

## LOAD BALANCER SOLUTION WITH NGINX AND SSL/TLS

In this project we will register our website with [LetsEncrypt](#) Certificate Authority, to automate certificate issuance, we will use a shell client recommended by LetsEncrypt - [certbot](#).

### Task

This project consists of two parts:

Configure Nginx as a Load Balancer

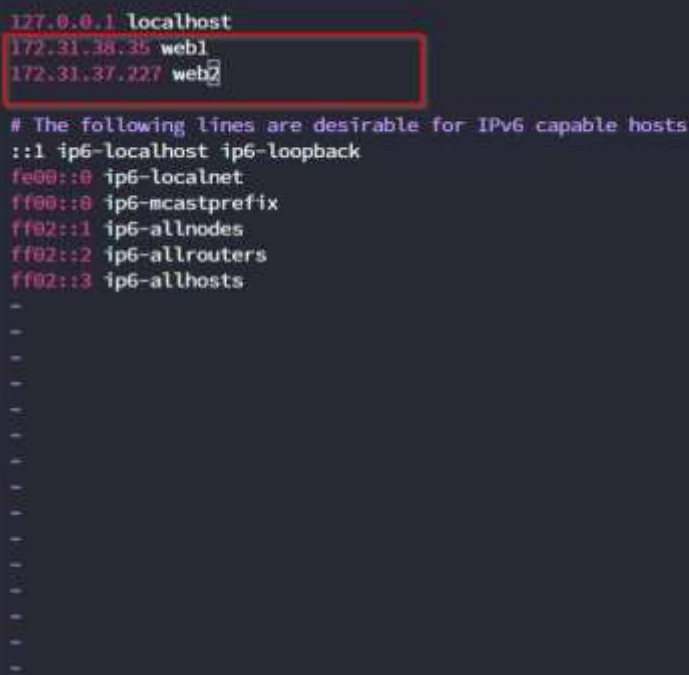
Register a new domain name and configure secured connection using SSL/TLS certificates.

### CONFIGURE NGINX AS A LOAD BALANCER

As a continuation from our previous project - DevOps tooling website Implementation - where we deployed the 3-tier architecture with a single database and an NFS server, we will create an EC2 instance based on Ubuntu Server 20.04 LTS and name it **Nginx-LB**.

- Update /etc/hosts file for local DNS with Web Servers' names (e.g., **Web1** and **Web2**) and their respective private IP addresses.

```
$ sudo vi /etc/hosts
```



```
127.0.0.1 localhost
172.31.38.35 web1
172.31.37.227 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
```

Since we are concerned about secured HTTPs and HTTP connections, will have to edit inbound rules to open ports 443 and 80 respectively from the instance security group.

- Update the instance and install nginx, then confirm status running.

```
$ sudo apt update
```

```
$ sudo apt install nginx -y
```

```
$ sudo systemctl status nginx
```

```
ubuntu@ip-172-31-33-240:~$ 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 Tue 2023-06-27 01:18:31 UTC; 1 day 17h ago
     Docs: man:nginx(8)
  Main PID: 28531 (nginx)
    Tasks: 2 (limit: 1141)
   Memory: 4.0M
   CGroup: /system.slice/nginx.service
           └─28531 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─28960 nginx: worker process

Jun 27 01:18:31 ip-172-31-33-240 systemd[1]: Starting A high performance web server and a reverse proxy server...
Jun 27 01:18:31 ip-172-31-33-240 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-33-240:~$
```

- Configure **Nginx-LB** using Web Servers' names defined in /etc/hosts. We will do this by opening the default nginx configuration file.

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

Paste in the configuration below into the 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/*;
```

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

server {
    server_name www.domain.com;
    location / {
        proxy_pass http://myproject;
    }
}
```

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

```
$ sudo systemctl restart nginx
```

```
$ sudo systemctl status nginx
```

Now we will move to the next part.

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

### For this part

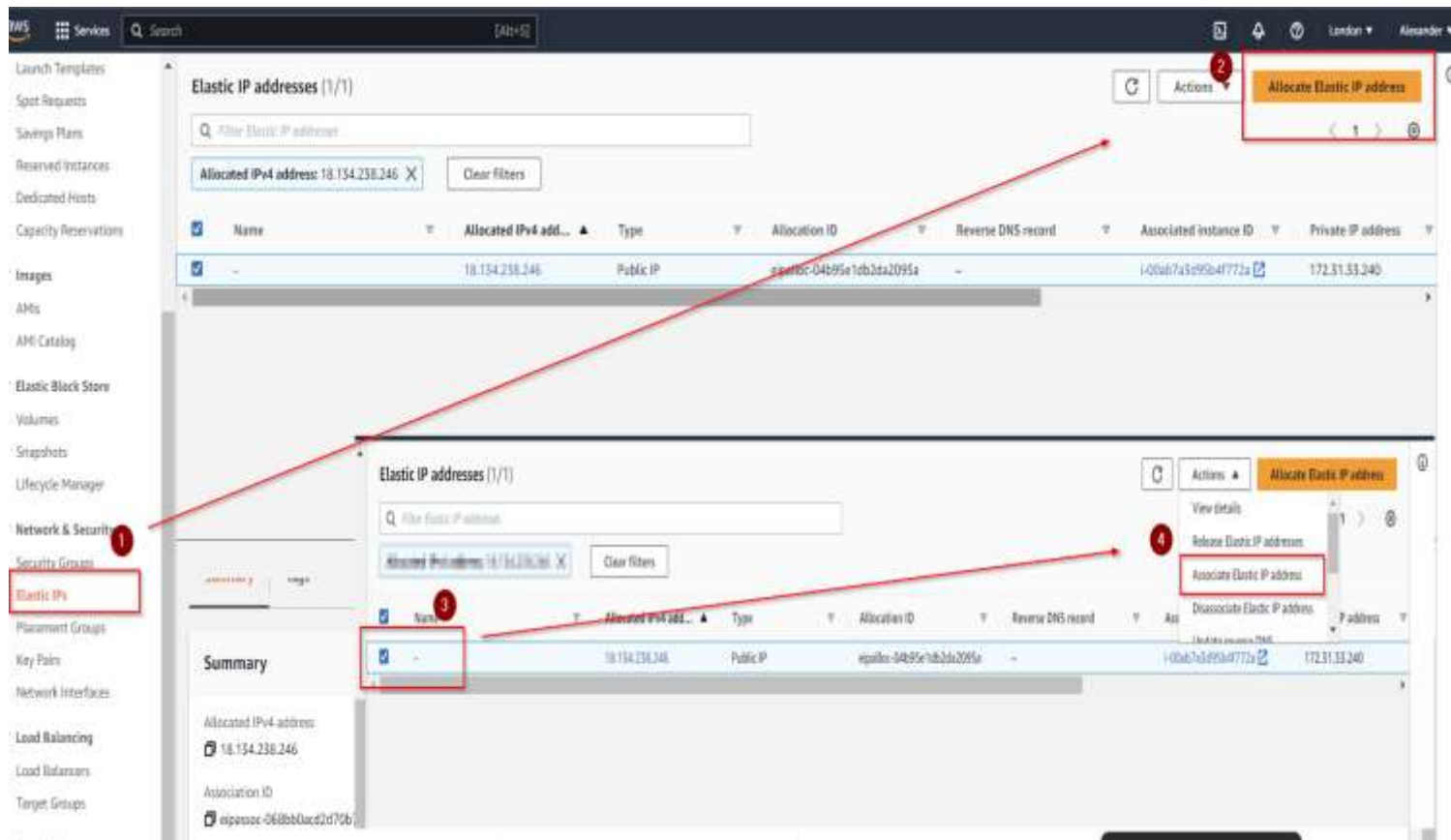
This section is where we will make necessary configurations to make connections to our Tooling Web Solution secured!

In order to get a valid SSL certificate – you need to register a new domain name; you can do it using any Domain name register – a company that manages reservation of domain names. The most popular ones are: [Godaddy.com](https://godaddy.com), [Domain.com](https://domain.com), [Bluehost.com](https://bluehost.com). There are so many others available when you search on google at a minimal fee. *In my case, I made use of [Fasthosts](https://fasthosts.com).*

1. You can go ahead and register a new domain name with any registrar of your choice in any domain zone.

2. We will assign an Elastic IP to our Nginx-LB server and associate our domain name with this Elastic IP.

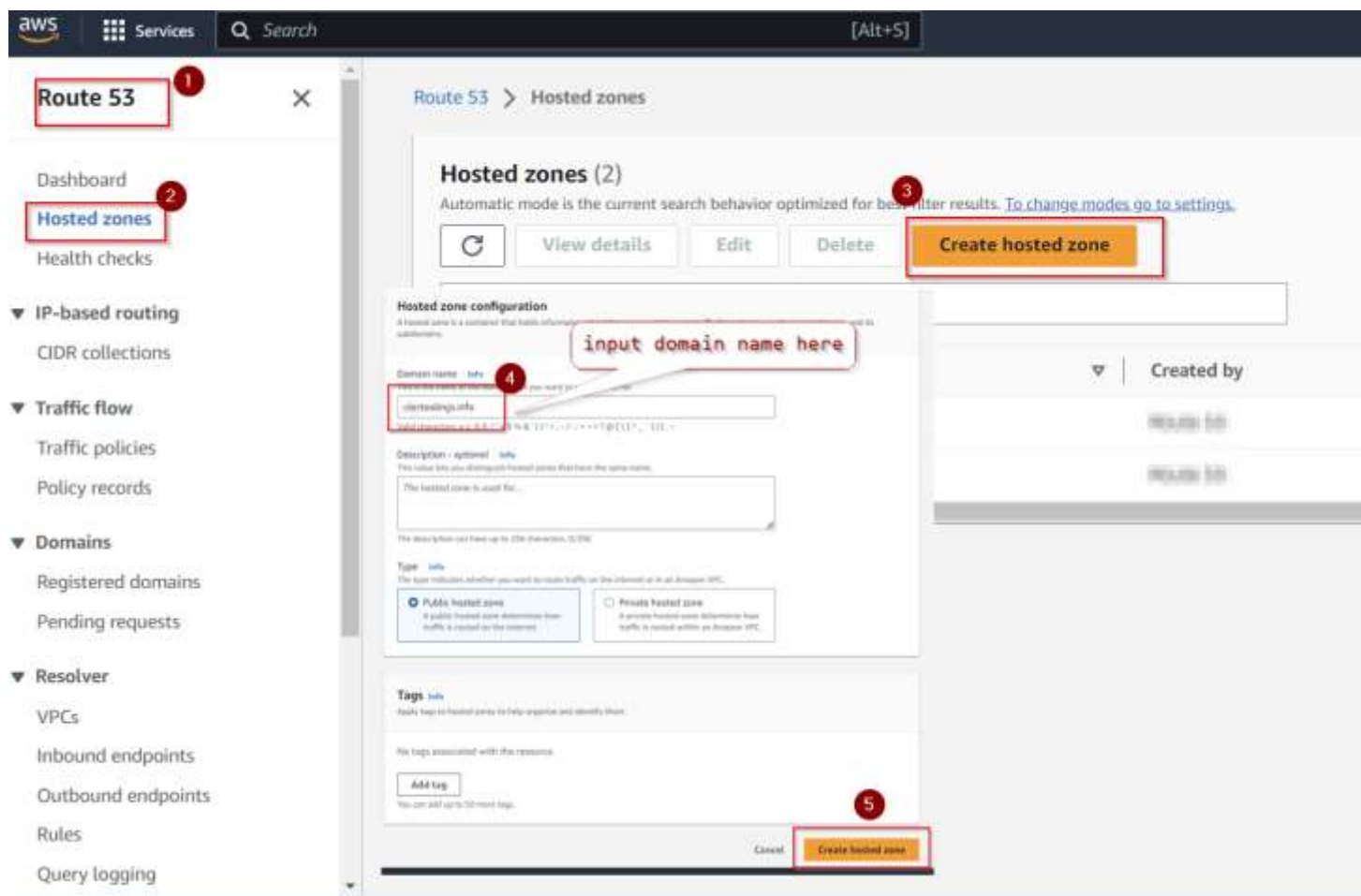
(Every time you restart or stop/start your EC2 instance you get a new public IP address. When you want to associate your domain name – it is better to have a static IP address that does not change after reboot. Elastic IP is the solution for this problem, see how to allocate an Elastic IP and associate it with an EC2 server instance below)



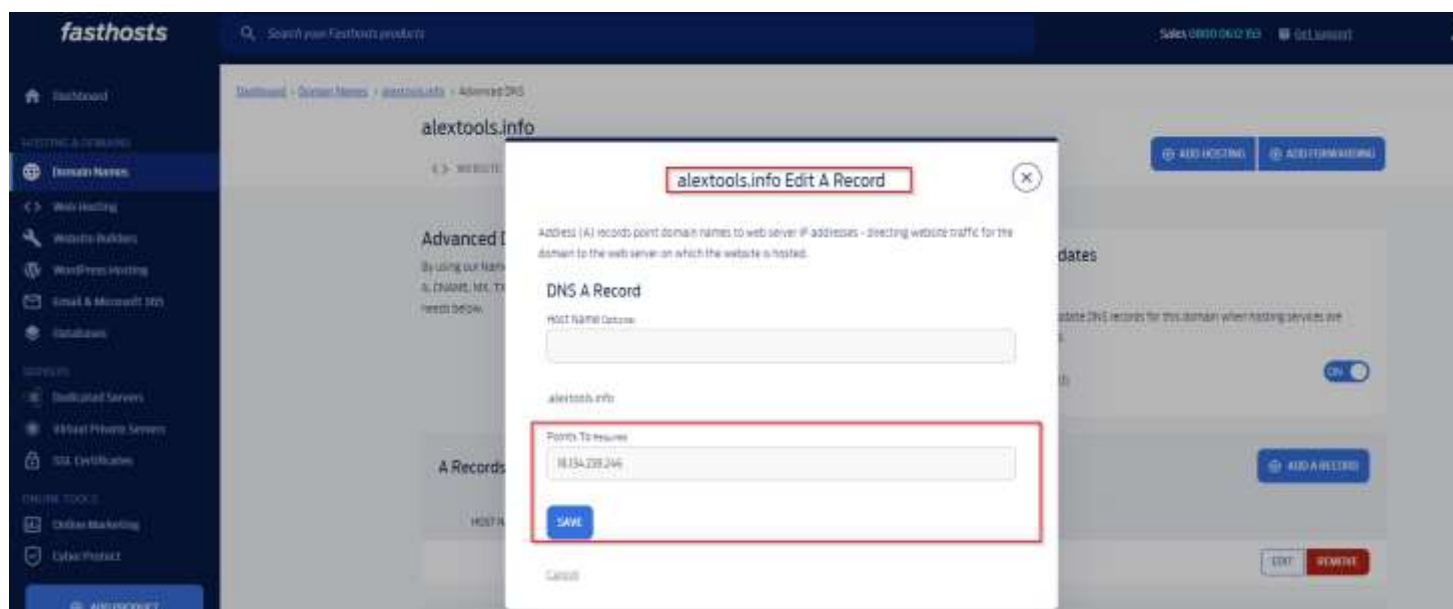
3. Update the **A record** in your registrar to point to **Nginx-LB** using your derived Elastic IP address. Also, you will need to update the nameserver (ns) record on the domain name settings and on AWS route 53.

See how to associate your new domain name to your Elastic IP below.

- Search for **route 53** on AWS and click.



#Create A record



aws Services Search [Alt+S]

Quick create record Switch to wizard

▼ Record 1 Delete

Record name Info  
 alextools.info  
Keep blank to create a record for the root domain.

Record type Info  
A – Routes traffic to an IPv4 address and some AWS resources

☐ Alias

Value Info  
  
Enter multiple values on separate lines.

TTL (seconds) Info  
 1m 1h 1d  
Recommended values: 60 to 172800 (two days)

Routing policy Info  
Simple routing

Add another record

Cancel Create records

aws Services Search [Alt+S]

Route 53 ×

Dashboard  
Hosted zones  
Health checks

▼ IP-based routing  
ODR collection

▼ Traffic flow  
Traffic policies  
Policy records

▼ Domains  
Registered domains  
Pending requests

▼ Resolver  
VPCs  
Inbound endpoints  
Outbound endpoints  
Rules

Created A record can be seen here

Hosted zone details Edit hosted zone

Records (4) DNSSEC signing Hosted zone tags (0)

Records (4) Info  
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

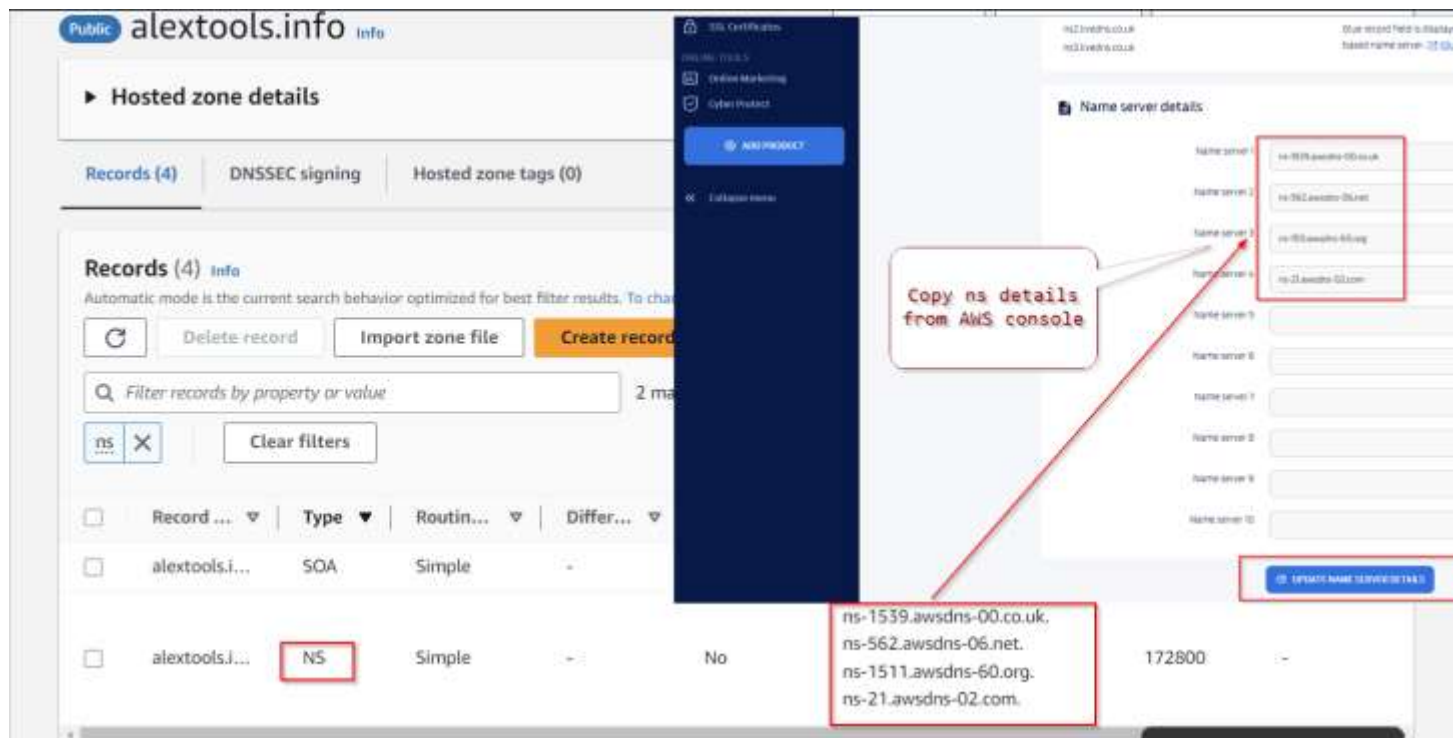
Refresh Delete record Import zone file Create record

Type ▼ Routing policy ▼ Alias ▼ < 1 > ⓘ

<input type="checkbox"/>	Record ... ▼	Type ▼	Routin... ▼	Differ... ▼	Alias ▼	Value/Routes traffic to ▼	TTL (s... ▼	Health ... ▼
<input type="checkbox"/>	alextools.i...	A	Simple	-	No	18.134.238.246	300	-
<input type="checkbox"/>	alextools.i...	NS	Simple	-	No	ns-1539.awsdns-00.co.uk, ns-562.awsdns-06.net, ns-1511.awsdns-60.org, ns-21.awsdns-02.com	172800	-
<input type="checkbox"/>	alextools.i...	SOA	Simple	-	No	ns-1539.awsdns-00.co.uk; a...	900	-
<input type="checkbox"/>	www.alex...	A	Simple	-	No	18.134.238.246	300	-

#Update nameserver (ns) record on domain website.





*Note: Depending on the domain name registrar you adopt to host your website, it can take from a few minutes to over 24hours for your account to get activated*

Update your nginx.conf with server\_name [www.<your-domain-name.com>](http://www.<your-domain-name.com>) instead of server\_name [www.domain.com](http://www.domain.com)

**\$ sudo vi /etc/nginx/nginx.conf**

```
##
# Virtual Host Configs
##

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

server {
    server_name alextools.info www.alextools.info;
    location / {
        proxy_pass http://myproject;
    }
}
```

Check that your Web Servers can be reached from your browser using new domain name using HTTP protocol - <http://<your-domain-name.com>>

In my case, it worked! With registered domain <http://alextools.info>.

Install certbot and request for an SSL/TLS certificate.

Make sure snapd service is active and running.

```
$ sudo systemctl status snapd
```

```
ubuntu@ip-172-31-33-240:~$ sudo systemctl status snapd
● snapd.service - Snap Daemon
   Loaded: loaded (/lib/systemd/system/snapd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-06-25 06:08:08 UTC; 3 days ago
     TriggeredBy: ● snapd.socket
   Main PID: 12609 (snapd)
      Tasks: 11 (limit: 1141)
     Memory: 88.4M
    CGroup: /system.slice/snapd.service
            └─12609 /usr/lib/snapd/snapd

Jun 26 19:48:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "amazon-ssm-agent", "core18", "core20", "lx
Jun 27 02:03:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "amazon-ssm-agent", "core18", "core20", "lx
Jun 27 02:03:17 ip-172-31-33-240 snapd[12609]: api_snaps.go:366: Installing snap "certbot" revision unset
Jun 27 07:18:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "amazon-ssm-agent", "certbot", "core18", "c
Jun 27 16:53:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "amazon-ssm-agent", "certbot", "core18", "c
Jun 27 21:43:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "amazon-ssm-agent", "certbot", "core18", "c
Jun 28 05:43:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "t", "core18", "c
Jun 28 10:43:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "t", "core18", "c
Jun 28 13:48:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "t", "core18", "c
Jun 28 19:53:08 ip-172-31-33-240 snapd[12609]: storehelpers.go:769: cannot refresh: snap has no updates available: "t", "core18", "c
[lines 1-20/20 (END)]
```

Install certbot

```
$ sudo snap install --classic certbot
```

Request your certificate (just follow the certbot instructions - you will need to choose which domain you want your certificate to be issued for, domain name will be looked up from **nginx.conf** file so make sure you have updated it as per previous step.

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

```
$ sudo certbot --nginx
```



```

ubuntu@ip-172-31-33-240:~$ 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):
Invalid email address: .

If you really want to skip this, you can run the client with
--register-unsafely-without-email but you will then be unable to receive notice
about impending expiration or revocation of your certificates or problems with
your Certbot installation that will lead to failure to renew.

Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): x_obail000@yahoo.co.uk

-----
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: alextools.info
2: www.alextools.info
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 1,2

```

Set up periodical renewal of your SSL/TLS certificate.

By default, LetsEncrypt certificate is valid for 90 days, so it is recommended to renew it at least every 60 days or more frequently.

You can test renewal command in dry-run mode

**\$ sudo certbot renew --dry-run**

```

ubuntu@ip-172-31-33-240:~$ sudo certbot renew --dry-run
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
Processing /etc/letsencrypt/renewal/alextools.info.conf
-----
Simulating renewal of an existing certificate for alextools.info and www.alextools.info

-----
Congratulations, all simulated renewals succeeded:
  /etc/letsencrypt/live/alextools.info/fullchain.pem (success)
-----

```

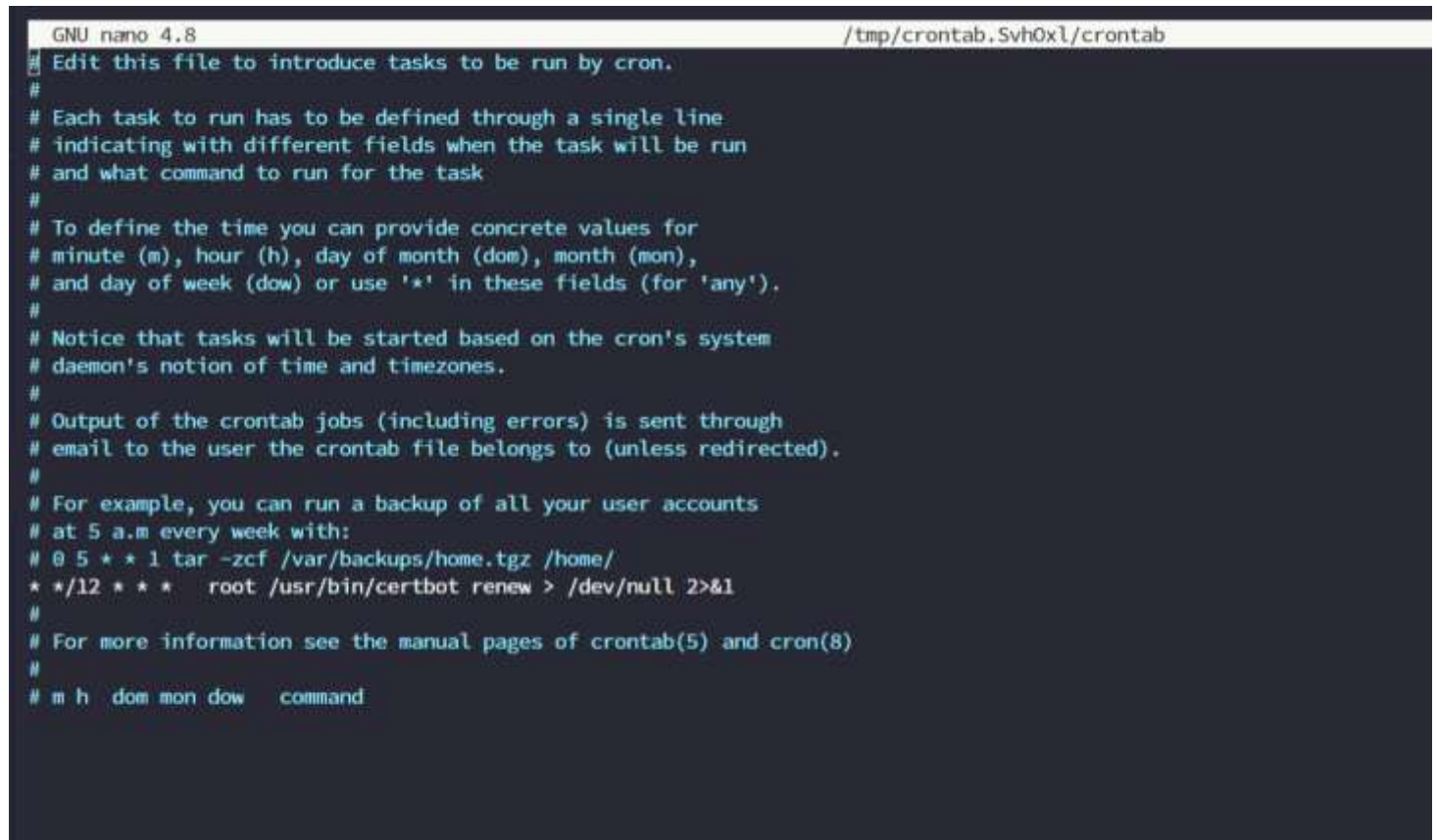
Best practice is to have a scheduled job that can run renew command periodically. Let us configure a cronjob to run the command twice a day.

To do so, lets edit the [crontab file](#) with the following command:

```
$ crontab -e
```

Add following line:

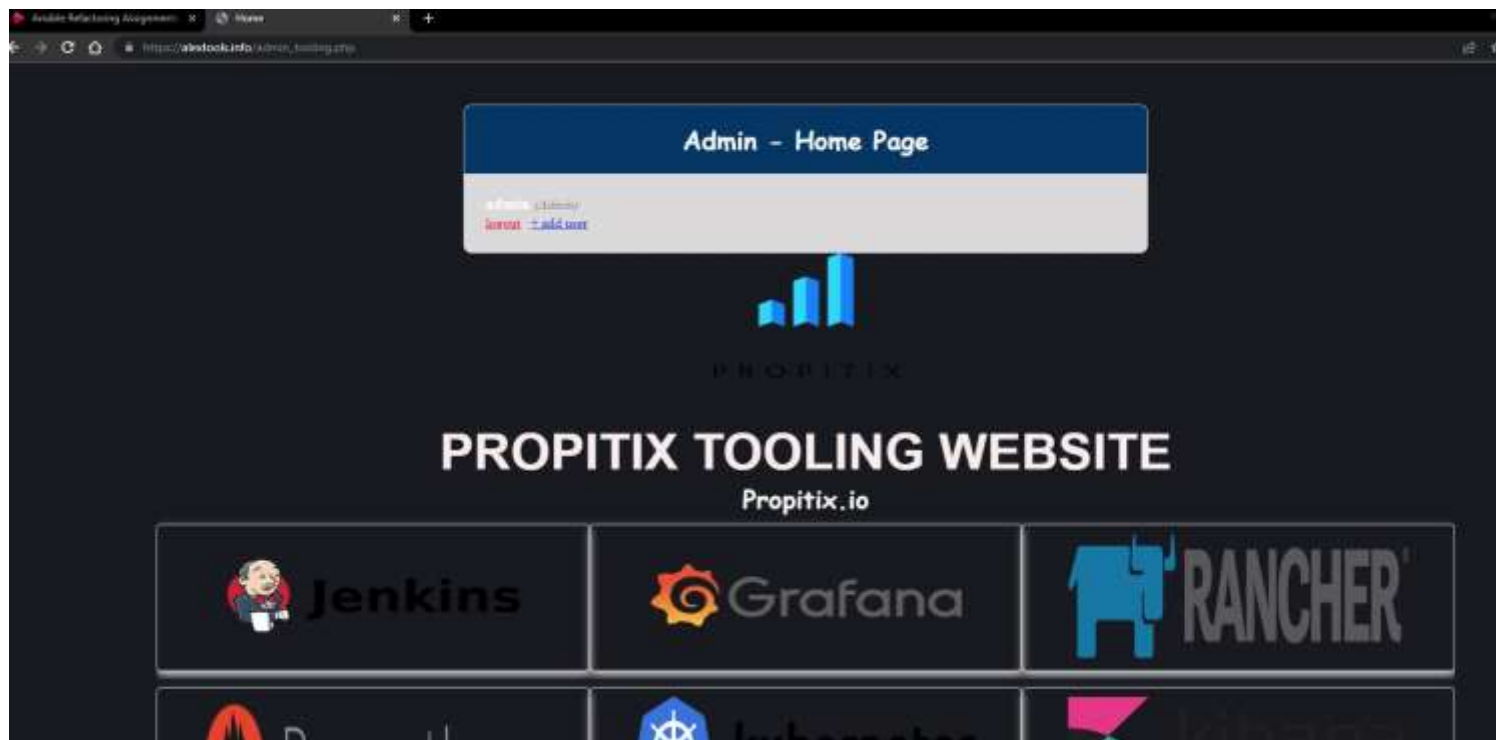
```
* */12 * * * root /usr/bin/certbot renew > /dev/null 2>&1
```



```
GNU nano 4.8 /tmp/crontab.Svh0x1/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
* */12 * * * root /usr/bin/certbot renew > /dev/null 2>&1
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
```

You can always change the interval of this cronjob if twice a day is too often by adjusting schedule expression.

Now check that your Web Servers can be reached from your browser using new domain name using HTTPs protocol - <https://<your-domain-name.com>> and that the padlock pictorial is visible with the domain url.



## Congratulations!

You have just implemented an Nginx Load Balancing Web Solution with secured HTTPS connection with periodically updated SSL/TLS certificates.