# 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**
    a. sudo apt-get update
    b. 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
```

    a. Troubleshooting log: '/var/opt/mssql/log/'

3. **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
```

4. **Publish Website files for Linux:**
    a. Navigate to the website files directory
    b. Publish files: ( dotnet publish -c Release -r linux-x64 )

c. Published files will be saved on: \bin\Release\net8.0\linux-x64\publish\

**5. 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
```

6. **Configure Database Connection String and HTTP Requests in appsettings.json File, Based on Server Details:**
    a. Replace the relevant sections in the appsettings.json file in the root directory of the published file

```
{
  "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;"
  }
}
```

**Tip: Default 'User Id' in connection-string is 'sa'**

7. **Transfer Website Published Files to the Server**
    a. Published files will be saved on: \bin\Release\net8.0\linux-x64\publish\

8. **Upload and Attach the Database to the SQL Server**
    a. Copy the Database backup file to the server
    b. Connect to the SQL Server ( sqlcmd -S localhost -U SA -P '<YourPassword>' -C )

c. 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
```

d. Verify the Restoration:

```
USE [DeployOnLinuxDB];
GO
SELECT * FROM sys.tables;
GO
```

9. **Create a Service and Start It**
    a. sudo nano /etc/systemd/system/dotnet-app.service
    b. 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
```

    c. Start and enable the service:

```
sudo systemctl daemon-reload
sudo systemctl start dotnet-app.service
sudo systemctl enable dotnet-app.service
```