# **CONFIGURING JENKINS WITH SSL USING NGNIX AS REVERSE PROXY BY LETSENCRYPT**

# **Description:-**

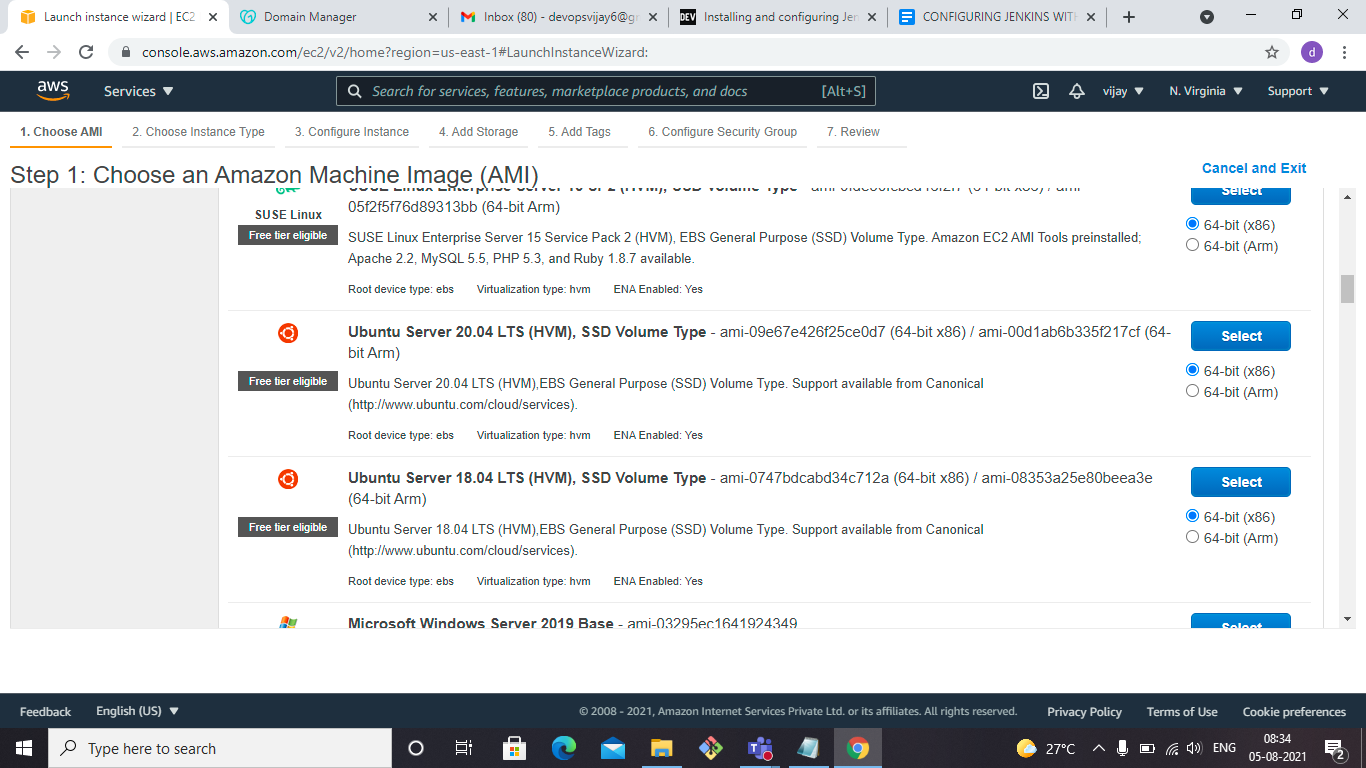
Iam are going to install Jenkins in ubuntu 20.04 AMI

# **Installation:-**

## **#1 First Step - Launching Instance:-**

----------------------------------------------------------------------------

### **Selecting the AMI**



### **Type of Instance**

In the next step you need to select an Instance Type, you can select t2.micro for the eligible free tier or any Instance according to your needs.

### 

### **Configure Instance**

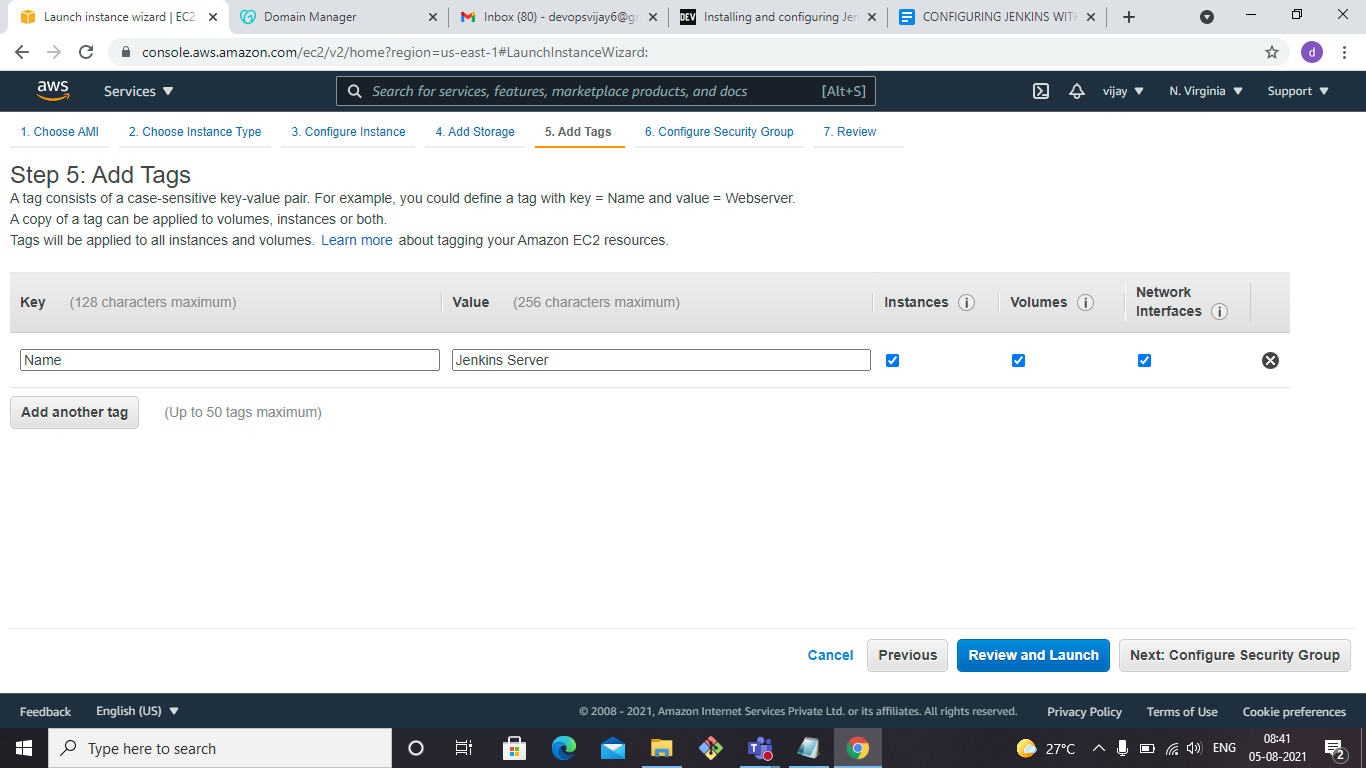
Configure the instance to suit your requirements.

### **Selecting the storage**

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage.I have gone with default 8GB

### **Adding Tags**

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.



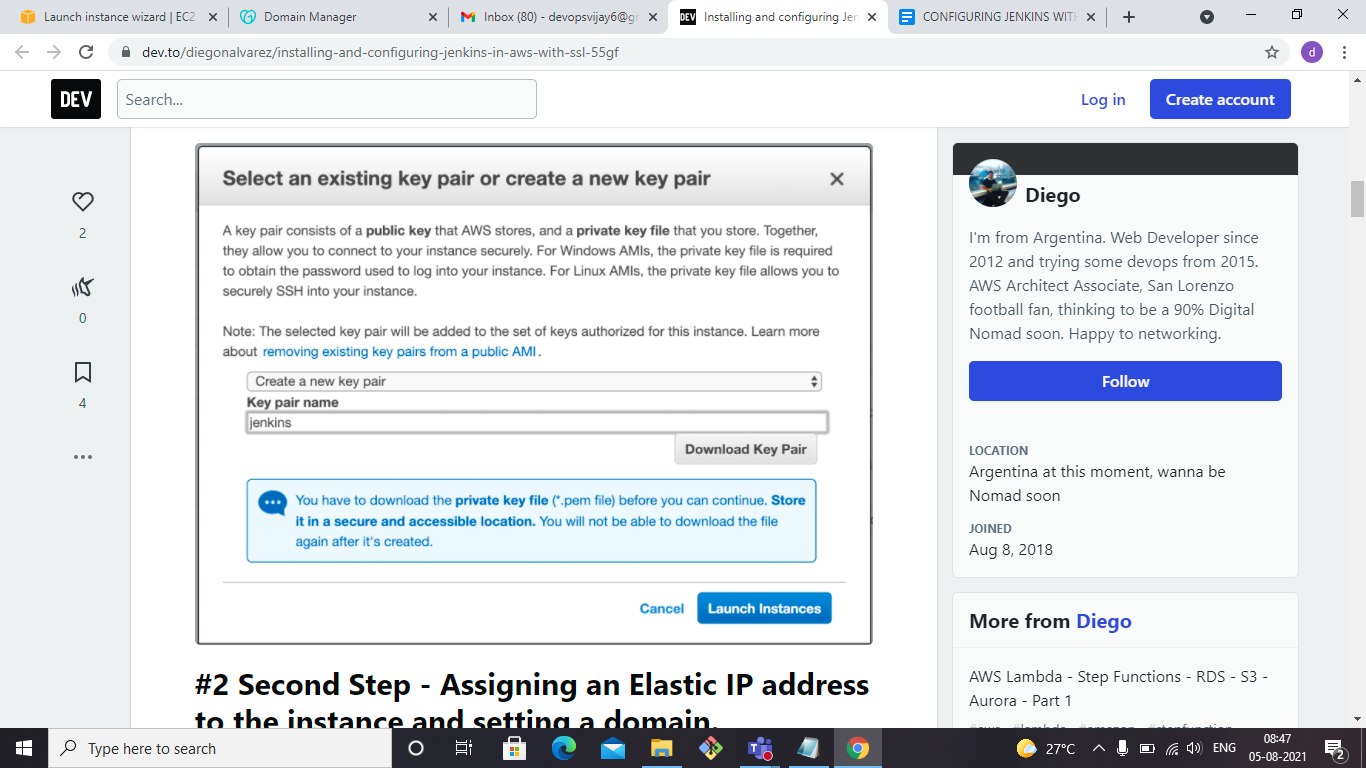
### **Security Group**

### Here set the firewall rules, to allow access to your instance

Add ports 22,8080,80,443

### **Review and launch**

In the last step you can review the total configuration and set the key pair to connect through SSH to your instance. We going to use this key pair in the next steps.



## 

## **#2 Second Step-connect the instance via SSH using Git Bash**

-------------------------------------------------------------------------------------------------------------------------------

## You can use this command in your terminal to access the instance via SSH:

ssh -i "jenkins.pem" [ubuntu@ec2-3-223-140-206.compute-1.amazonaws.com](mailto:ubuntu@ec2-3-223-140-206.compute-1.amazonaws.com)

## **#3 Fourth Step - Installing Jenkins**

--------------------------------------------------------------------------

Execute the below commands in your Server

sudo apt-get update -y

sudo apt install openjdk-8-jre-headless -y

sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

sudo apt-add-repository "deb http://pkg.jenkins-ci.org/debian binary/"

sudo apt update -y

sudo apt install jenkins -y

**Install Certbot:-**

Certbot is a letsencrypt client, we need to download.certbot can automatically configure NGINX for SSL/TLS.The Certbot packages on your system come with a cron job or systemd timer that will renew your certificates automatically before they expire.

sudo add-apt-repository ppa:certbot/certbot

sudo apt-get update

sudo apt-get install python3-certbot-nginx

Sudo apt-get install nginx -y

## 

## 

## **#4 Fourth Step-Godaddy entry for Jenkins Server**

--------------------------------------------------------------------------------------------------------

My Domain name is Registered in GODADDY.so,

- Create a DNS record that associates your domain name and your server’s public IP address.

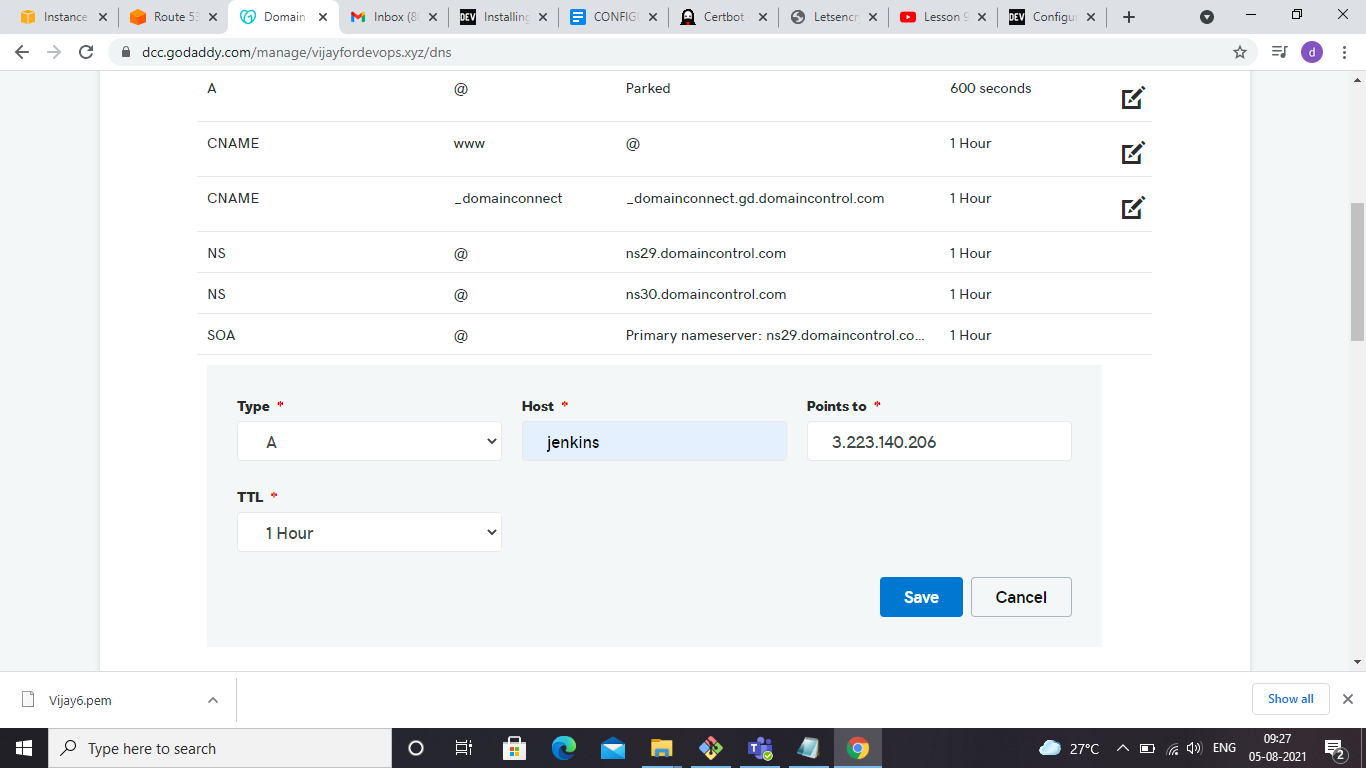
- Add "A" record for subdomain in DNS manage record on Godaddy console.

Go to DNS Management,click on Add

type:- A

host:- jenkins

points to:- <Jenkins server’s public-ip>



## 

## 

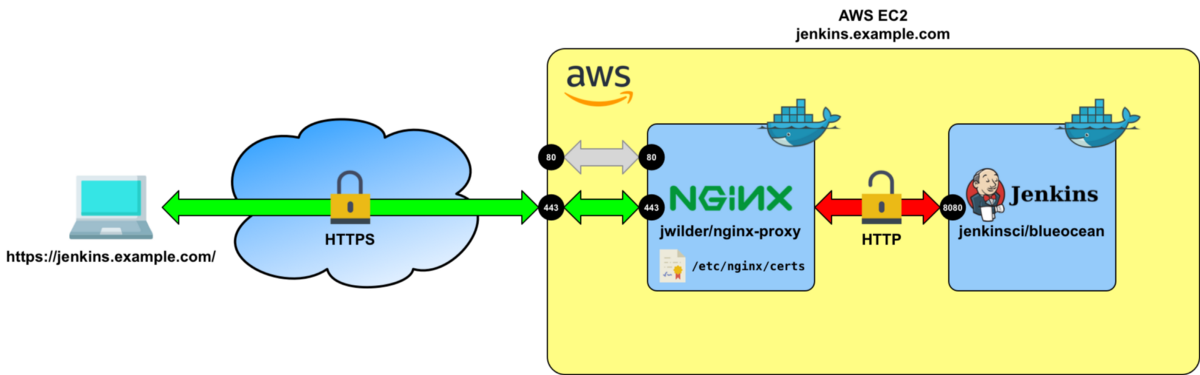
## 

## **#5 Sixth Step - Nginx configuration with SSL**

----------------------------------------------------------------------------------------------

- In situations where you have existing web sites on your server, you may find it useful to run Jenkins (or the servlet container that Jenkins runs in) behind [Nginx](https://nginx.org/), so that you can bind Jenkins to the part of a bigger website that you may have.

- When a request arrives for certain URLs, Nginx becomes a proxy and further forward that request to Jenkins, then it forwards the response back to the client.



go to

cd /etc/nginx/sites-available

make backup of default file:-

cd /etc/nginx/sites-available/

sudo mv default default.backup

sudo vi default (paste the below content in the default file)

server {

listen 80;:wq

server\_name jenkins.vijayfordevops.xyz;

location / {

proxy\_set\_header Host $host:$server\_port;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

# Fix the "It appears that your reverse proxy set up is broken" error.

proxy\_pass http://127.0.0.1:8080;

proxy\_read\_timeout 90;

proxy\_redirect http://127.0.0.1:8080 https://jenkins.vijayfordevops.xyz;

# proxy\_redirect http:// https://;

# Required for new HTTP-based CLI

proxy\_http\_version 1.1;

proxy\_request\_buffering off;

}

}

sudo nginx -t --(for verifying syntax)

sudo systemctl restart nginx

## **#6 Sixth Step-Change Jenkins bind address**

----------------------------------------------------------------------------------------------

By default Jenkins listens on all network interfaces. But we need to disable it because we are using Nginx as a reverse proxy and there is no reason for Jenkins to be exposed to other network interfaces.

We can change this by editing /etc/default/jenkins

sudo vi /etc/default/jenkins

Locate the line starting with JENKINS\_ARGS (It’s usually the last line) and append

--httpListenAddress=127.0.0.1

JENKINS\_ARGS="--webroot=/var/cache/$NAME/war --httpPort=$HTTP\_PORT --httpListenAddress=127.0.0.1"

sudo systemctl restart jenkins

sudo systemctl status jenkins

## **#7 Sixth Step-Install SSL Certificates by using Letsencrypt**

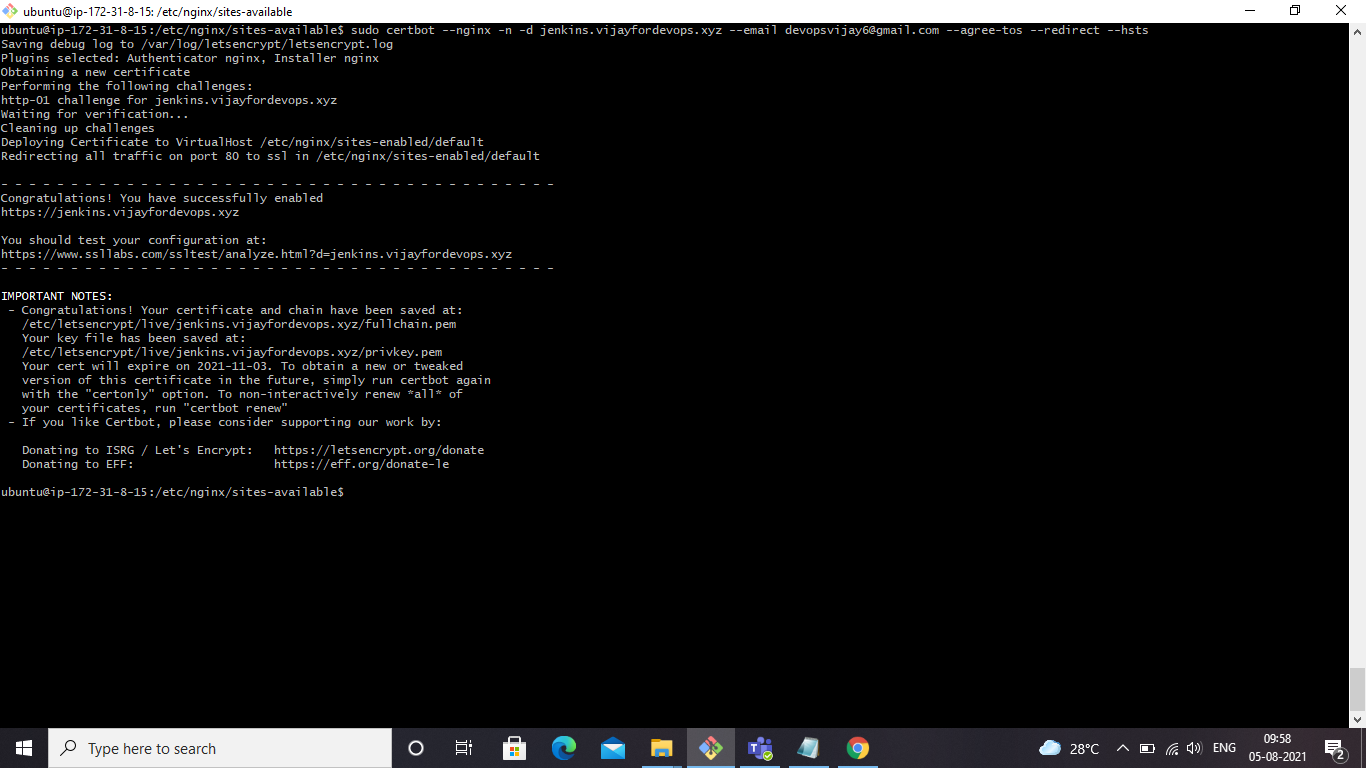
----------------------------------------------------------------------------------------------------------------------------

sudo certbot --nginx -n -d jenkins.vijayfordevops.xyz --email devopsvijay6@gmail.com --agree-tos --redirect --hsts

(it is NGINX plug‑in for certbot to generate certificates)

## How Let’s Encrypt Works:-

Before issuing a certificate, Let’s Encrypt validates ownership of your domain. The Let’s Encrypt client, running on your host, creates a temporary file (a token) with the required information in it. The Let’s Encrypt validation server then makes an HTTP request to retrieve the file and validates the token, which verifies that the DNS record for your domain resolves to the server running the Let’s Encrypt client.



Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/default

Redirecting all traffic on port 80 to ssl in

/etc/nginx/sites-enabled/default

Note:-

certbot can automatically configure NGINX for SSL/TLS

You can open sudo vi /etc/nginx/sites-available/default file

and check ssl certificates are automatically configured by Nginx

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/jenkins.vijayfordevops.xyz/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/jenkins.vijayfordevops.xyz/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

## **#8 Sixth Step-Check HTTPS**

------------------------------------------------------------

goto

internet explorer browser and Type

<https://jenkins.vijayfordevops.xyz>

