Reference:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SSL-on-amazon-linux-2.html#ssl_enable>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/set-hostname.html>

<https://aws.amazon.com/blogs/compute/extending-amazon-linux-2-with-epel-and-lets-encrypt/>

Tutorial: Configure SSL/TLS on Amazon Linux 2

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS on an EC2 instance with Amazon Linux 2 and Apache web server.

## **Prerequisites**

Before you begin this tutorial, complete the following steps:

* Launch an EBS-backed Amazon Linux 2 instance.
* Configure your security groups to allow your instance to accept connections on the following TCP ports:
  + SSH (port 22)
  + HTTP (port 80)
  + HTTPS (port 443)
* Install the Apache web server.
* To identify and authenticate websites, the TLS public key infrastructure (PKI) relies on the Domain Name System (DNS). To use your EC2 instance to host a public website, you need to register a domain name for your web server or transfer an existing domain name to your Amazon EC2 host.

## **Step 1: Enable TLS on the server**

This procedure takes you through the process of setting up TLS on Amazon Linux 2 with a self-signed digital certificate.

**To enable TLS on a server**

1. [Connect to your instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-connect-to-instance-linux) and confirm that Apache is running.

[ec2-user ~]$ **sudo systemctl is-enabled httpd**

If the returned value is not "enabled," start Apache and set it to start each time the system boots.

[ec2-user ~]$ **sudo systemctl start httpd && sudo systemctl enable httpd**

[ec2-user ~]$ **sudo yum update -y**

1. Now that your instance is current, add TLS support by installing the Apache module mod\_ssl.

[ec2-user ~]$ **sudo yum install -y mod\_ssl**

1. Your instance now has the following files that you use to configure your secure server and create a certificate for testing:
   * /etc/httpd/conf.d/ssl.conf

The configuration file for mod\_ssl. It contains *directives* telling Apache where to find encryption keys and certificates, the TLS protocol versions to allow, and the encryption ciphers to accept.

* + /etc/pki/tls/certs/make-dummy-cert

A script to generate a self-signed X.509 certificate and private key for your server host. This certificate is useful for testing that Apache is properly set up to use TLS. Because it offers no proof of identity, it should not be used in production. If used in production, it triggers warnings in Web browsers.

1. Run the script to generate a self-signed dummy certificate and key for testing

.

[ec2-user ~]$ **cd /etc/pki/tls/certs**

**sudo ./make-dummy-cert localhost.crt**

1. Open the /etc/httpd/conf.d/ssl.conf file using your favorite text editor (such as **vim** or **nano**) and comment out the following line, because the self-signed dummy certificate also contains the key. If you do not comment out this line before you complete the next step, the Apache service fails to start.

SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

1. Restart Apache.

[ec2-user ~]$ **sudo systemctl restart httpd**

1. Your Apache web server should now support HTTPS (secure HTTP) over port 443. Test it by entering the IP address or fully qualified domain name of your EC2 instance into a browser URL bar with the prefix **https://**.

Because you are connecting to a site with a self-signed, untrusted host certificate, your browser may display a series of security warnings. Override the warnings and proceed to the site.

If the default Apache test page opens, it means that you have successfully configured TLS on your server. All data passing between the browser and server is now encrypted.

**To change the system hostname to a public DNS name**

1. For Amazon Linux 2: Use the **hostnamectl** command to set your hostname to reflect the fully qualified domain name (such as **webserver.mydomain.com**).

[ec2-user ~]$ **sudo hostnamectl set-hostname** *webserver.mydomain.com*

* + For Amazon Linux AMI: On your instance, open the /etc/sysconfig/network configuration file in your favorite text editor and change the HOSTNAME entry to reflect the fully qualified domain name (such as **webserver.mydomain.com**).

HOSTNAME=*webserver.mydomain.com*

# Extending Amazon Linux 2 with EPEL and Let’s Encrypt

### Step 1: Install and enable EPEL

1. [Connect](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-connect-to-instance-linux) to your Amazon Linux 2 instance at the Elastic IP address that you just created.
2. Download and install the EPEL repository using the following commands:
3. cd /tmp
4. wget -O epel.rpm –nv \

<https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

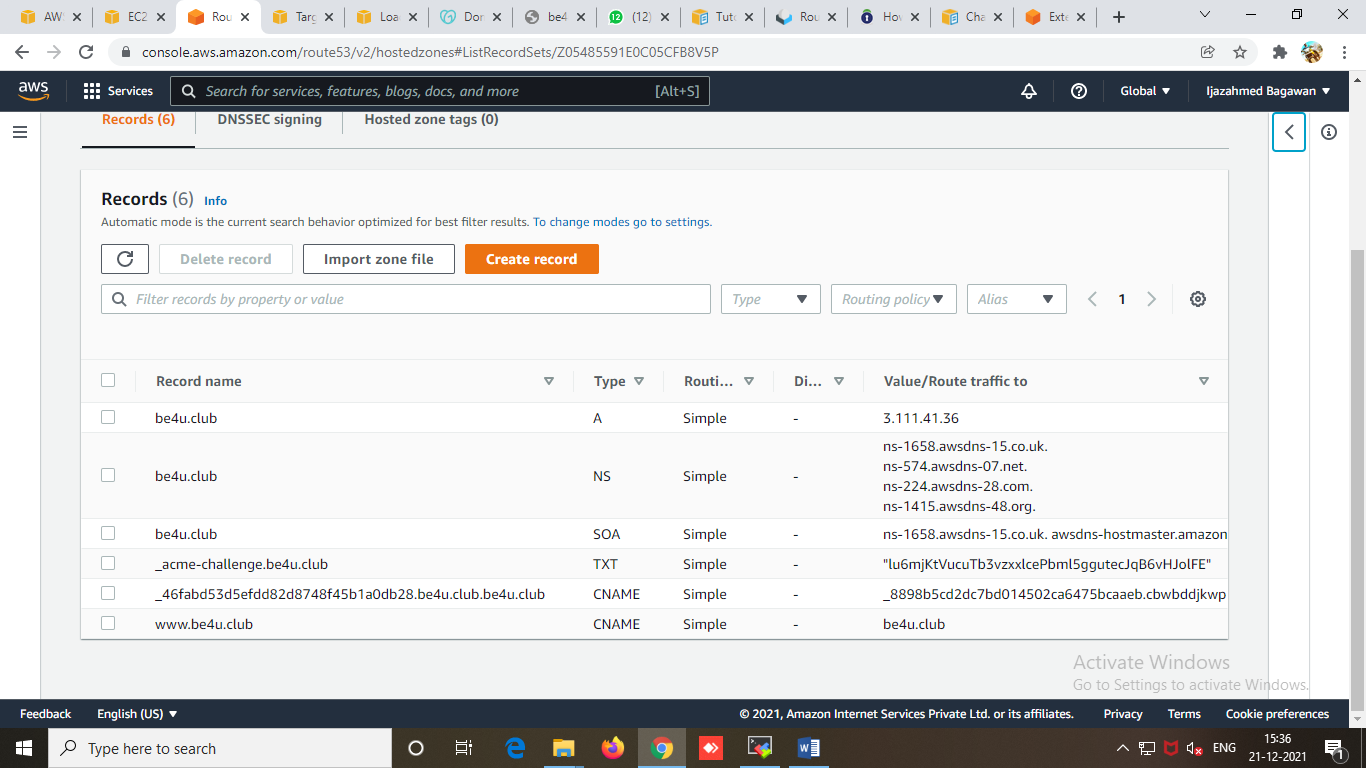
1. sudo yum install -y ./epel.rpm

### Step 3: Install and configure Let’s Encrypt

1. If you are no longer connected to the Amazon Linux 2 instance, [connect](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-connect-to-instance-linux) to it at the Elastic IP address that you just created.
2. Install certbot, the Let’s Encrypt client to be used to obtain an SSL/TLS certificate and install it into Apache.

sudo yum install python2-certbot-apache.noarch

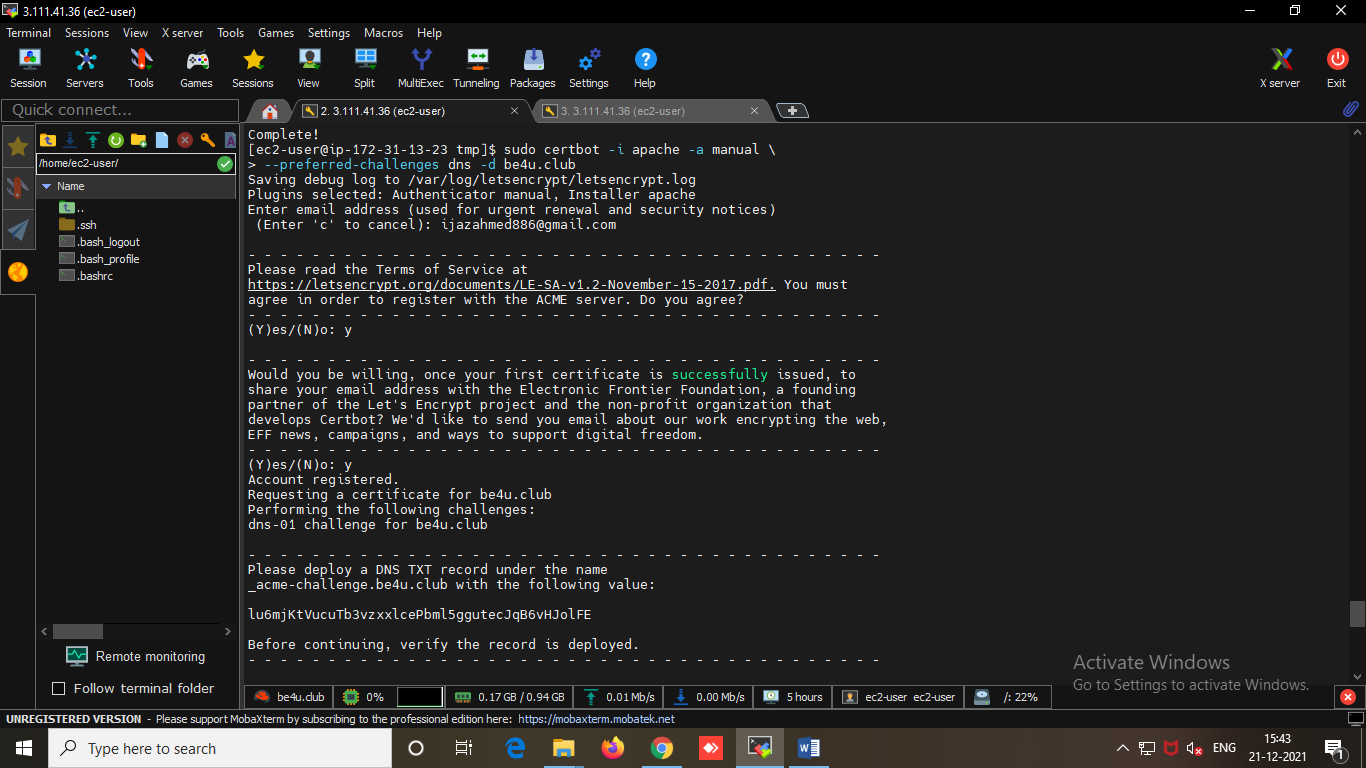
1. Create a DNS “A record” that maps a host name to the Elastic IP address. For this post, assume that the name of the host is lamp.example.com. If you are hosting your DNS in Amazon Route 53, do this by [creating the appropriate record set](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-creating.html).



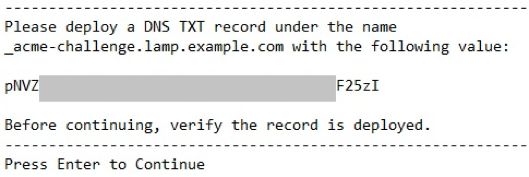
1. After the “A record” has propagated, browse to *lamp.example.com*. The Apache test page should appear. If the page does not appear, use a tool such as *nslookup* on your workstation to confirm that the DNS record has been properly configured.
2. Use the following command to install certbot:

sudo certbot -i apache -a manual \

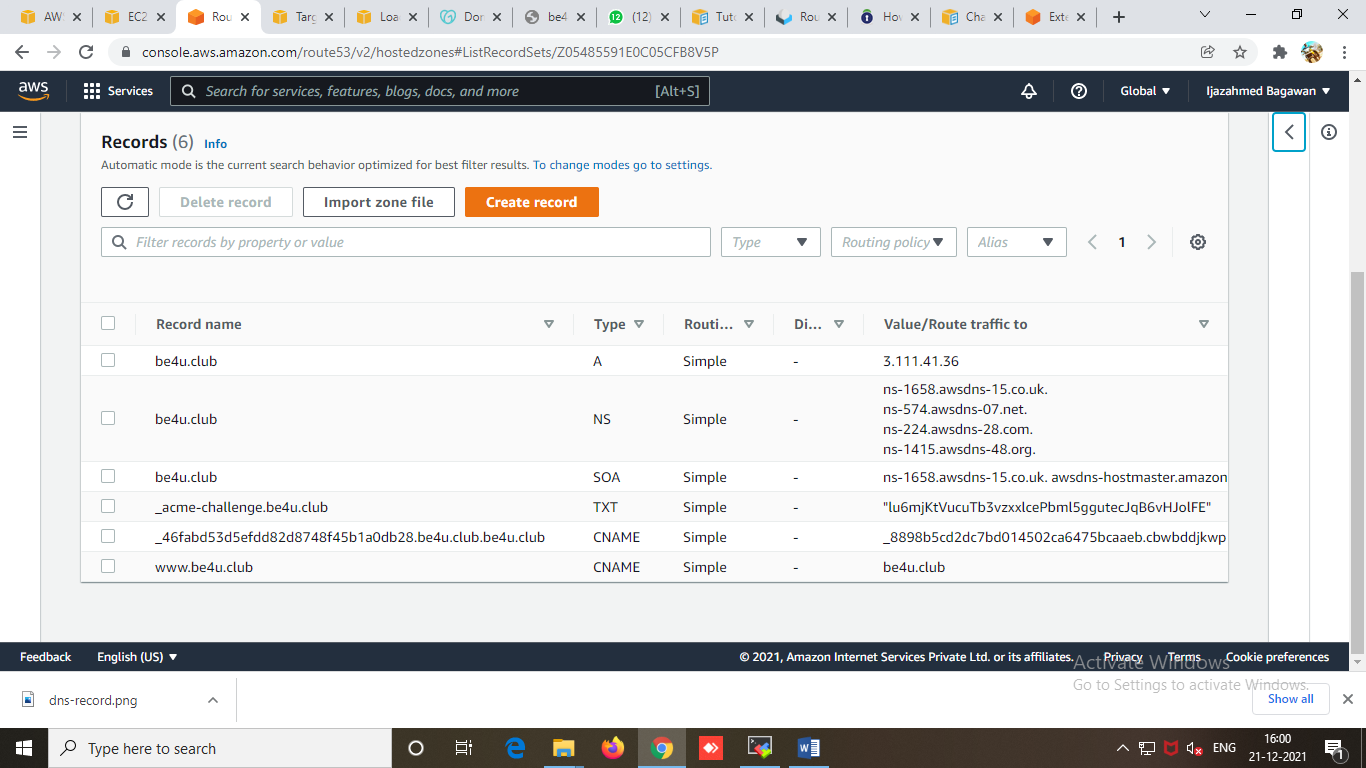
--preferred-challenges dns -d be4u.club



You are prompted to deploy a DNS TXT record with the name “\_acme-challenge.be4u.club” with the supplied value, as shown below.



**Enter the DNS TXT RECORD generated by EC2 instance in Route53**



**Open a SSH duplicate Window of same EC2 machine :**

After you enter the record, wait until the TXT record propagates. To look up the TXT record to confirm the deployment, use the nslookup command in a separate command window, as shown below. Remember to use the set ty=txt command before entering the

TXT record.[](https://d2908q01vomqb2.cloudfront.net/1b6453892473a467d07372d45eb05abc2031647a/2018/05/15/nslookup.png)