Host ASP.NET Core on Rocky/RHEL (Apache reverse proxy)

Introduction

This is to prep the Linux server to host Visual Studio code.

This is for either RedHat or Rocky Linux installations.

For this configuration were are installing on a Rocky Linux server.

Platform prep: This page covers platform preparation (runtime, Apache reverse proxy, and systemd basics). For on-prem ADO pipelines, secrets, and troubleshooting, see the sibling page: On-prem ADO → Apache/.NET Kestrel deploy runbook.

Preparation

Make sure that the server is up to date

```
dnf update -y
```

Configuration

Install the ASP.NET Core runtime for the target major version (8, 9, or 10)

```
dnf install aspnetcore-runtime-8.0 -y
```

Verification

```
dotnet --info
```

This should bring back the installation results

```
Host:
  Version:      8.0.8
  Architecture: x64
  Commit:       08338fcaa5
  RID:          rocky.9-x64
.NET SDKs installed:
  No SDKs were found.
.NET runtimes installed:
  Microsoft.AspNetCore.App 8.0.8 [/usr/lib64/dotnet/shared/Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 8.0.8 [/usr/lib64/dotnet/shared/Microsoft.NETCore.App]
Other architectures found:
  None
Environment variables:
  DOTNET_ROOT       [/usr/lib64/dotnet]
global.json file:
  Not found
Learn more:
  https://aka.ms/dotnet/info
Download .NET:
  https://aka.ms/dotnet/download
```

Setup the Project

You will need to SSH to to the Linux server and navigate to the deployment folder.

In our case the project is an API project that is called MyFFLBookAPI

Once in the folder you can start the project this way

```
cd /var/www/html/api
dotnet MyFFLBookAPI.dll
```

```
[/var/www/html/api]# dotnet MyFFLBookAPI.dll
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
      No XML encryptor configured. Key {3cacc8f4-baca-4475-9498-610b5e187653} may be persisted to storage in
unencrypted form.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /var/www/html/api
```

As you can see the server started on localhost port 5000. If you wish you can change the port to another if needed.

Simply use the Ctrl+C key to shutdown the project

Change Port

You will need to add an entry for Kestrel to redirect the port after the allowed hosts entry

```
  "AllowedHosts": "*",
   "Kestrel": {
        "Endpoints": {
        "Http": {
          "Url": "http://*:8081"
        }
        }
    }
```

Now to test restart the project

```
dotnet MyFFLBookAPI.dll
```

```
[/var/www/html/api]# dotnet MyFFLBookAPI.dll
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:8081
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /var/www/html/api
```

Now you can see that is started on the new port 8081

Setup a Service

First you will have the create a unit file for the service by doing the following

```
vi /lib/systemd/system/myfflbookapi.service
```

then for the entries within the unit service file

```
[Unit]
Description=MyFFLBookAPI
```

```
[Service]
WorkingDirectory=/var/www/html/api
ExecStart=/usr/bin/dotnet /var/www/html/api/MyFFLBookAPI.dll
Restart=always
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=myfflbookapi
User=aspnet
Environment=ASPNETCORE_ENVIRONMENT=Production

[Install]
WantedBy=multi-user.target
```

Once configured you can test the service

To start the service do the following

```
systemctl start myfflbookapi
```

Then verify the status of the service

```
systemctl status myfflbookapi
```

Results of the above command

```
* myfflbookapi.service - MyFFLBookAPI
     Loaded: loaded (/usr/lib/systemd/system/myfflbookapi.service; disabled; preset: disabled)
     Active: active (running) since Sat 2024-08-31 10:14:38 EDT; 6s ago
   Main PID: 59786 (dotnet)
      Tasks: 28 (limit: 408004)
     Memory: 40.5M
        CPU: 515ms
     CGroup: /system.slice/myfflbookapi.service
             `-59786 /usr/bin/dotnet /var/www/html/api/MyFFLBookAPI.dll
Aug 31 10:14:38 web01.example.com systemd[1]: Started MyFFLBookAPI.
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]: info: Microsoft.Hosting.Lifetime[14]
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]:       Now listening on: http://[::]:8081
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]: info: Microsoft.Hosting.Lifetime[0]
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]:       Application started. Press Ctrl+C to shut down.
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]: info: Microsoft.Hosting.Lifetime[0]
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]:       Hosting environment: Production
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]: info: Microsoft.Hosting.Lifetime[0]
Aug 31 10:14:38 web01.example.com myfflbookapi[59786]:       Content root path: /var/www/html/api
```

As you can see the service is running.

Now you can set the service to auto start on boot

```
systemctl enable myfflbookapi
```

It will auto create a link for boot

```
Created symlink /etc/systemd/system/multi-user.target.wants/myfflbookapi.service ->
/usr/lib/systemd/system/myfflbookapi.service.
```

Redirecting Ports

in the "/etc/httpd/sites-available" folder you will need to create a file for the website, in our case we will be using "sflservicesllc.io.conf"

```
cd /etc/httpd/sites-available
```

Then create the following file

```
vi sflservicesllc.io.conf
```

Add the following in the file

```
<VirtualHost *:80>

        ServerName sflservicesllc.io
        ServerAlias www.sflservicesllc.io
        ProxyPreserveHost On
        ProxyPass / http://127.0.0.1:8081/
        ProxyPassReverse / http://127.0.0.1:8081/

        ErrorLog logs/API_error_log
        TransferLog logs/API_access_log
        RewriteEngine on
        RewriteCond %{SERVER_NAME} =sflservicesllc.io
        RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]

</VirtualHost>
<VirtualHost *:443>

        SSLProxyEngine On
        SSLProxyVerify none
        SSLProxyCheckPeerCN off
        SSLProxyCheckPeerName off
```

```
        SSLProxyCheckPeerExpire off
        #SSLProxyCACertificateFile

        ServerName sflservicesllc.io
        ServerAlias www.sflservicesllc.io
        ProxyPreserveHost On
        ProxyPass / http://127.0.0.1:8081/
        ProxyPassReverse / http://127.0.0.1:8081/

        ErrorLog logs/ssl_API_error_log
        TransferLog logs/ssl_API_access_log

</VirtualHost>
```

Now create a link to allow Apache to start the website

```
cd /etc/httpd/sites-enabled
ln -s /etc/httpd/sites-available/sflservicesllc.io.conf
```

Then restart the Apache web service

```
systemctl restart httpd
```

House notes

- Document ASP.NET Core runtimes 8, 9, and 10; the app host currently has 8.0.30, 9.0.19, and 10.0.11. This is not Visual Studio-only guidance or .NET 8-only guidance.
- Behind Apache, proxy to `http://127.0.0.1:PORT`; TLS terminates at the edge Apache. Do not default to `https://localhost:8081`.
- Prefer a dedicated `User=` account in systemd; `User=root` is not for production.
- The Apache path is `/etc/httpd/sites-available` (not `site-available`).
- MyFFLBookAPI is one worked example, not the house standard for every API.
- If the TLS edge uses `ProxyPass "/.well-known/" "!"` for ACME, narrow the exclusion to `/.well-known/acme-challenge/`; a blanket exclusion breaks OpenID discovery/JWKS.

Other Materials

You can also find on the Microsoft web site how to install on different versions other then the ones mentioned here

Install the .NET SDK or the .NET Runtime on RHEL and CentOS Stream

Note: .NET 8 and .NET 9 reach end of support on November 10, 2026. .NET 10 (LTS) is supported until November 14, 2028.

Last verified: October 2026

Sources:

- https://devblogs.microsoft.com/dotnet/dotnet-8-9-end-of-support/
- https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core

---

Revision #1
Created 31 August 2024 12:50:42 by Admin
Updated 24 October 2026 14:10:46 by Admin