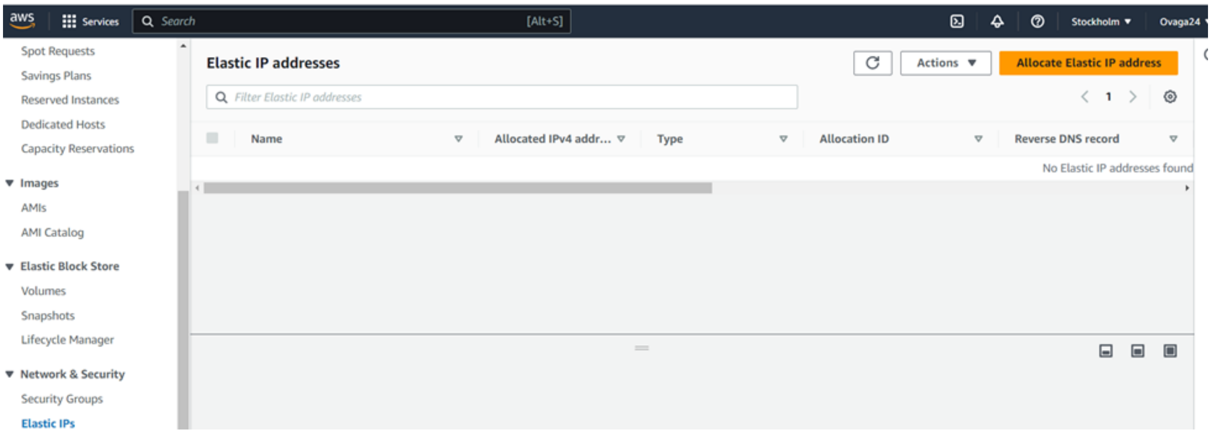
1. Register a new domain name with any registrar

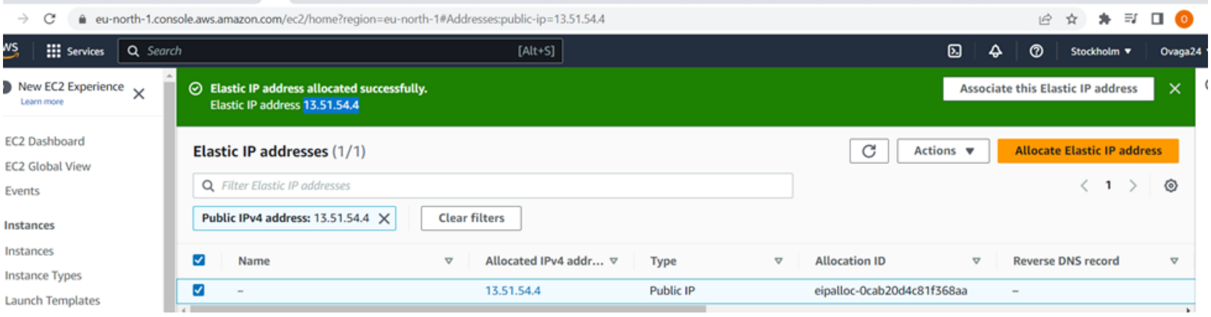
   In my case, I registered with “Namecheap” via this url; https://ap.www.namecheap.com

   - Search for any domain you want.



2. Assign an Elastic IP to Nginx LB server and associated domain name with the Elastic IP.
   - Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   - In the navigation pane, choose Elastic IPs.
   - Select the Elastic IP address to associate and choose Actions, Associate Elastic IP address
   - For Resource type, choose Instance.
   - For instance, choose the instance with which to associate the Elastic IP address. You can also enter text to search for a specific instance.
   - (Optional) For Private IP address, specify a private IP address with which to associate the Elastic IP address.
   - Choose Associate.





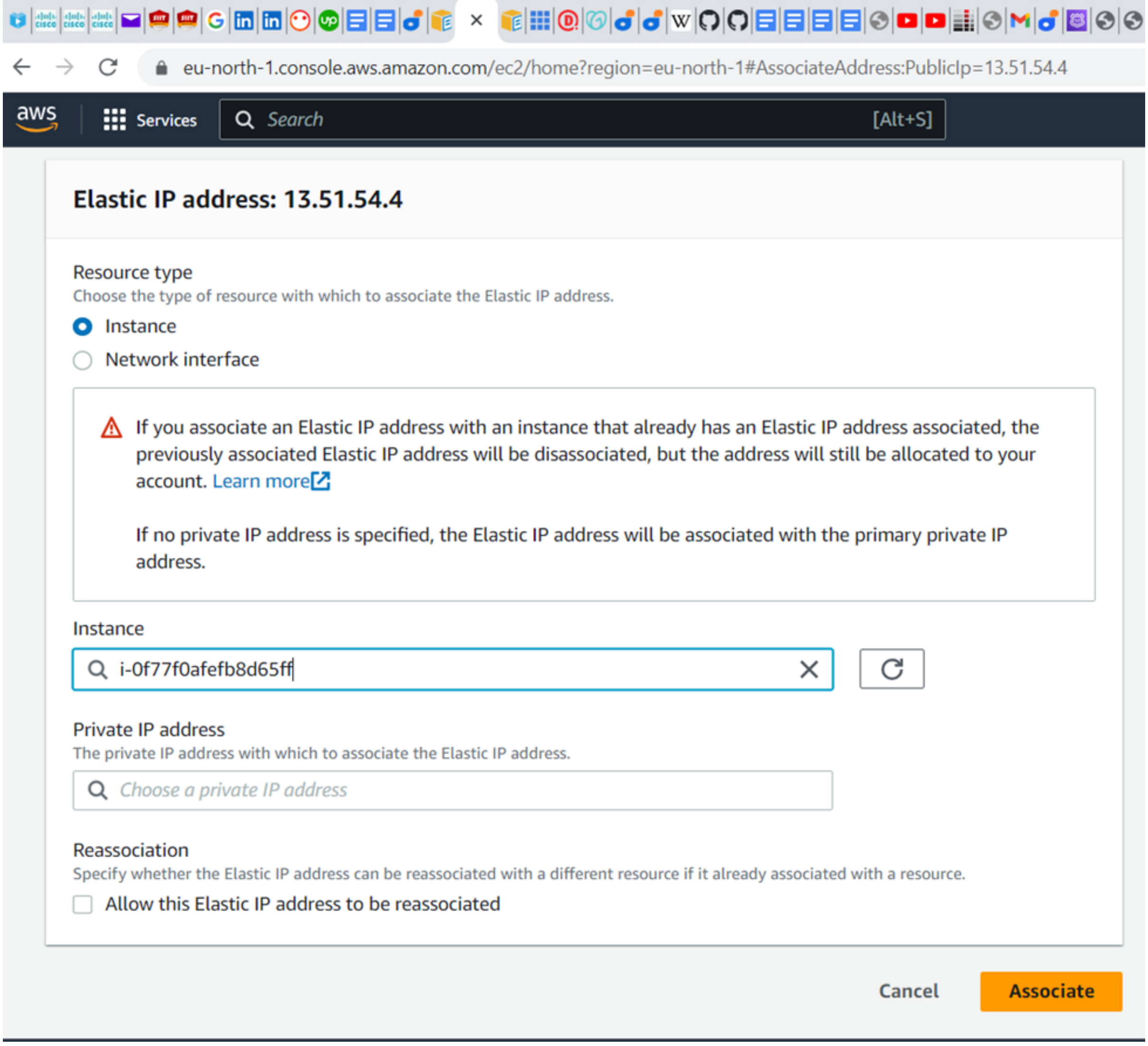
eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Addresses:public-ip=13.51.54.4

Elastic IP address allocated successfully.
Elastic IP address 13.51.54.4

Elastic IP addresses (1/1)

Public IPv4 address: 13.51.54.4

☑	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record
☑	–	13.51.54.4	Public IP	eipalloc-0cab20d4c81f368aa	–



eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#AssociateAddress:PublicIp=13.51.54.4

Elastic IP address: 13.51.54.4

Resource type
Choose the type of resource with which to associate the Elastic IP address.

● Instance
○ Network interface

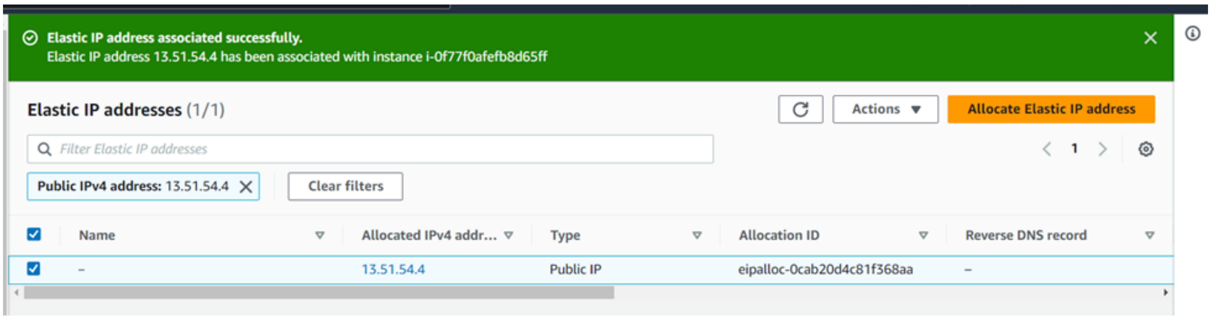
If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance
i-0f77f0afefb8d65ff

Private IP address
The private IP address with which to associate the Elastic IP address.
Choose a private IP address

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
□ Allow this Elastic IP address to be reassociated

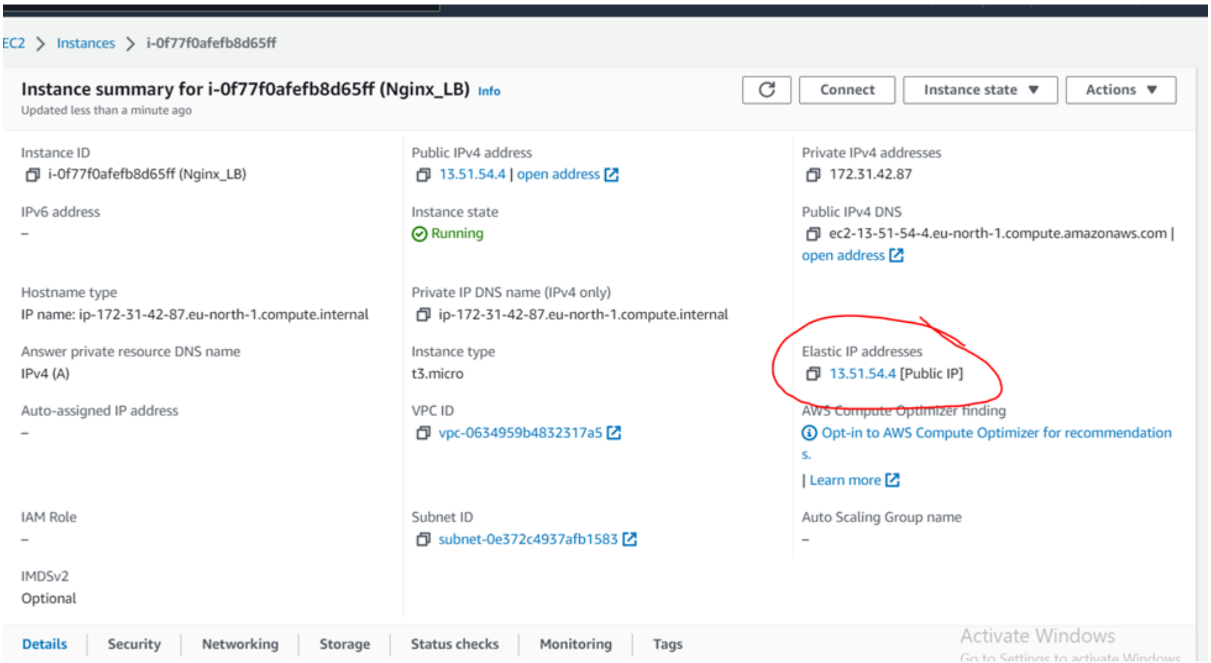


Elastic IP address associated successfully.
Elastic IP address 13.51.54.4 has been associated with instance i-0f77f0afefb8d65ff

Elastic IP addresses (1/1)

Public IPv4 address: 13.51.54.4

☑	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record
☑	–	13.51.54.4	Public IP	eipalloc-0cab20d4c81f368aa	–



Instance summary for i-0f77f0afefb8d65ff (Nginx_LB) Info
Updated less than a minute ago

Instance ID
i-0f77f0afefb8d65ff (Nginx_LB)

Public IPv4 address
13.51.54.4 | open address

Private IPv4 addresses
172.31.42.87

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-13-51-54-4.eu-north-1.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-42-87.eu-north-1.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-42-87.eu-north-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t3.micro

Elastic IP addresses
13.51.54.4 [Public IP]

Auto-assigned IP address
–

VPC ID
vpc-0634959b4832317a5

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

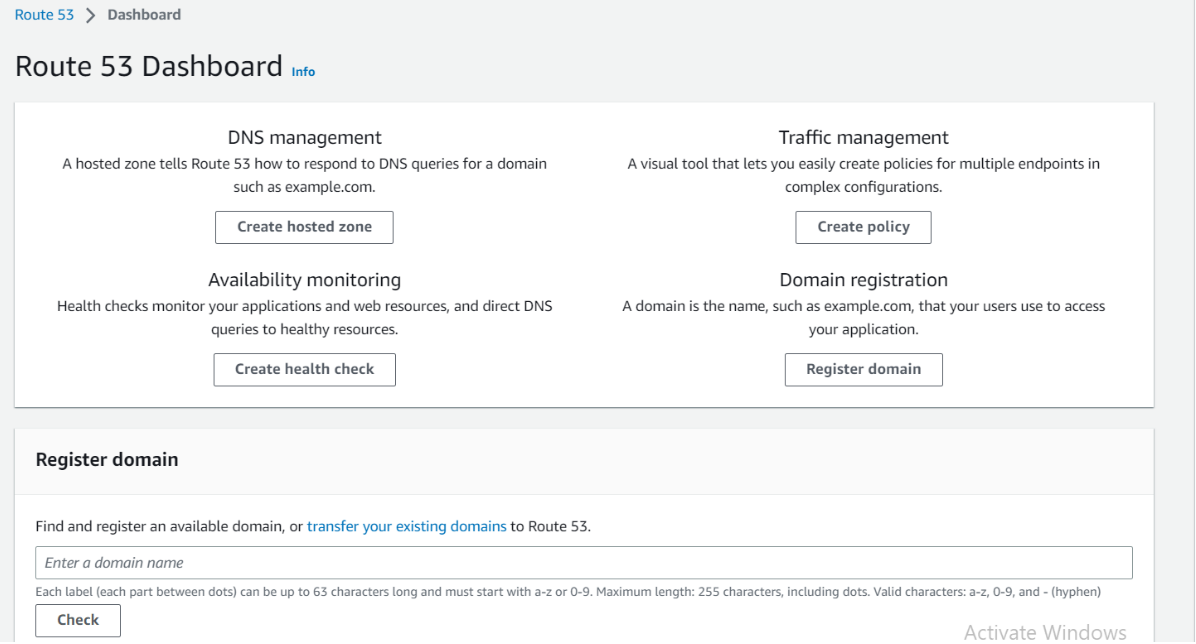
IAM Role
–

Subnet ID
subnet-0e372c4937afb1583

Auto Scaling Group name
–

IMDSv2
Optional

3. Go to your aws account search for Route 53



Route 53 Dashboard Info

DNS management
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

Traffic management
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

Availability monitoring
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

Domain registration
A domain is the name, such as example.com, that your users use to access your application.

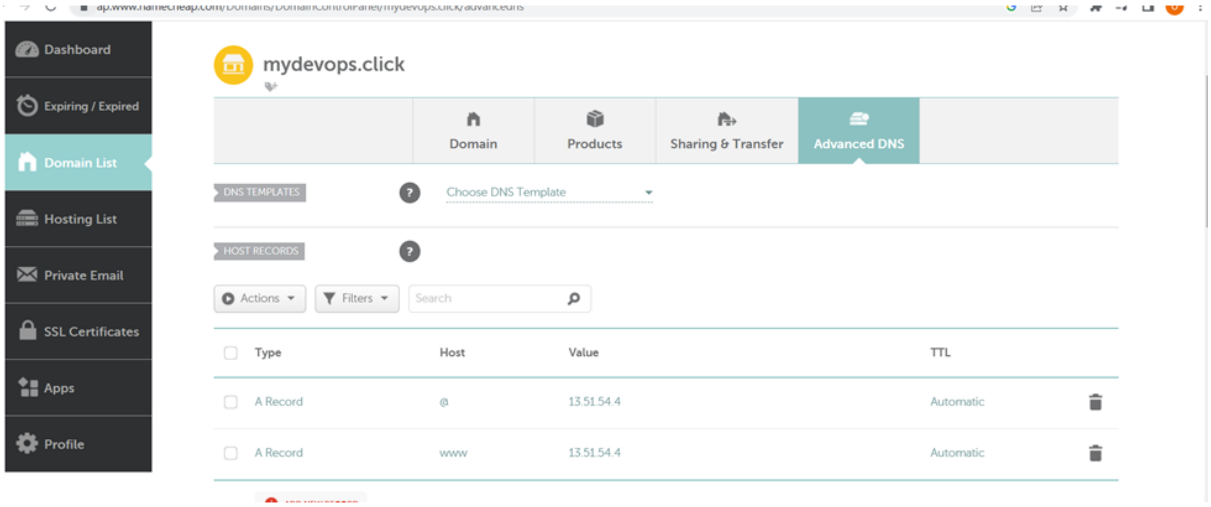
Register domain

Find and register an available domain, or transfer your existing domains to Route 53.

Enter a domain name

Each label (each part between dots) can be up to 63 characters long and must start with a-z or 0-9. Maximum length: 255 characters, including dots. Valid characters: a-z, 0-9, and - (hyphen)

3. Update A record in your registrar to point to Nginx LB using Elastic IP address

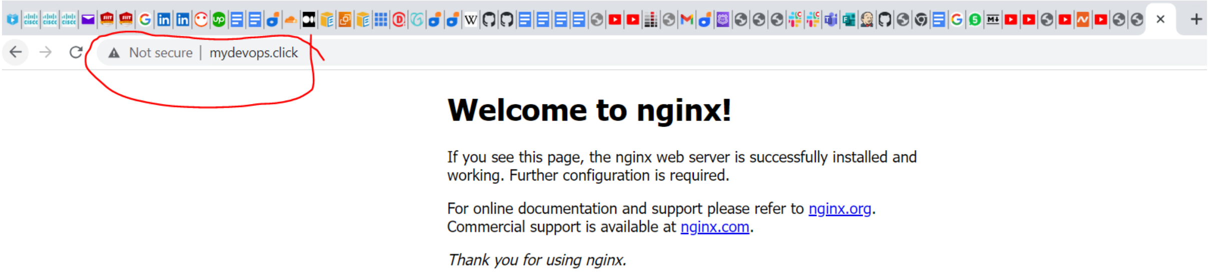


mydevops.click

Type	Host	Value	TTL
A Record	@	13.51.54.4	Automatic
A Record	www	13.51.54.4	Automatic

Check that your Web Servers can be reached from your browser using new domain name using HTTP protocol –

`http://<your-domain-name.com>`



Not secure | mydevops.click

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

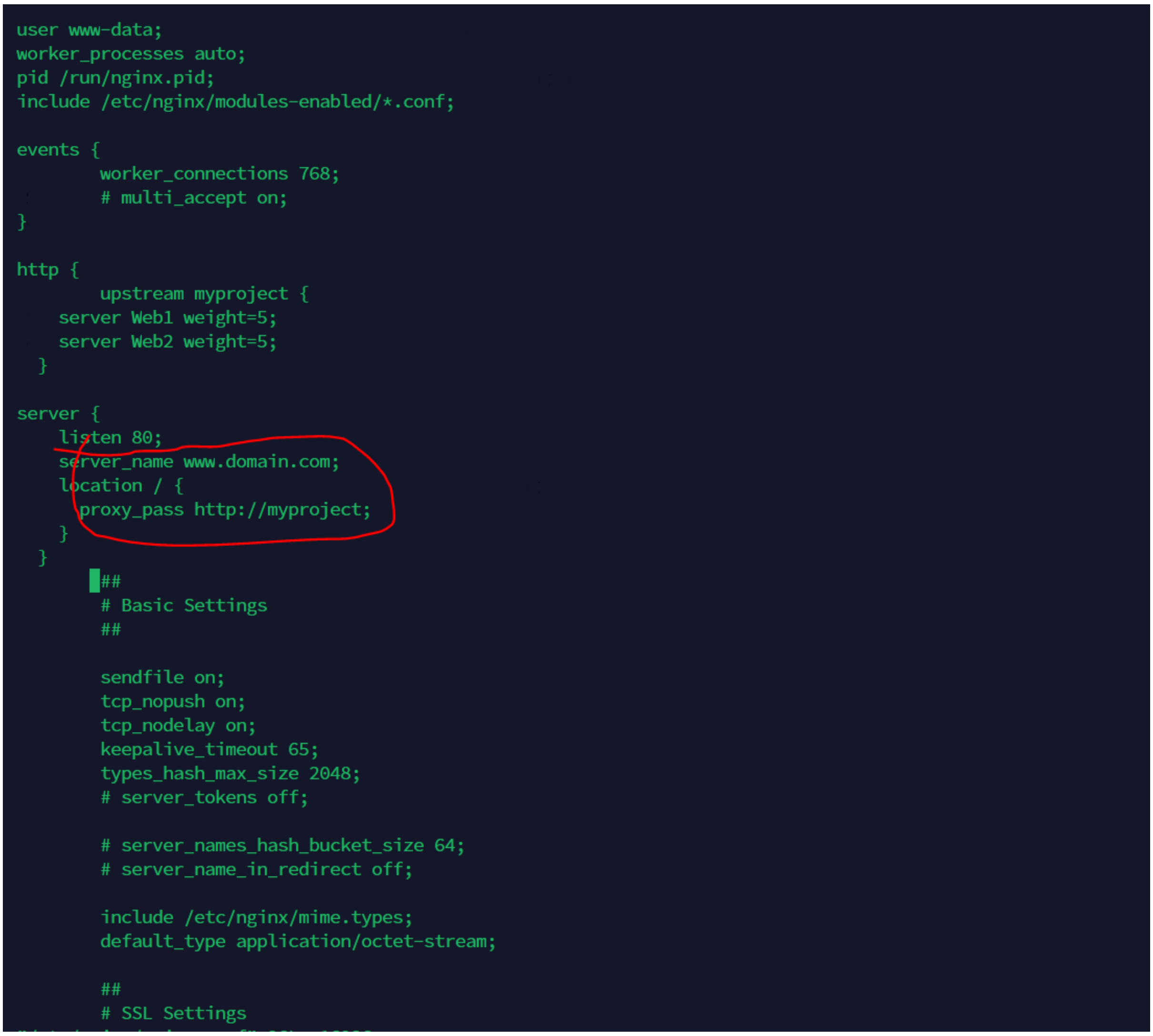
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

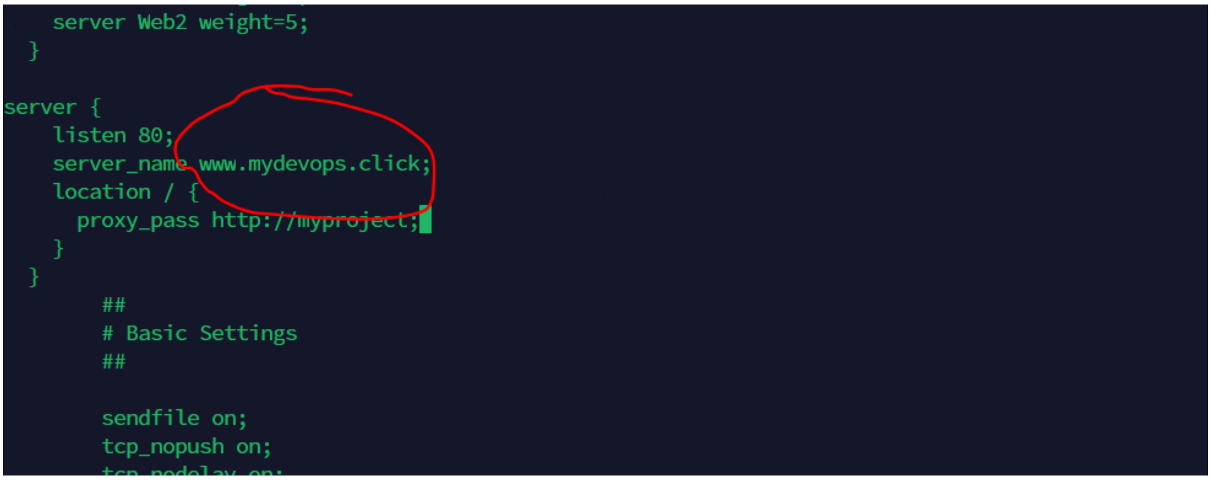
4. Configure Nginx to recognize your new domain name

Update your `nginx.conf` with `server_name www.<your-domain-name.com>` instead of `server_name` www.domain.com

Configure Nginx to recognize the new domain name. This was done by Updating the /etc/nginx/nginx.conf file with



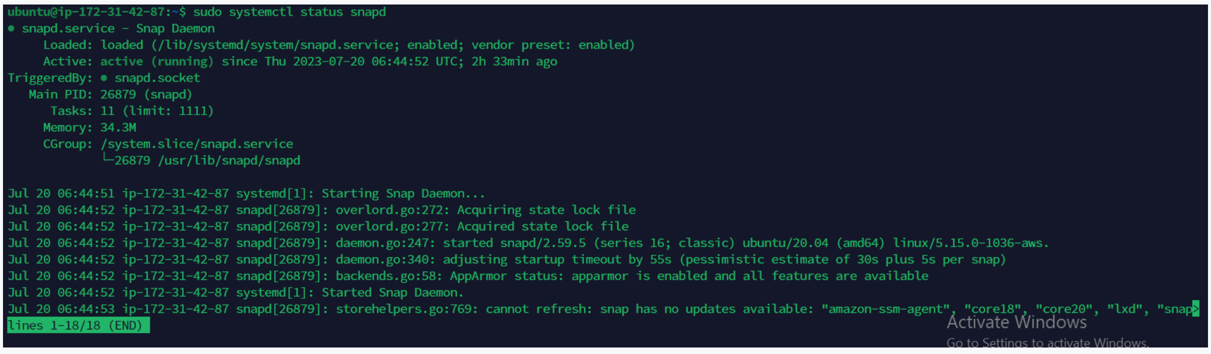
```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {
        upstream myproject {
    server Web1 weight=5;
    server Web2 weight=5;
  }

server {
    listen 80;
    server_name www.domain.com;
    location / {
      proxy_pass http://myproject;
    }
  }
        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # SSL Settings
```



```
    server Web2 weight=5;
  }

server {
    listen 80;
    server_name www.mydevops.click;
    location / {
      proxy_pass http://myproject;
    }
  }
        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
```

5. Install certbot and request for an SSL/TLS certificate for the domain name. N.B: Make sure snapd is running on the server.

    `sudo systemctl status snapd`

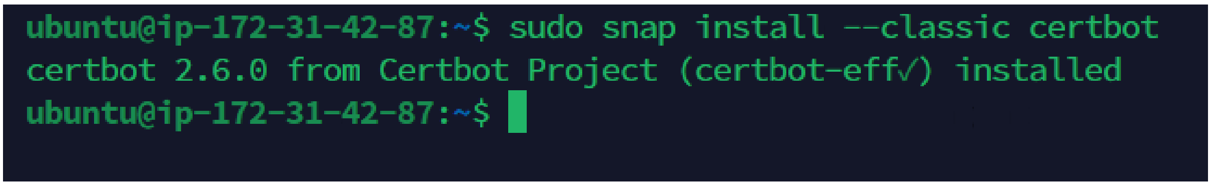


```
ubuntu@ip-172-31-42-87:~$ sudo systemctl status snapd
● snapd.service - Snap Daemon
     Loaded: loaded (/lib/systemd/system/snapd.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-07-20 06:44:52 UTC; 2h 33min ago
TriggeredBy: ● snapd.socket
   Main PID: 26879 (snapd)
      Tasks: 11 (limit: 1111)
     Memory: 34.3M
     CGroup: /system.slice/snapd.service
             └─26879 /usr/lib/snapd/snapd

Jul 20 06:44:51 ip-172-31-42-87 systemd[1]: Starting Snap Daemon...
Jul 20 06:44:52 ip-172-31-42-87 snapd[26879]: overlord.go:272: Acquiring state lock file
Jul 20 06:44:52 ip-172-31-42-87 snapd[26879]: overlord.go:277: Acquired state lock file
Jul 20 06:44:52 ip-172-31-42-87 snapd[26879]: daemon.go:247: started snapd/2.59.5 (series 16; classic) ubuntu/20.04 (amd64) linux/5.15.0-1036-aws.
Jul 20 06:44:52 ip-172-31-42-87 snapd[26879]: daemon.go:340: adjusting startup timeout by 55s (pessimistic estimate of 30s plus 5s per snap)
Jul 20 06:44:52 ip-172-31-42-87 snapd[26879]: backends.go:58: AppArmor status: apparmor is enabled and all features are available
Jul 20 06:44:52 ip-172-31-42-87 systemd[1]: Started Snap Daemon.
Jul 20 06:44:53 ip-172-31-42-87 snapd[26879]: storehelpers.go:769: cannot refresh: snap has no updates available: "amazon-ssm-agent", "core18", "core20", "lxd", "snap>
lines 1-18/18 (END)
```

Activate Windows
Go to Settings to activate Windows.

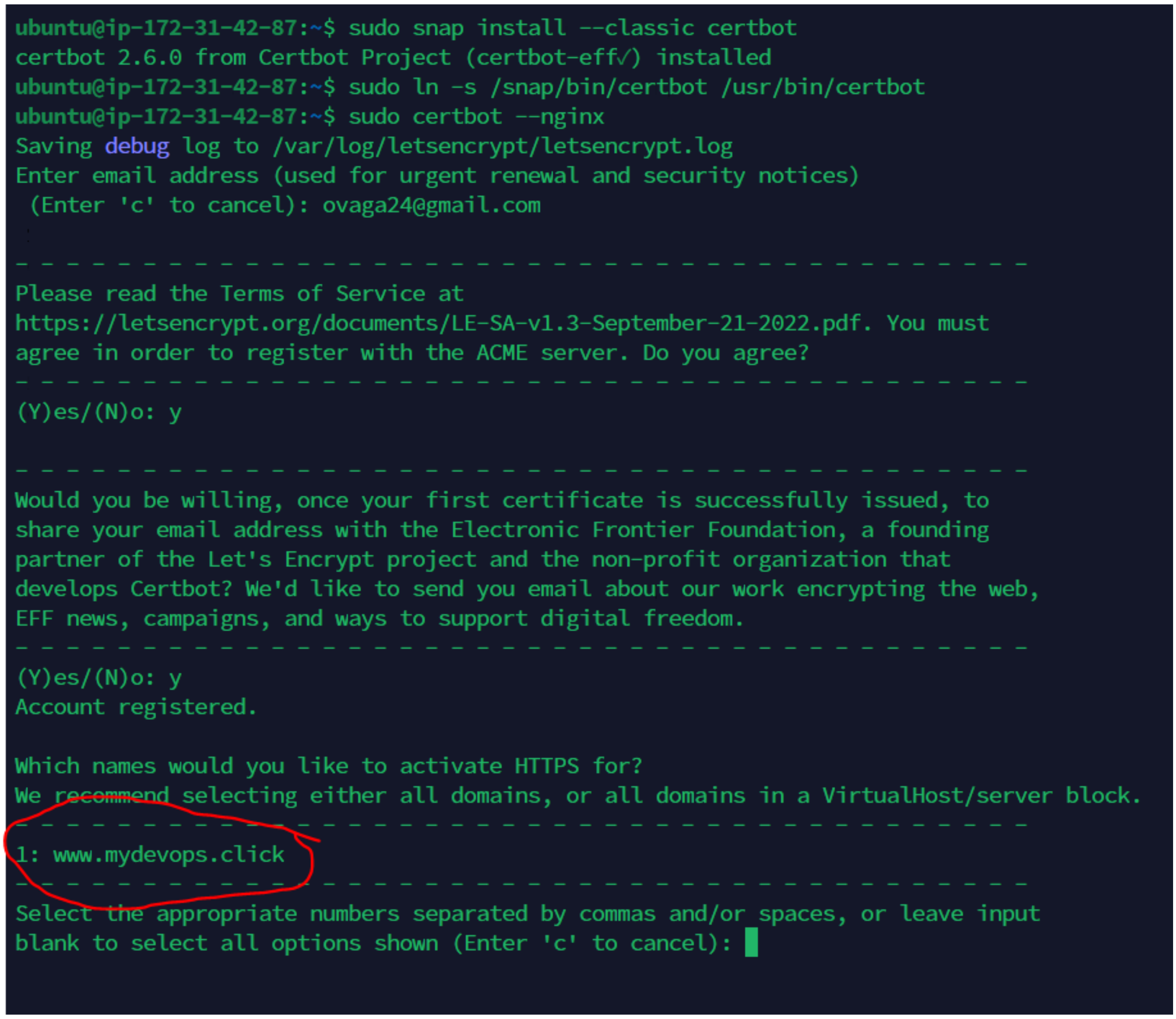
- Install certbot

`sudo snap install --classic certbot`

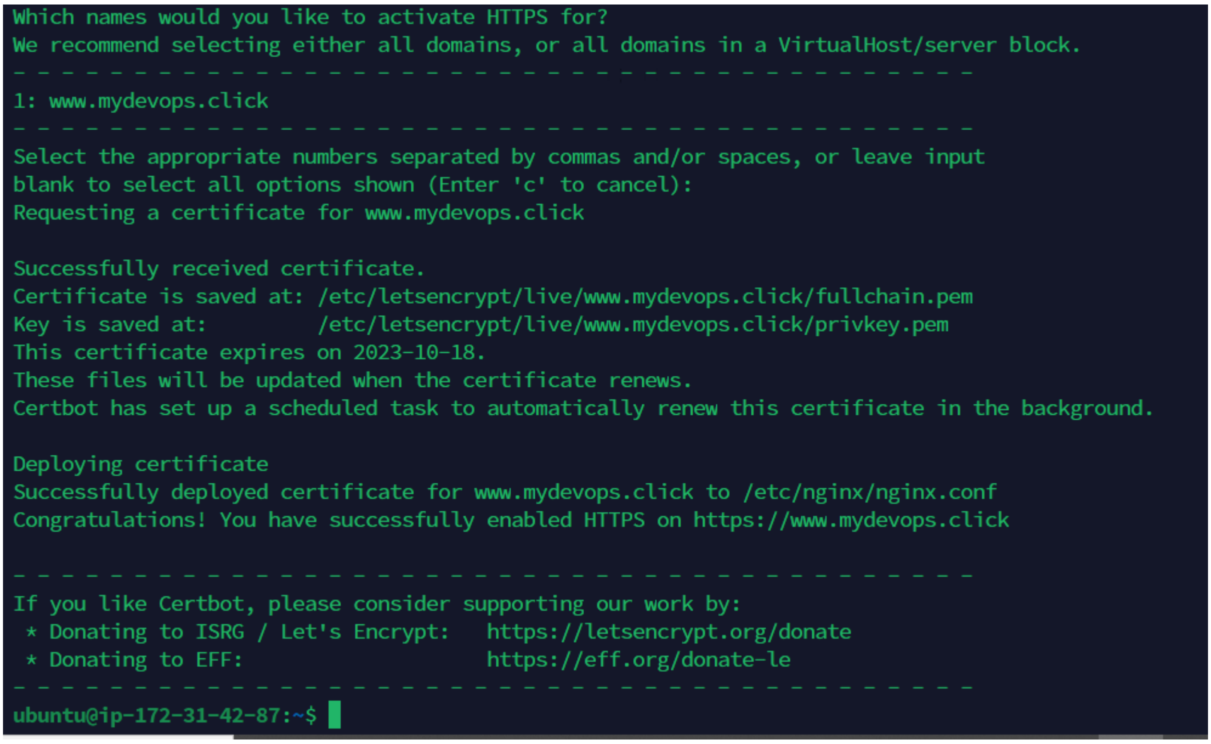


```
ubuntu@ip-172-31-42-87:~$ sudo snap install --classic certbot
certbot 2.6.0 from Certbot Project (certbot-eff✓) installed
ubuntu@ip-172-31-42-87:~$
```

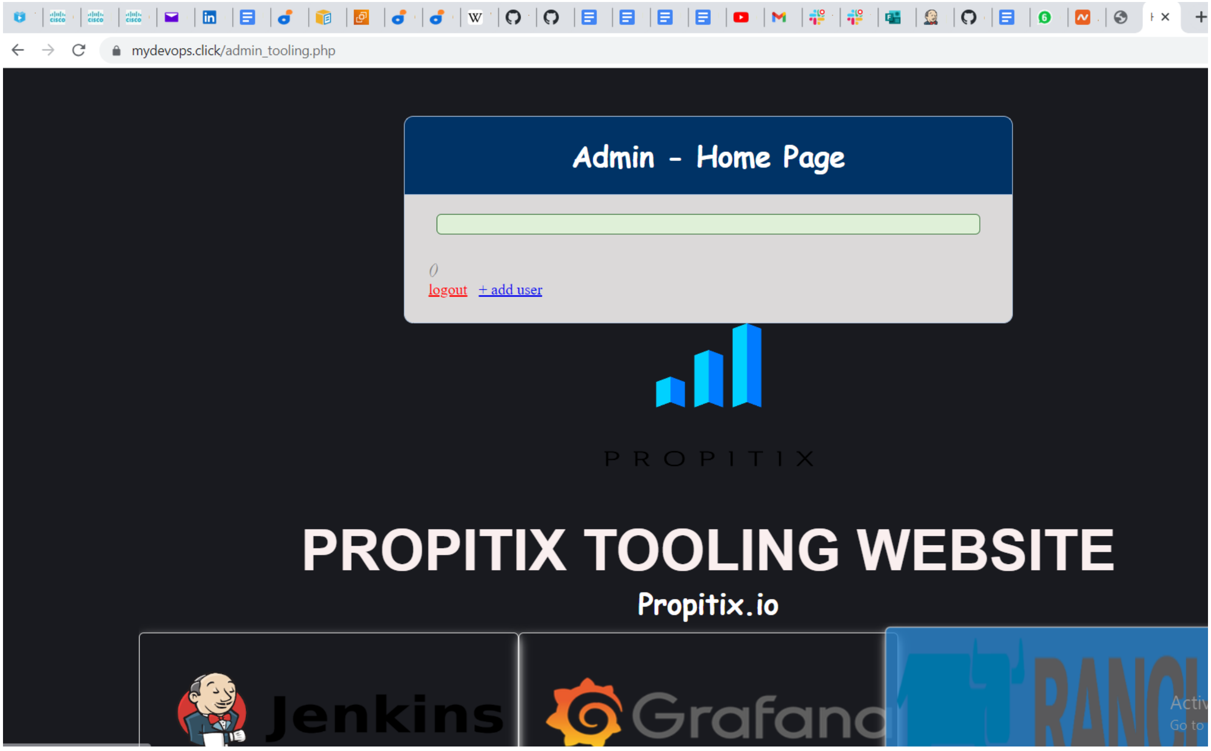
6. Make a Request your certificate for the domain name.

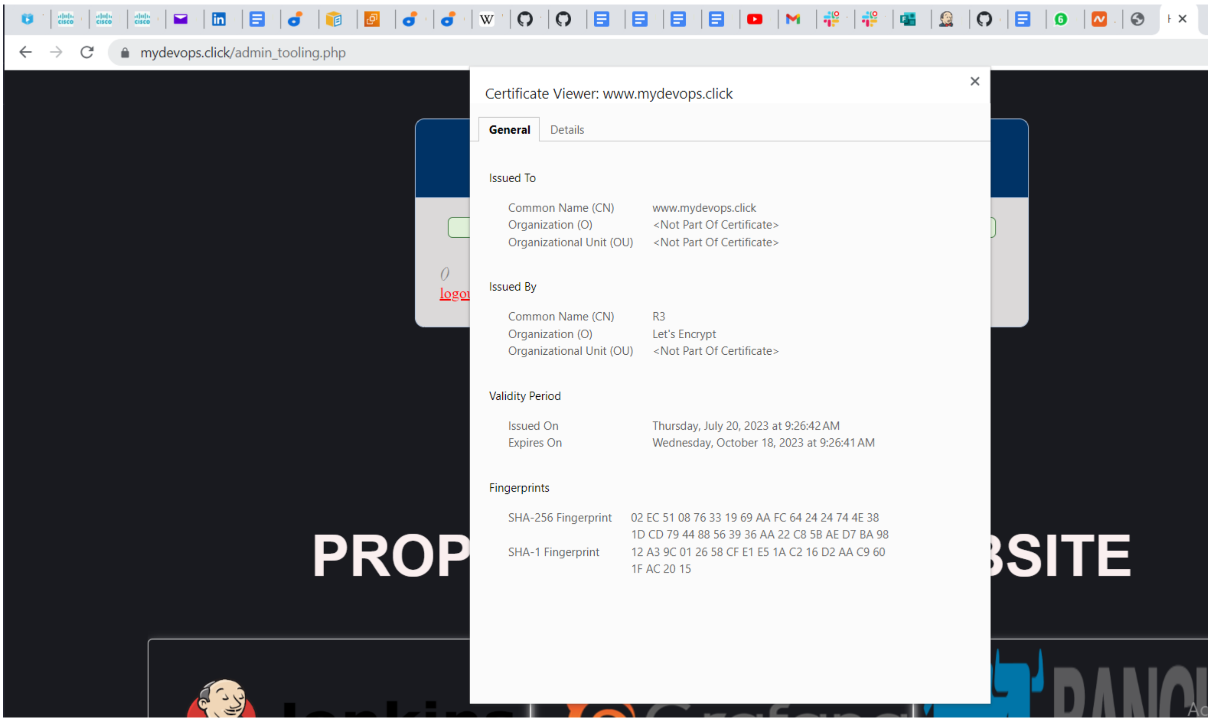


```
ubuntu@ip-172-31-42-87:~$ sudo snap install --classic certbot
certbot 2.6.0 from Certbot Project (certbot-eff✓) installed
ubuntu@ip-172-31-42-87:~$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
ubuntu@ip-172-31-42-87:~$ sudo certbot --nginx
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
 (Enter 'c' to cancel): ovaga24@gmail.com

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: y
Account registered.

Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: www.mydevops.click
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
```



```
Which names would you like to activate HTTPS for?
We recommend selecting either all domains, or all domains in a VirtualHost/server block.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: www.mydevops.click
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
Requesting a certificate for www.mydevops.click

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/www.mydevops.click/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/www.mydevops.click/privkey.pem
This certificate expires on 2023-10-18.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for www.mydevops.click to /etc/nginx/nginx.conf
Congratulations! You have successfully enabled HTTPS on https://www.mydevops.click

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ubuntu@ip-172-31-42-87:~$
```





Certificate Viewer: www.mydevops.click

Issued To

Common Name (CN) www.mydevops.click
Organization (O) <Not Part Of Certificate>
Organizational Unit (OU) <Not Part Of Certificate>

Issued By

Common Name (CN) R3
Organization (O) Let's Encrypt
Organizational Unit (OU) <Not Part Of Certificate>

Validity Period

Issued On Thursday, July 20, 2023 at 9:26:42 AM
Expires On Wednesday, October 18, 2023 at 9:26:41 AM

Fingerprints

SHA-256 Fingerprint 02 EC 51 08 76 33 19 69 AA FC 64 24 24 74 4E 38 1D CD 79 44 88 56 39 36 AA 22 C8 5B AE D7 BA 98
SHA-1 Fingerprint 12 A3 9C 01 26 58 CF E1 E5 1A C2 16 D2 AA C9 60 1F AC 20 15

6. Set up periodical renewal of your SSL/TLS certificate

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
sudo certbot --nginx
```

Follow the instruction displayed.

7. Lets Encrypt renews every 90 days and you can renew your certificate manually by running the following command
   sudo certbot renew --dry-run

We can also create a cron job to do this same thing at a stipulated time.

- Edit cron file

```
crontab -e
```

- Add the following line to the crontab file

```
5 */12 * */2 *   root /usr/bin/certbot renew > /dev/null 2>&1
```

- Save the crontab file