

# Certbot & NGINX on AWS

Did you know you can use CertBot and NGINX to have a wildcard certificate? Here's how to do it with an AWS Ubuntu sever.

---

## Prerequisites:

- AWS Route 53 DNS hosted zone
  - Web server using NGINX
  - Website already configured using SSL
  - SSH access with sudo (root) privileges
  - Knowledge and comfort navigating linux using the bash shell
  - Knowledge and comfort on how to view and edit files in linux (ie. vi, vim, nano...)
- 

## Overview:

The high level process to achieve our objective is as follows:

- Installing CertBot
- Installing DNS Plugin
- Create IAM Policy
- Create IAM Role
- Associate IAM Role with EC2 Instance
- Run CertBot and get new Certs
- Update NGINX to use new SSL Certs
- Test and restart NGINX
- Validate SSL Cert
- Test and review CertBot auto renewal

**Disclaimer:** *As with any change, please make sure that you have created a Jira ticket, received proper approval, notified business partners, scheduled the action and taken the necessary actions to backup and recover should anything go wrong.*

---

## Installing CertBot:

SSH to the web server and run the following commands:

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt-repository universe
```

```
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install certbot python-certbot-nginx
```

---

## Install DNS Plugin:

SSH to the web server and run the following command:

```
sudo apt-get install python3-certbot-dns-route53
```

---

## Create IAM Policy:

See also: <https://certbot-dns-route53.readthedocs.io/en/stable/>

Create new IAM policy using the AWS Route53 ZoneID of the hosted zone that you want to get an SSL Cert for.

```
{
  "Version": "2012-10-17",
  "Id": "certbot-dns-route53 sample policy",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53:ListHostedZones",
        "route53:GetChange"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect" : "Allow",
      "Action" : [
        "route53:ChangeResourceRecordSets"
      ],
      "Resource" : [
        "arn:aws:route53:::hostedzone/YOURHOSTEDZONEID"
      ]
    }
  ]
}
```

```
]
}
]
}
```

---

## Create a new IAM Role:

- Click [Create Role] > [AWS Service] > [EC2] > [Next: Permissions]
- Search for and select your newly created Policy (one created from above)
- Click [Next: Tags] > (Enter a TAG if you wish) > [Next: Review]
- Give your new role a meaningful name and description
- Click [Create Role]

---

## Associate Role with EC2 Instance:

- Click to select your EC2 Instance
- Click [Actions] > Instance settings > [Attach / Replace IAM Role]
- In the “IAM Role” dropdown list, click and select the IAM Role that you created (from above)
- Click [Apply] > [Close]

---

## Run CertBot and get new Certs:

It's important to get both the example.com and \*.example.com as WILDCARD certs need to include the naked domain as well as any sub domains.

**Note:** *Be sure to review/update example.com, \*.example.com before running the below command.*

```
sudo certbot certonly --dns-route53 -d example.com -d *.example.com --dns-route53-propagation-seconds 30 -m domains@mysite.com --agree-tos
```

---

If the above command runs successfully, it will populate the necessary certificate key files into the /etc/letsencrypt/live/example.com/ directory.

---

## Update NGINX to use new SSL Certs:

The next step requires that you update the existing SSL configuration of the NGINX server to use the new LetsEncrypt certs. There are a few common locations to check:

- /etc/nginx/nginx.conf
- /etc/nginx/sites-available/<site name>
- /etc/nginx/snippets/
- Update the following folders with new “fullchain.pem and privkey.pem”
- beta\_ssl.conf , fastcgi-php.conf , rc\_ssl.conf , snakeoil.conf

Between these locations, you should be able to locate the SSL configuration/settings. What you're looking for are the following two keys:

- ssl\_certificate
- ssl\_certificate\_key

Below is a description of the newly downloaded LetsEncrypt keys

- `privkey.pem` : the private key for your certificate.
- `fullchain.pem` : the certificate file used in most server software.
- `chain.pem` : used for OCSP stapling in Nginx  $\geq 1.3.7$ .
- `cert.pem` : will break many server configurations, and should not be used without reading further documentation

You need to update the following SSL entries to point to the new LetsEncrypt keys

- ssl\_certificate /etc/letsencrypt/live/`example.com`/fullchain.pem;
- ssl\_certificate\_key /etc/letsencrypt/live/`example.com`/privkey.pem;

---

## Test and restart NGINX:

Test that there are no errors in any of your NGINX files by running the following command

```
sudo nginx -t
```

If all of the tests come back as successful, you can go ahead and restart the nginx service

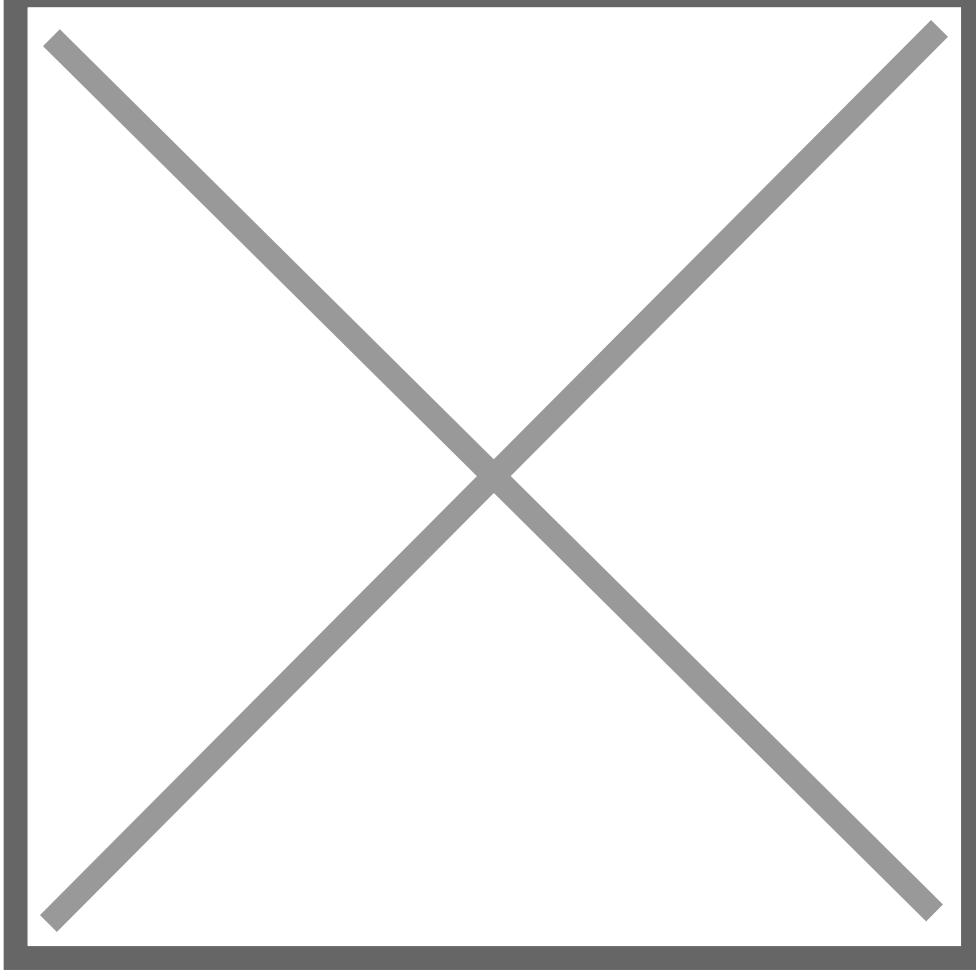
```
sudo service nginx restart
```

---

## Validate SSL Cert:

Once restarted, open a browser window and visit your site. You want to validate that the website is using the new LetsEncrypt SSL cert and that the expiration is set 90 days out. Individual browser instructions can be found in the link provided below, however what you're looking for is something like this:

Image not found or type unknown



SSL Cert Info or type unknown

Instructions on how to view SSL certificate details in each browser can be found at <https://www.globalsign.com/en/blog/how-to-view-ssl-certificate-details/>

---

## Test and review CertBot auto renewal:

The last thing to do before finishing up is making sure that both the automatic renewal process will work and that it's scheduled.

To test the auto renewal process run the following on the web server:

```
sudo certbot renew --dry-run
```

If successful you can check to see if a scheduled task is set to automatically run the renew process. By default, Certbot tries to renew the cert once every 12 hours. The command to renew certbot will be installed in one of the following locations:

- /etc/crontab/

- /etc/cron.\*/\* – (ie. /etc/cron.d/certbot)
- systemctl list-timers

To check the status of the certbot including the auto renew cron job run the following command:

```
sudo tail -50 /var/log/letsencrypt/letsencrypt.log
```

---

## More information:

- <https://certbot.eff.org/lets-encrypt/ubuntuxenial-nginx>
- <https://certbot-dns-route53.readthedocs.io/en/stable/>

---

Revision #1

Created 14 June 2024 00:57:36 by peterd

Updated 14 June 2024 00:58:13 by peterd