**Installing .NET 8 on Ubuntu 22.04**

**Note: Apache does not need to be installed, as .NET can handle HTTP requests.**

1. **Install .NET Runtime** 
   1. sudo apt-get update
   2. sudo apt-get install -y aspnetcore-runtime-8.0
2. **Install MS SQL Server  
   Note: Set your password for SQL Server during MSSQL configuration  
   Note: Choose 'Developer Edition' (option 2) if prompted.**

curl -fsSL https://packages.microsoft.com/keys/microsoft.asc | sudo gpg --dearmor -o /usr/share/keyrings/microsoft-prod.gpg

curl https://packages.microsoft.com/keys/microsoft.asc | sudo tee /etc/apt/trusted.gpg.d/microsoft.asc  
  
curl -fsSL https://packages.microsoft.com/config/ubuntu/22.04/mssql-server-2022.list | sudo tee /etc/apt/sources.list.d/mssql-server-2022.list  
  
curl -fsSL https://packages.microsoft.com/config/ubuntu/22.04/mssql-server-2022.list | sudo tee /etc/apt/sources.list.d/mssql-server-2022.list  
  
sudo apt-get update

sudo apt-get install -y mssql-server  
  
sudo /opt/mssql/bin/mssql-conf setup  
  
systemctl status mssql-server --no-pager

* 1. Troubleshooting log: ‘/var/opt/mssql/log/’

1. **Install SQL Server Command-Line Tools:**

curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add - && \

curl https://packages.microsoft.com/config/ubuntu/22.04/prod.list | sudo tee /etc/apt/sources.list.d/mssql-release.list && \

sudo apt-get update && \

sudo apt-get install -y mssql-tools unixodbc-dev && \

sudo ln -s /opt/mssql-tools/bin/sqlcmd /usr/local/bin/sqlcmd

1. **Publish Website files for Linux:**
   1. Navigate to the website files directory
   2. Publish files: ( dotnet publish -c Release -r linux-x64 )
   3. Published files will be saved on: \bin\Release\net8.0\linux-x64\publish\
2. **Generate SSL Certificate:**

sudo apt update && sudo apt install -y openssl

openssl genpkey -algorithm RSA -out private-key.pem -aes256

openssl req -new -key private-key.pem -out certificate.csr

openssl x509 -req -days 365 -in certificate.csr -signkey private-key.pem -out certificate.crt

openssl pkcs12 -export -out certificate.pfx -inkey private-key.pem -in certificate.crt

1. **Configure Database Connection String and HTTP Requests in appsettings.json File, Based on Server Details:**
   1. Replace the relevant sections in the appsettings.json file in the root directory of the published file **Tip: Default ‘User Id’ in connection-string is ‘sa’**

{

"Kestrel": {

"Endpoints": {

"Http": {

"Url": "http://\*:80"

},

"Https": {

"Url": "https://\*:443",

"Certificate": {

"Path": "/root/certificate.pfx",

"Password": " **your\_password** "

}

}

}

},

"ConnectionStrings": {

"DefaultConnection": "Server=127.0.0.1; Database=DeployOnLinuxDB; User Id= **your\_username**; Password= **your\_password**;"

}

}

1. **Transfer Website Published Files to the Server**
   1. Published files will be saved on: \bin\Release\net8.0\linux-x64\publish\
2. **Upload and Attach the Database to the SQL Server**
   1. Copy the Database backup file to the server
   2. Connect to the SQL Server ( sqlcmd -S localhost -U SA -P '<YourPassword>' -C )
   3. Restore The Database:  
      **Note: in this example, the backup file path is ‘/var/www/’**

RESTORE DATABASE [DeployOnLinuxDB]

FROM DISK = N'/var/www/DeployOnLinuxDB.bak'

WITH MOVE 'DeployOnLinuxDB' TO '/var/opt/mssql/data/DeployOnLinuxDB.mdf',

MOVE 'DeployOnLinuxDB\_Log' TO '/var/opt/mssql/data/DeployOnLinuxDB\_log.ldf',

REPLACE;

GO

* 1. Verify the Restoration:

USE [DeployOnLinuxDB];

GO

SELECT \* FROM sys.tables;

GO

1. **Create a Service and Start It**
   1. sudo nano /etc/systemd/system/dotnet-app.service
   2. write the configuration into the file:

[Unit]

Description=Dotnet Application

[Service]

User=root

WorkingDirectory=/var/www/html

ExecStart=/usr/bin/dotnet /var/www/html/DeployOnLinux.dll

Restart=always

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=dotnet-app

Environment=ASPNETCORE\_ENVIRONMENT=Production

[Install]

WantedBy=multi-user.target

* 1. Start and enable the service:

sudo systemctl daemon-reload

sudo systemctl start dotnet-app.service

sudo systemctl enable dotnet-app.service