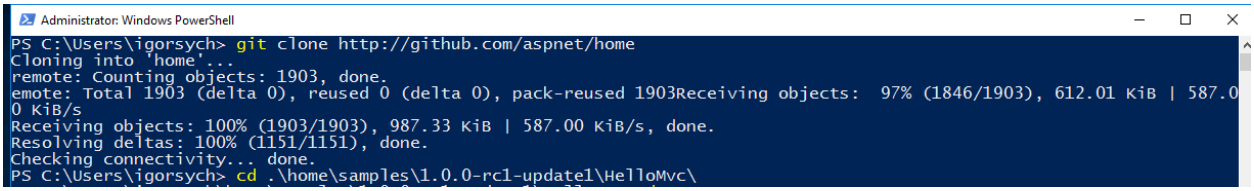
Для выполнения гайда вам понадобится либо Windows Server 2016 Preview5 либо Windows 10 Preview.

Exercise 1. Run Asp.Net Core on Host (without visual studio)

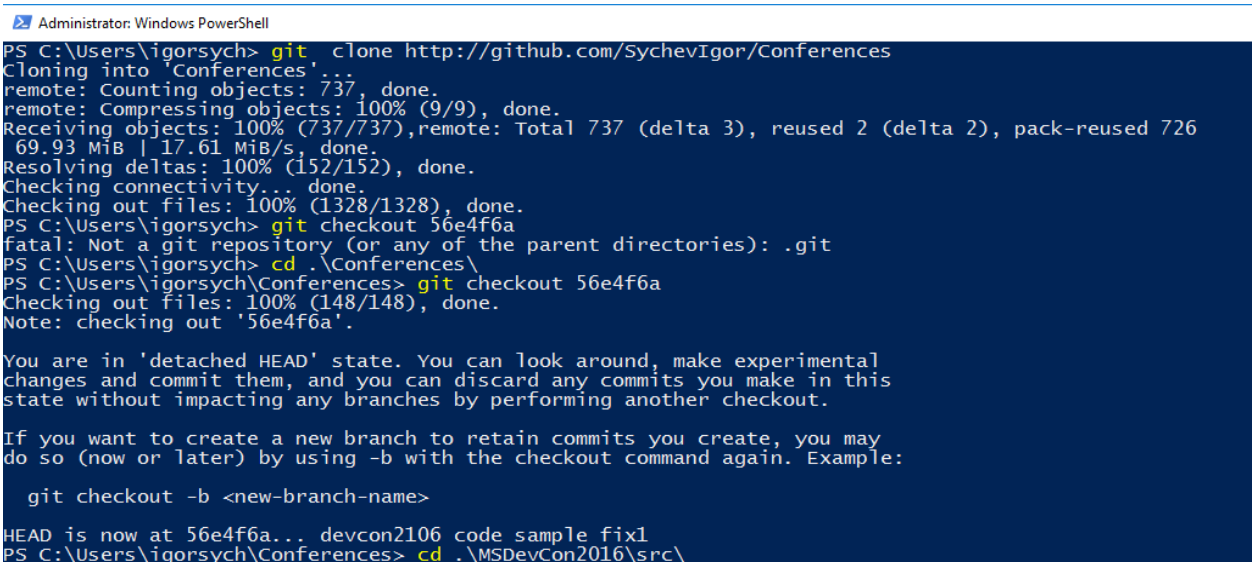
1. install chrome (optional step- edge is not working under admin account on srv2016 ctp5 by default)



2. install .net core and .net sdk https://www.microsoft.com/net/download

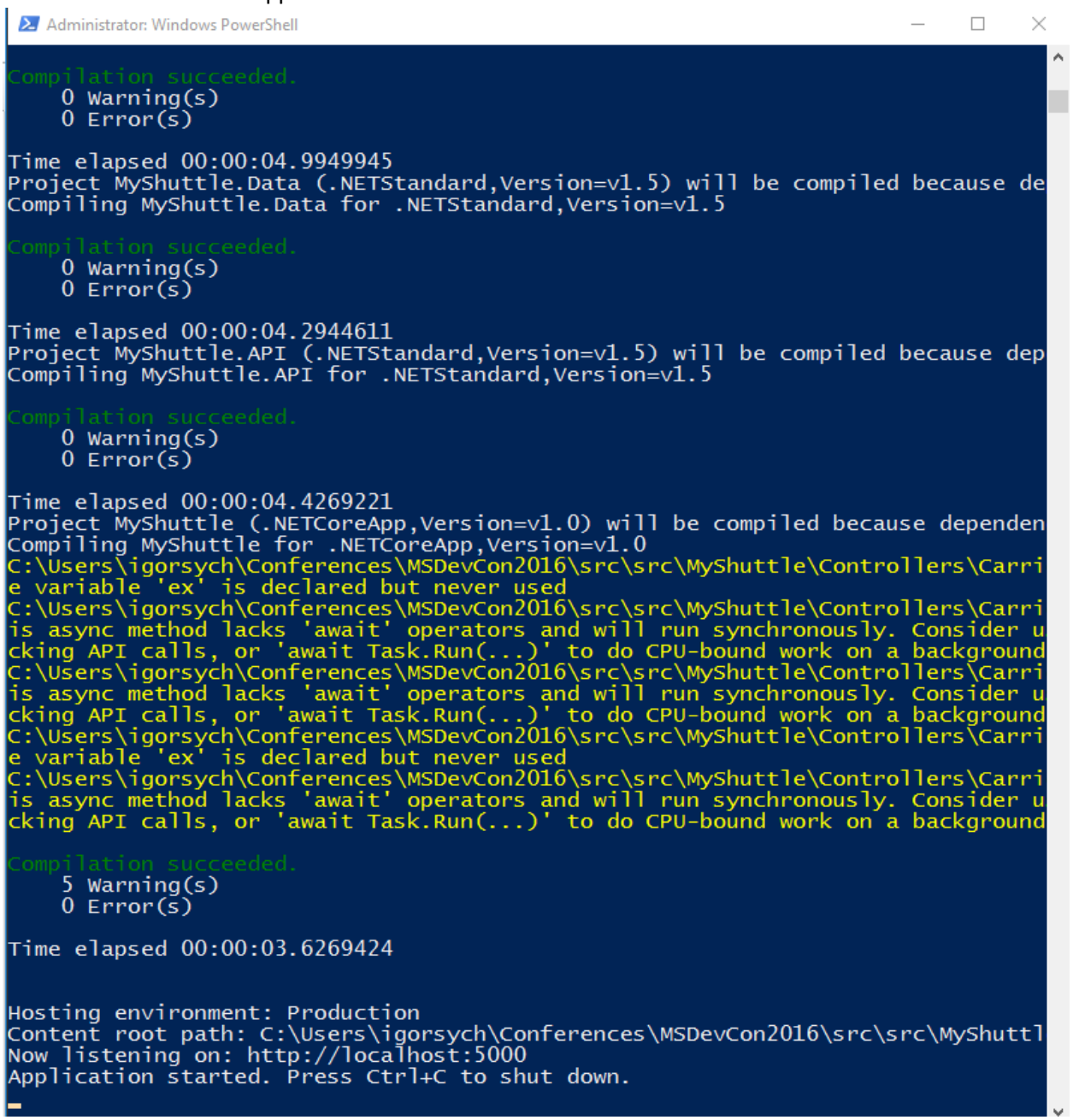
	.NET Core Installer (RC2)	.NET Core SDK Installer (Preview 1)
Windows	x64 / x86 .exe	x64 / x86 .exe
Windows (Server Hosting)	x64 / x86 .exe	N/A
Ubuntu 14.04	*soon:* "apt install dotnet"	
Debian 8.2	N/A	N/A
Mac OS X	x64 .pkg	x64 .pkg
CentOS 7.1	N/A	N/A
RHEL 7.2	*soon:* "yum install dotnet"	

3. install git https://git-scm.com/download/win
4. Нужно выбрать папку куда делать следующий шаг для клонирования репозитория.
5. Clone any asp.net project from github
6. For example, “git clone https://github.com/SychevIgor/conferences” (checkout master branch)



7.

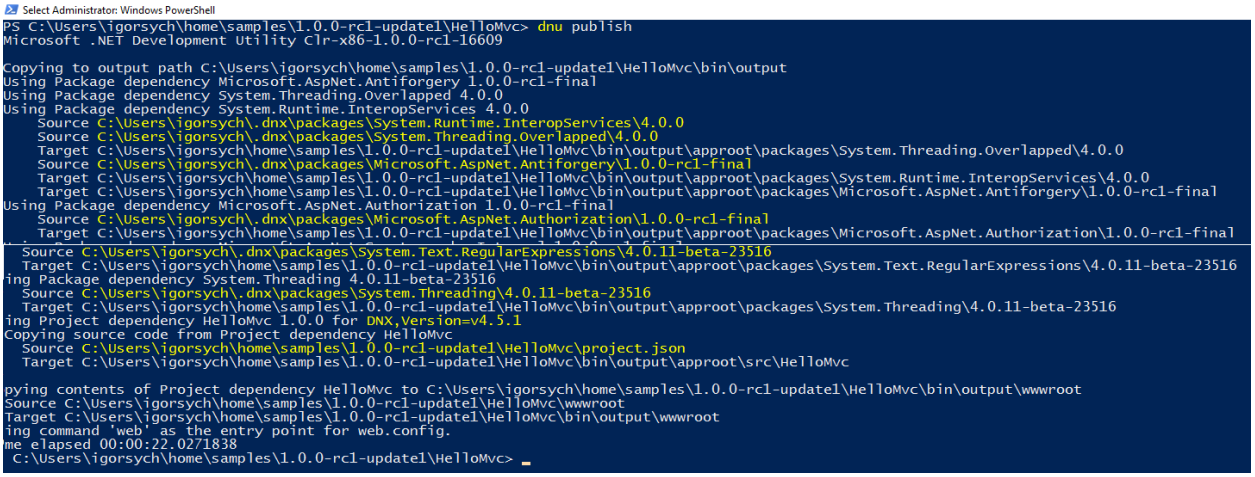
8. navigate to folder "conferences /MSDevCon2016/src " and restore packages in this folder "dotnet restore"
9. Navigate to MyShuttle folder and execute "dotnet build" to check that everything compiled correctly
10. "Dotnet run" to start an app



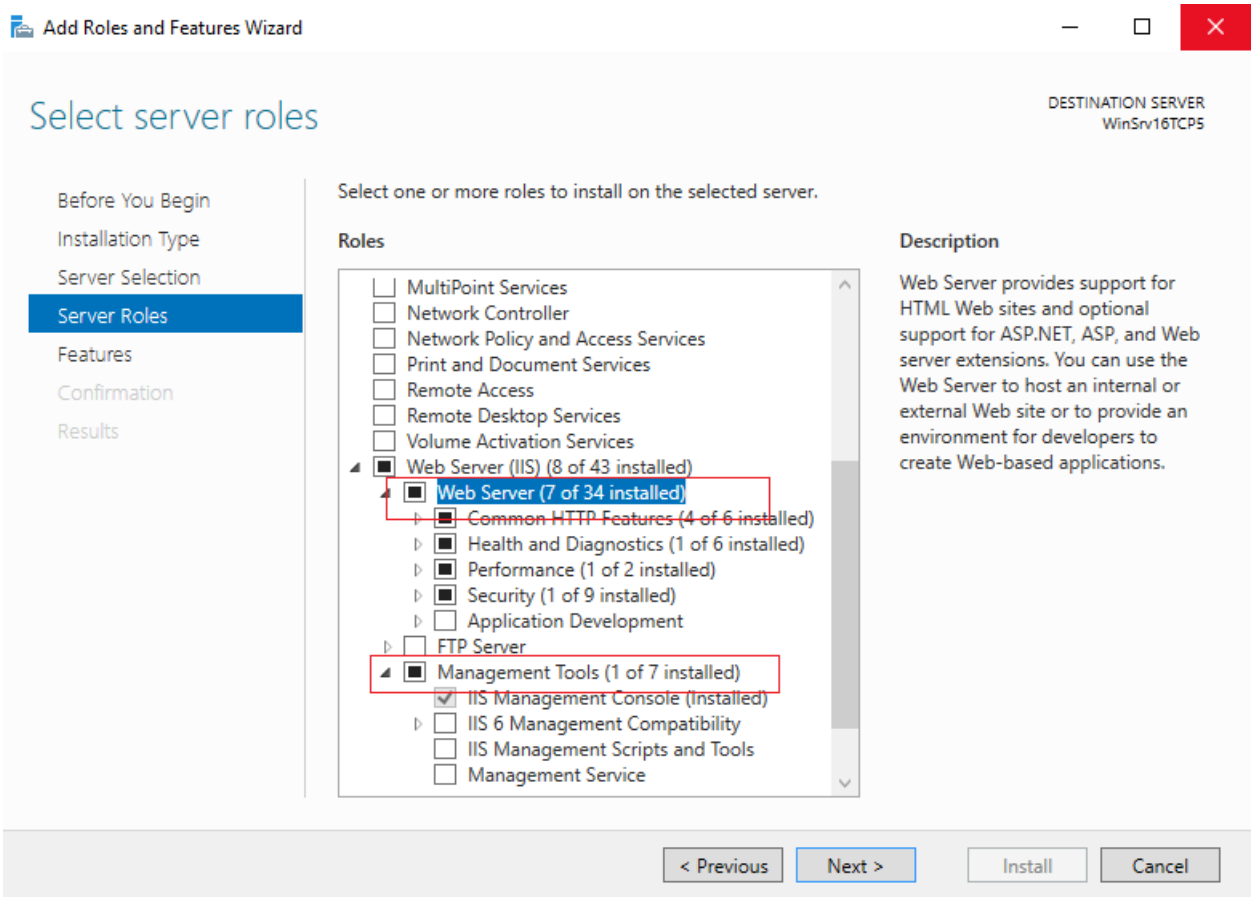
11. Open http://localhost:5000 and check that site is working

12. Press Ctrl-C to shut down the web server

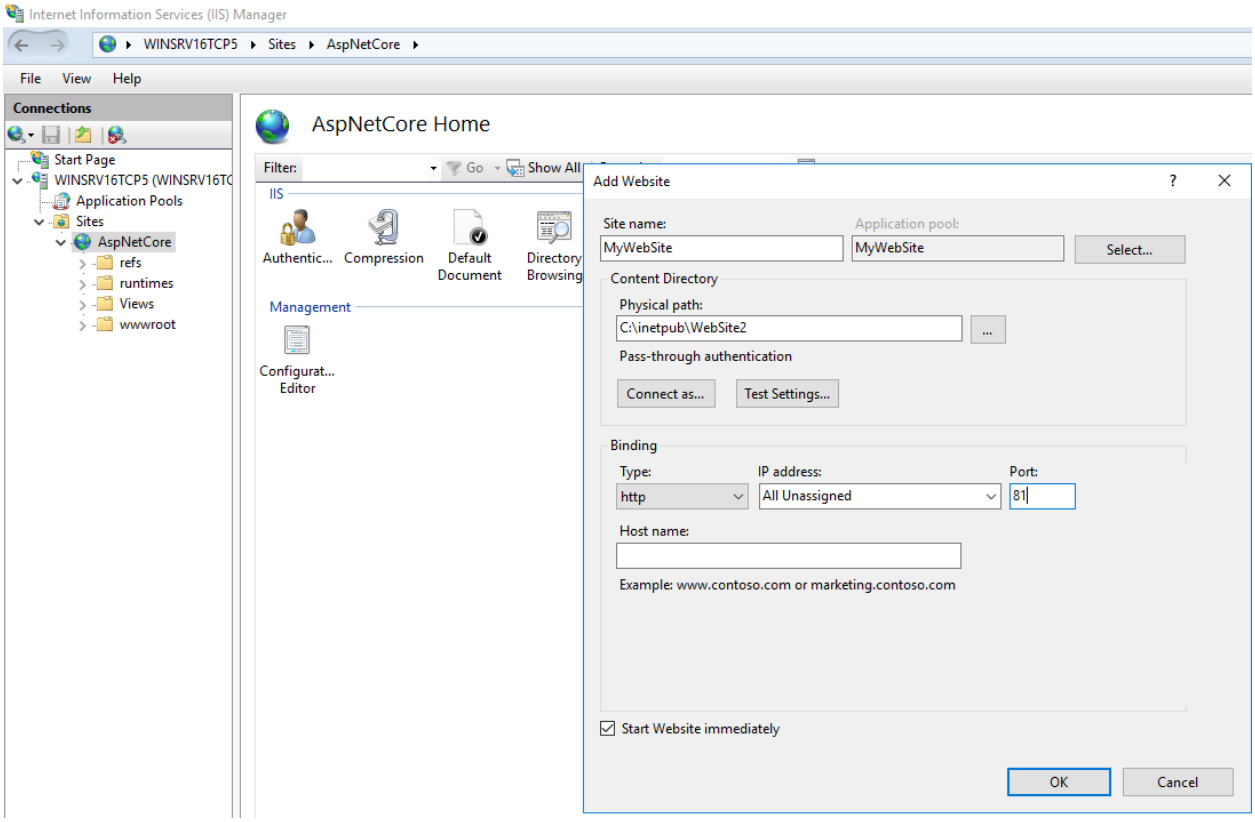
13. Publish artifacts "dotnet publish" to folder (если публиковать без параметров, то по умолчанию это будет сделано в текущем каталоге.).



14. Copy build artifacts from artifacts folder to destination folder.
15. Install Windows Server feature – web server (if not installed earlier)



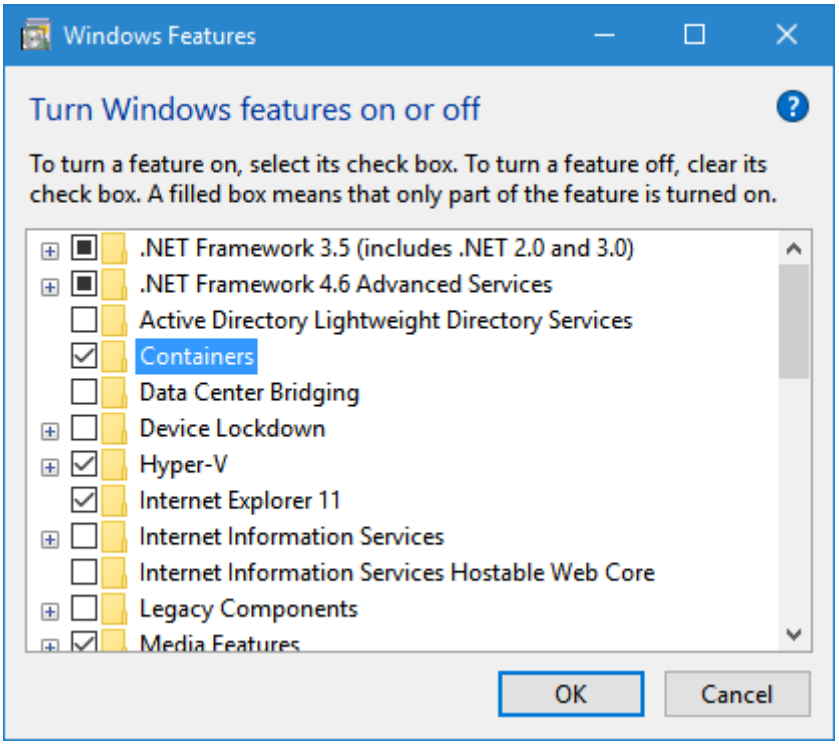
16. Start new app on IIS from this folder



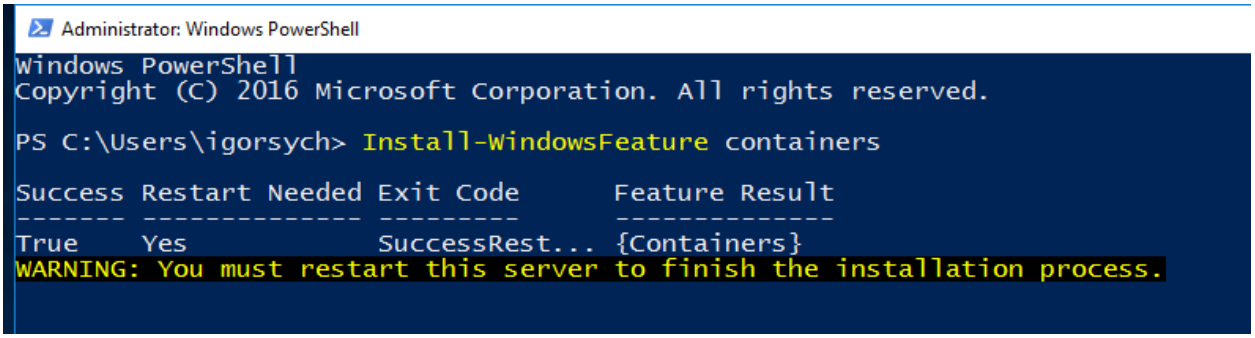
17. Open browser with http://localhost:yourport

Exercise 2. Prepare host machine to Container

1. Для клиентской Windows потребуется включить функцию руками – Control Panel – Uninstall a program – Turn Windows features on or off



2. Для Windows Server можно использовать PowerShell:



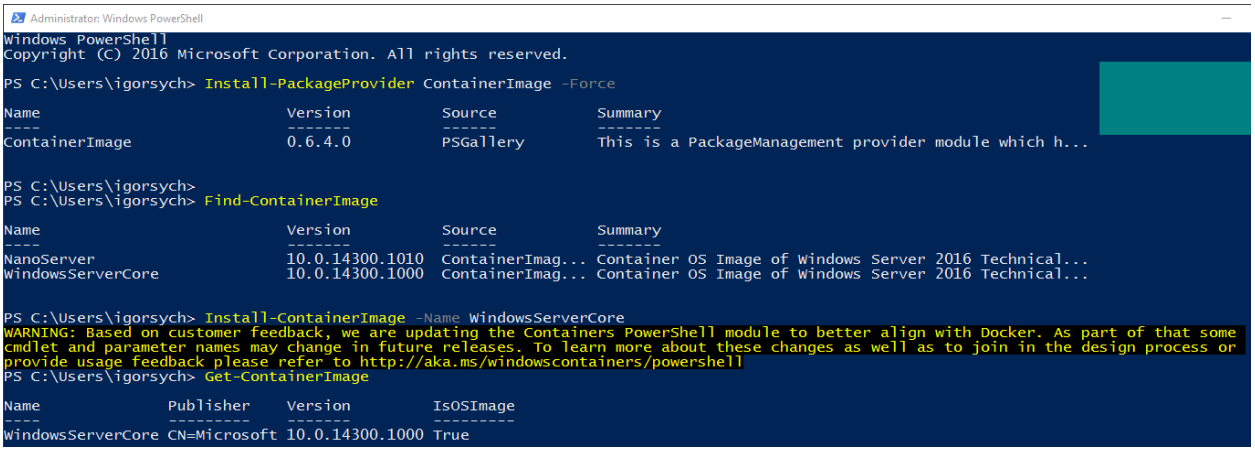
3. Prepare host machine to containers https://msdn.microsoft.com/en-us/virtualization/windowscontainers/deployment/deployment

```
Install-PackageProvider ContainerImage -Force
```

4. Then install ServerCore image. It will take time approximately 20 minutes to download this image.

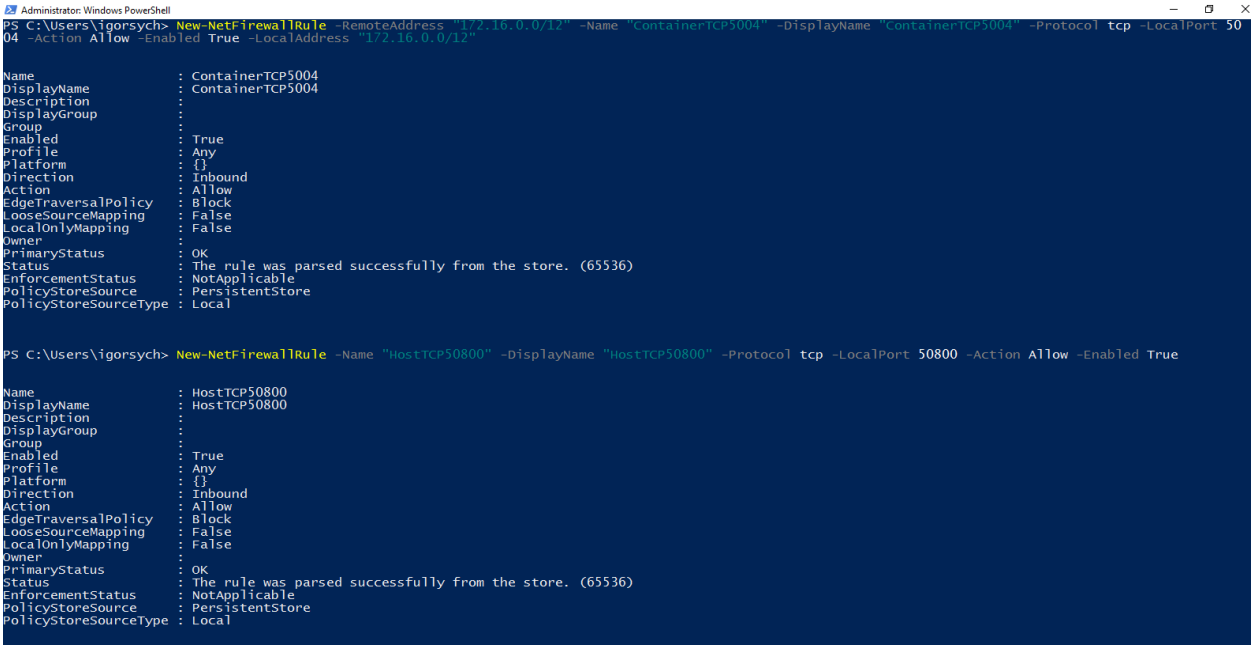
```
Find-ContainerImage
```

```
Install-ContainerImage -Name WindowsServerCore
```

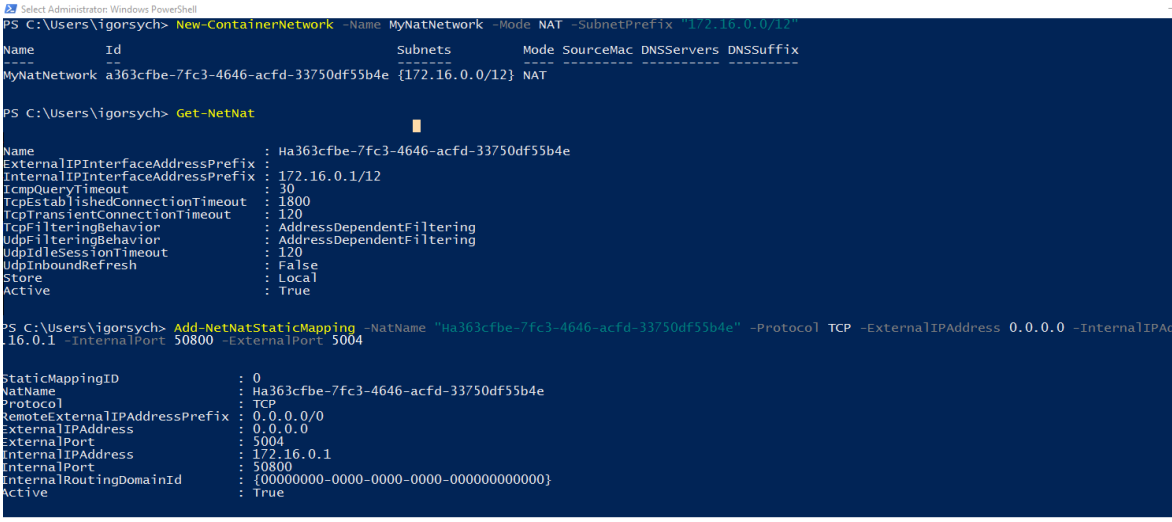


   a. if you are using servercore- do not install nanoserver image. Because container is a virtualized environment, windows images should share the same core.
   b. NanoServer and ServerCore are not sharing core. (but you can user nano server image with hyperv containers)

5. Configure network nat/firewall



   a. New-NetFirewallRule -RemoteAddress "172.16.0.0/12" -Name "ContainerTCP5000" -DisplayName "ContainerTCP5000" -Protocol tcp -LocalPort 5000 -Action Allow -Enabled True -LocalAddress "172.16.0.0/12". Make sure that you are using port 5000, because app will by default use port 5000
   b. New-NetFirewallRule -Name "HostTCP50800" -DisplayName "HostTCP50800" -Protocol tcp -LocalPort 50800 -Action Allow -Enabled True

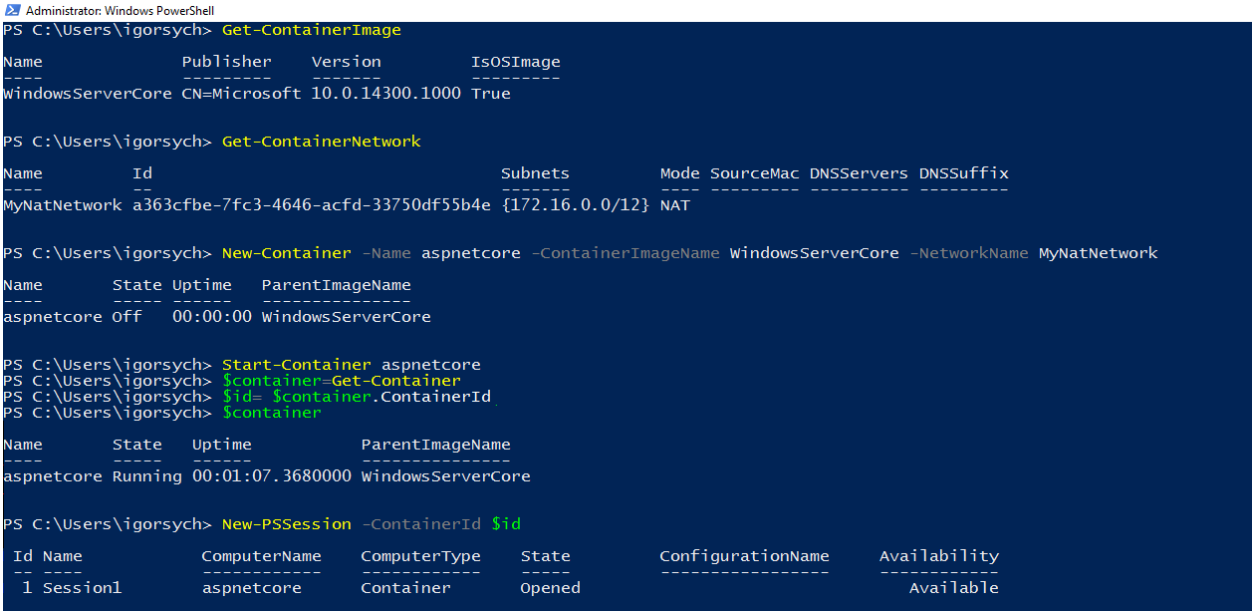


   c. Add-NetNatStaticMapping -NatName "Hb38e9905-d1c3-4ed0-8e67-3a9a35687ef3" -Protocol TCP -ExternalIPAddress 0.0.0.0 -InternalIPAddress 172.16.0.1 -InternalPort 50800 -ExternalPort 5004
6. Instead of NatName, use name from Get-NetNat output

Exercise 3 Create Container

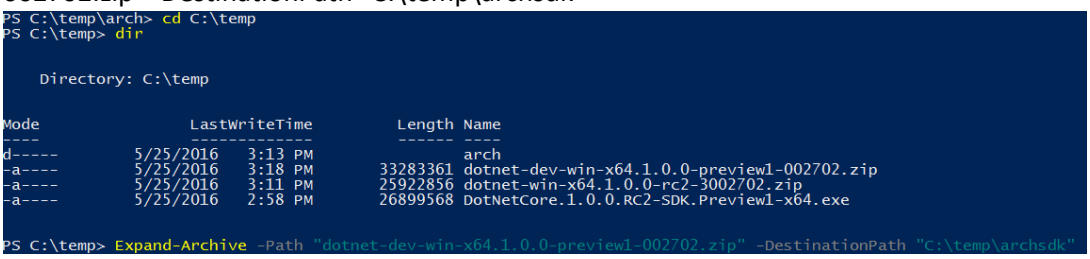
1. Create new Container using WindowsServerCore image
   New-Container -Name aspnetcore -ContainerImageName WindowsServerCore -Network MyNatNetwork
   Start-Container aspnetcore
   $container= (Get-Container)[0]

$id= $container.ContainerId
New-PSSession -ContainerId $id



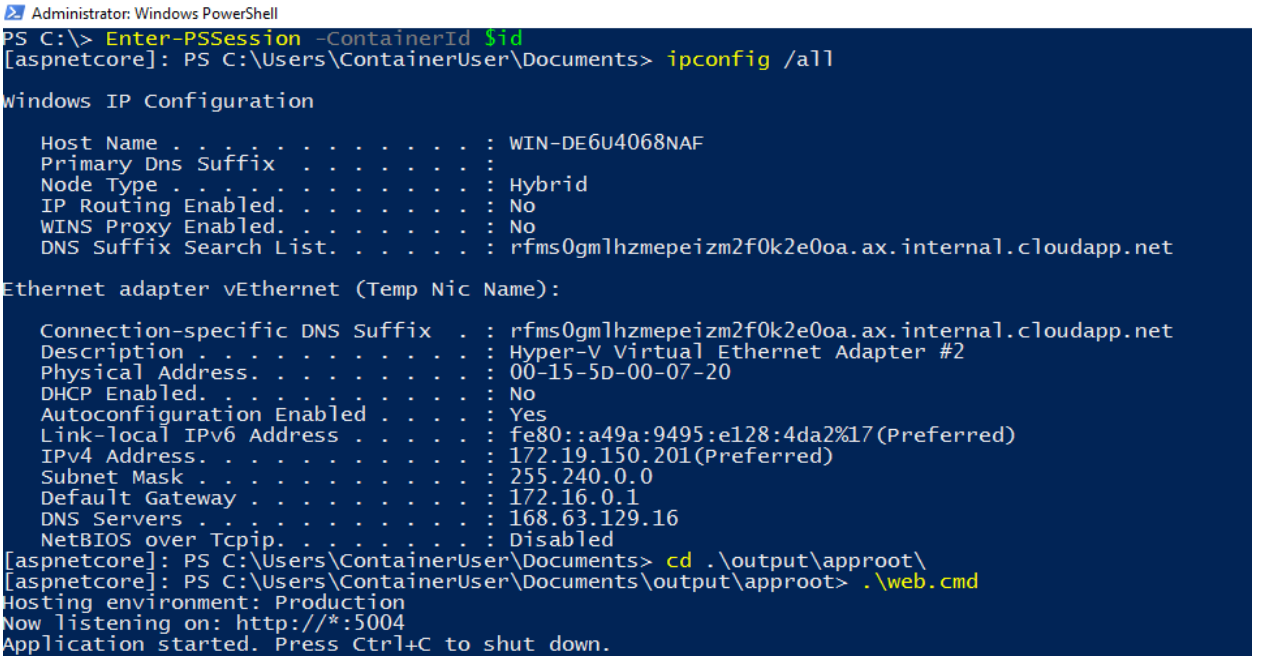
2. Connect to container and install .net core 2

    a. Open PS Session to container using “Enter-PSSession -containerId $Id”
    b. In the session download .net sdk binaries
        i. $webclient = New-Object System.Net.WebClient
        ii. $webclient.DownloadFile("http://download.microsoft.com/download/2/1/0/2107669A-0DF9-4A91-A275-74735D433045/dotnet-dev-win-x64.1.0.0-preview1-002702.zip","C:\temp\dotnet-dev-win-x64.1.0.0-preview1-002702.zip")
    c. Extract archive to a folder “Expand-Archive -Path "dotnet-dev-win-x64.1.0.0-preview1-002702.zip" -DestinationPath "C:\temp\archsdk"”

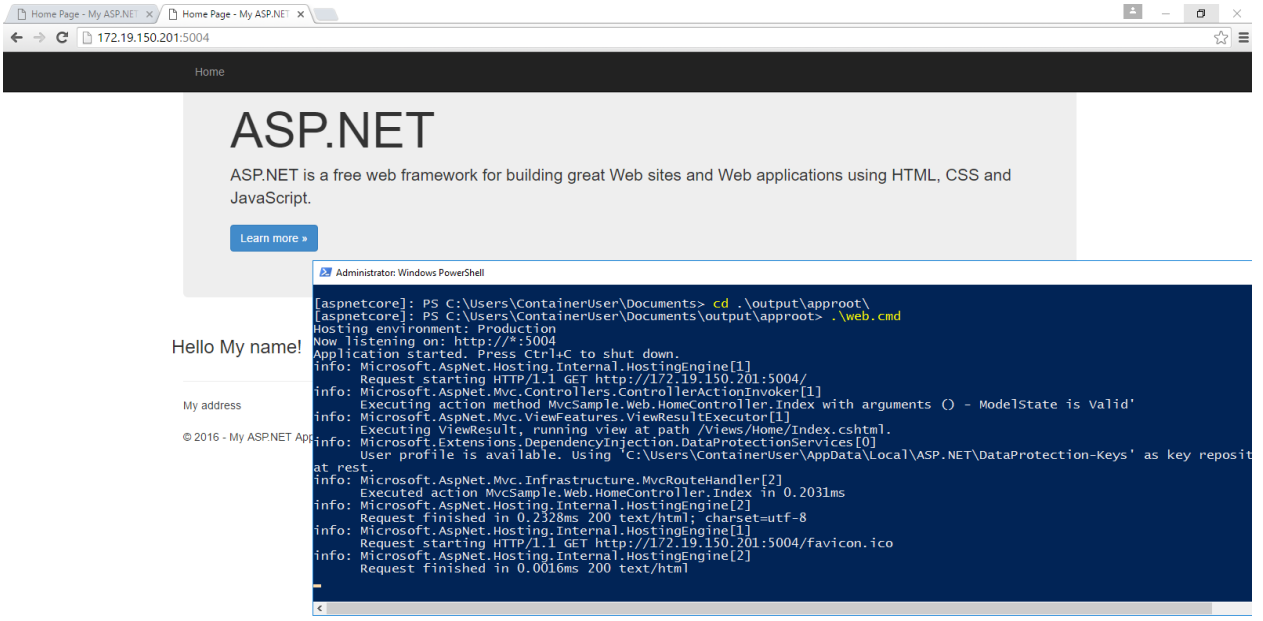


3. Exit PSSession
4. Copy published version of web app from localhost to container. Command Example:
5. Copy-Item -ToSession $session -Path "C:\Users\igorsych\Conferences\MSDevCon2016\src\src\MyShuttle" -Destination "C:\Users\ContainerUser\Documents\archsdk\WebSite" -Recurse -Exclude "*_._*"

6. Enter in PSSession again and restore dependencies “dotnet restore” and run web app using “dotnet run” command. By default, application will be available on port 5000.



7. Check that site is working in a browser. Get an IP from ipconfig /all output IPv4 address



References

- https://msdn.microsoft.com/en-us/virtualization/windowscontainers/management/container_networking
- https://channel9.msdn.com/Blogs/containers/Quick-Start-Deploying-and-Managing-Windows-Server-Containers-with-PowerShell
- https://technet.microsoft.com/en-us/library/dn283352(v=wps.630).aspx
- https://msdn.microsoft.com/en-us/virtualization/windowscontainers/management/manage_images