**Publishing AspNet Core App**

First, we need to publish the app to a folder directory (you can choose any one).

Lets use directory **C:\core-app**.

- For angular app, make sure the follow code has been written in “app-web.csproj”:

*<Target Name="Build Angular" Condition="'$(Configuration)'=='Release'" BeforeTargets="Build">*

*<Message Text="\* \* \* \* \* \* Building Angular App \* \* \* \* \* \*" Importance="high" />*

*<Exec Command="ng build --prod --aot" />*

*</Target>*

If is Linux Server, copy the published files to a directory in Linux Server, lets call **/opt/aspnetcore-app/**. And the startup file of the project must contains the following lines:

*app.UseForwardedHeaders(new ForwardedHeadersOptions*

*{*

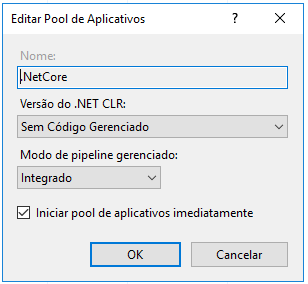
*ForwardedHeaders = ForwardedHeaders.XForwardedFor | ForwardedHeaders.XForwardedProto*

*});*

**IIS (Windows)**

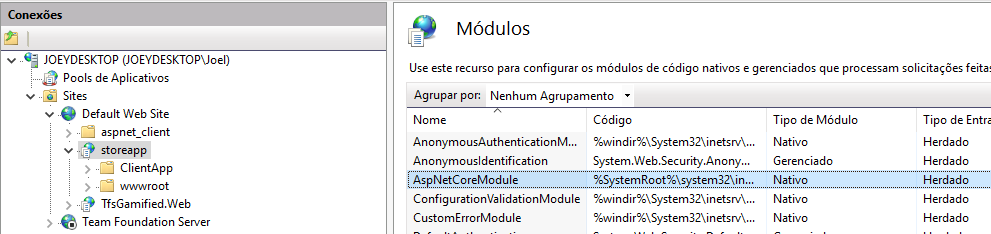
To host aspnet core app on IIS we need to install dotNet Core Rutime (Windows hosting 2.1.1) (since the app was built on 2.1.3v). Available at: https://www.microsoft.com/net/download/windows.

After that, on IIS Management, create a pool with no managed code like this:



Then, create an application to **C:\core-app** with this pool**.**

Make sure that exist an ***AspNetCoreModule*** in the App Modules.



And its done. You can check *“localhost/storeapp”* on browser.

**LINUX Server**

First, we gonna need to install dotNet CLI on Linux Server.

Before that we need to update de packages and the commands to use it depends of the SO version. A list of options can be found here: <https://docs.microsoft.com/pt-br/dotnet/core/linux-prerequisites?tabs=netcore2x>.

I used the follow:

***sudo sh -c 'echo "deb [arch=amd64] https://packages.microsoft.com/repos/microsoft-ubuntu-xenial-prod xenial main" > /etc/apt/sources.list.d/dotnetdev.list'***

***sudo apt-get update***

After that, install the dotnet CLI:

***sudo apt-get install dotnet-sdk-2.1.4***

dotnet Core Apps are “self running” through kestrel server and we need to create a service that will start the app using the dotNet CLI.

To do this, we need to create a file at **/etc/systemd/system** directory and lets call**: kestrel-aspnetcore.service.** Looks like this**: /etc/systemd/system/ kestrel-aspnetcore.service**

The file must contain the following lines:

*[Unit]*

*Description=Example .NET Web API App running on Ubuntu*

*[Service]*

*WorkingDirectory=/opt/aspnetcore-app*

*ExecStart=/usr/bin/dotnet /opt/aspnetcore-app/StoreApp.Web.dll*

*Restart=always*

*RestartSec=10 # Restart service after 10 seconds if dotnet service crashes*

*SyslogIdentifier=dotnet-example*

*User=www-data*

*Environment=ASPNETCORE\_ENVIRONMENT=Production*

*Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false*

*[Install]*

*WantedBy=multi-user.target*

Make sure the dll file has been written with all lower and upper cases that it has, otherwise the dotnet CLI will not work.

After that, use the following command to enable and start the service:

***systemctl enable kestrel-*aspnetcore*.service***

***systemctl start kestrel-*aspnetcore*.service***

***systemctl status kestrel-*aspnetcore*.service***

The service will execute the dll file and the AspNet Core App will run through the port 5000. We gonna need a Web Server to redirect de requests from port 80 to <http://localhost:5000>. That is the reason we need the *UseForwardedHeaders* to keep headers integrity.

**(NGINX as Web Server)**

After install and start NGINX service with the following commands:

***sudo apt-get install nginx***

***sudo service nginx start***

We gonna need to re-write the file ***default*** at *the* ***/etc/nginx/sites-enabled directory***. Looks like this ***/etc/nginx/sites-enabled directory/default.*** With the folloing lines:

*server {*

*listen 80;*

*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;*

*}*

*}*

Restart NGINX server

***sudo service nginx restart***

Aand its done.