Install an ASP.NET Core Web API on Linux (Ubuntu 18.04) and host with Nginx and Install PostgreSQL on Ubuntu 20.04

Step 1

Install the .NET SDK or the .NET Runtime on Ubuntu 20.10

Open a terminal and run the following commands:

```
wget https://packages.microsoft.com/config/ubuntu/20.10/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
```

Install the SDK

```
sudo apt-get update; \
  sudo apt-get install -y apt-transport-https && \
  sudo apt-get update && \
  sudo apt-get install -y dotnet-sdk-3.1
```

Install the runtime

```
sudo apt-get update; \
  sudo apt-get install -y apt-transport-https && \
  sudo apt-get update && \
  sudo apt-get install -y aspnetcore-runtime-3.1
```

```
sudo apt-get install -y dotnet-runtime-3.1
```

Step 2

Install PostgreSQL on Ubuntu 20.04

To install PostgreSQL, first refresh your server’s local package index:

```
sudo apt update
```

Then, install the Postgres package along with a -contrib package that adds some additional utilities and functionality:

```
sudo apt install postgresql postgresql-contrib
```

Another way to access the PostgreSQL prompt without switching users, is to use the sudo command:

```
sudo -u postgres psql
```

To exit out of the PostgreSQL prompt, run the following:

```
Postgres=#       \q
```

Create a new PostgreSQL role:

```
sudo su - postgres -c "createuser  name_of_user"
```

Create a new PostgreSQL Database:

```
sudo su - postgres -c " createdb name_of_db"
```

To grant permissions to the user on the database, connect to the PostgreSQL shell:

```
sudo -u postgres psql
```

and run the following query:

```
grant all privileges on database name_of_db to name_of_user;
```

Enable Remote Access to PostgreSQL server

Open path this => /etc/postgresql/12/main/postgresql.conf

paste this commit

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------
# - Connection Settings –

listen_addresses = '*'     # what IP address(es) to listen on;
```

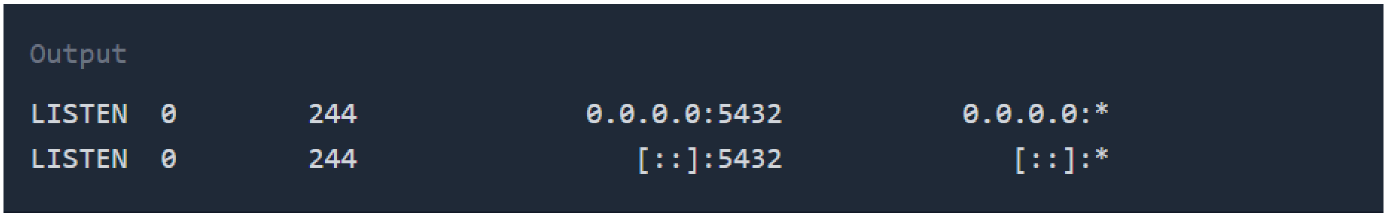
Save the file and restart the PostgreSQL service:

```
sudo service postgresql restart
```

Verify the changes with the ss utility:

```
ss -nlt | grep 5432
```

The output shows that the PostgreSQL server is listening on all interfaces (0.0.0.0):



```
Output
LISTEN  0       244             0.0.0.0:5432          0.0.0.0:*
LISTEN  0       244                [::]:5432             [::]:*
```

The next step is to configure the server to accept remote connections by editing the pg_hba.conf file.

Below are some examples showing different use cases:

```
/etc/postgresql/12/main/pg_hba.conf
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# The user jane can access all databases from all locations using md5 password
host    all             zaid            0.0.0.0/0               md5

# The user jane can access only the janedb from all locations using md5
password
host    zaiddb          zaid            0.0.0.0/0               md5

# The user jane can access all databases from a trusted location (192.168.1.134)
without a password
host    all             zaid            192.168.1.134           trust
```

The last step is to open the port 5432 in your firewall.

Assuming you are using UFW to manage your firewall, and you want to allow access from the 192.168.1.0/24 subnet, you would run the following command:

```
sudo ufw allow proto tcp from 192.168.1.0/24 to any port 5432
```

You can change the password of any user (including postgres) with:

```
ALTER USER name_of_user WITH PASSWORD 'my_password';
```

step 3

Install Nginx

Run the following commands to install Nginx

```
sudo apt update

sudo apt install nginx
```

restart of your Nginx

```
sudo systemctl restart nginx
```

You can check the status of your Nginx to make sure it’s got an active running status:

```
sudo systemctl status nginx
```

Step 4

Host ASP.NET Core Web API

1. create new folder in this path => var/www/aspcore/ new_folder
2. upload files a publish asp.net core in new_folder

3- Configure your Nginx server block

Open this path => /etc/nginx/sites-available/default

Your service block should look as follows:

```
server {
   listen        80;
   #server_name   {api.your-domain-name.com};
   root /var/www/aspcore/{your-folder-name};
   location / {
      proxy_pass         http://localhost:5000;
      proxy_http_version 1.1;
      proxy_set_header   Upgrade $http_upgrade;
      proxy_set_header   Connection keep-alive;
      proxy_set_header   Host $host;
      proxy_cache_bypass $http_upgrade;
      proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header   X-Forwarded-Proto $scheme;
   }
}
```

Note: -
listen   56 “or any port”;

4- After saving your config file you can start or restart your Nginx service

```
sudo service nginx restart
```

5- Create your Asp.Net service

Open this path => /etc/systemd/system/

Create the service file “filename.service”

Service file example:

```
[Unit]
Description=This is a sample application for my tutorial
[Service]
WorkingDirectory=/var/www/aspcore/new_floder
ExecStart=/usr/bin/dotnet  /var/www/aspcore/new_folder/Appapi.dll
Restart=always
# Restart service after 10 seconds if the dotnet service crashes:
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=dotnet-example
User=www-data
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
# If you need to run multiple services on different ports set
# the ports environment variable here:
# Environment=ASPNETCORE_URLS=http://localhost:6000
[Install]
WantedBy=multi-user.target
```

7- Save the file and you can enable the service as follows:

```
sudo systemctl enable dms.service
```

8- Then you need to start your service:

```
sudo systemctl start dms.service
```

9- Check that it is running as follows:

```
sudo systemctl status dms.service
```

http://45.77.65.134
Server info
root
S6c%8Z#QE(4w)qHa
———————————————————-
Database info
ameer
ameer123@root

Sources

https://docs.microsoft.com/en-us/dotnet/core/install/linux-ubuntu

https://linuxize.com/post/how-to-install-postgresql-on-ubuntu-20-04/

https://itsfoss.com/install-postgresql-ubuntu/

https://medium.com/@hbhhathorn/install-an-asp-net-core-web-api-on-linux-ubuntu-18-04-and-host-with-nginx-and-ssl-2ed9df7371fb