

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)

```
Administrator: Windows PowerShell
PS C:\Users\igorsych> git clone http://github.com/aspnet/home
Cloning into 'home'...
remote: Counting objects: 1903, done.
remote: Total 1903 (delta 0), reused 0 (delta 0), pack-reused 1903
Receiving objects: 97% (1846/1903), 612.01 KiB | 587.00 KiB/s
Receiving objects: 100% (1903/1903), 987.33 KiB | 587.00 KiB/s, done.
Resolving deltas: 100% (1151/1151), done.
Checking connectivity... done.
PS C:\Users\igorsych> cd .\home\samples\1.0.0-rc1-update1\HelloMvc\
```

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)

```
Administrator: Windows PowerShell
PS C:\Users\igorsych> git clone http://github.com/SychevIgor/Conferences
Cloning into 'Conferences'...
remote: Counting objects: 737, done.
remote: Compressing objects: 100% (9/9), done.
Receiving objects: 100% (737/737), remote: Total 737 (delta 3), reused 2 (delta 2), pack-reused 726
69.93 MiB | 17.61 MiB/s, done.
Resolving deltas: 100% (152/152), done.
Checking connectivity... done.
Checking out files: 100% (1328/1328), done.
PS C:\Users\igorsych> git checkout 56e4f6a
fatal: Not a git repository (or any of the parent directories): .git
PS C:\Users\igorsych> cd .\Conferences\
PS C:\Users\igorsych\Conferences> git checkout 56e4f6a
Checking out files: 100% (148/148), done.
Note: checking out '56e4f6a'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at 56e4f6a... devcon2106 code sample fix1
PS C:\Users\igorsych\Conferences> cd .\MSDevCon2016\src\
```

- 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

```
Administrator: Windows PowerShell

Compilation succeeded.
    0 Warning(s)
    0 Error(s)

Time elapsed 00:00:04.9949945
Project MyShuttle.Data (.NETStandard,Version=v1.5) will be compiled because de
Compiling MyShuttle.Data for .NETStandard,Version=v1.5

Compilation succeeded.
    0 Warning(s)
    0 Error(s)

Time elapsed 00:00:04.2944611
Project MyShuttle.API (.NETStandard,Version=v1.5) will be compiled because dep
Compiling MyShuttle.API for .NETStandard,Version=v1.5

Compilation succeeded.
    0 Warning(s)
    0 Error(s)

Time elapsed 00:00:04.4269221
Project MyShuttle (.NETCoreApp,Version=v1.0) will be compiled because dependen
Compiling MyShuttle for .NETCoreApp,Version=v1.0
C:\Users\igorsych\Conferences\MSDevCon2016\src\src\MyShuttle\Controllers\Carri
e variable 'ex' is declared but never used
C:\Users\igorsych\Conferences\MSDevCon2016\src\src\MyShuttle\Controllers\Carri
is async method lacks 'await' operators and will run synchronously. Consider u
cking API calls, or 'await Task.Run(...)' to do CPU-bound work on a background
C:\Users\igorsych\Conferences\MSDevCon2016\src\src\MyShuttle\Controllers\Carri
is async method lacks 'await' operators and will run synchronously. Consider u
cking API calls, or 'await Task.Run(...)' to do CPU-bound work on a background
C:\Users\igorsych\Conferences\MSDevCon2016\src\src\MyShuttle\Controllers\Carri
e variable 'ex' is declared but never used
C:\Users\igorsych\Conferences\MSDevCon2016\src\src\MyShuttle\Controllers\Carri
is async method lacks 'await' operators and will run synchronously. Consider u
cking API calls, or 'await Task.Run(...)' to do CPU-bound work on a background

Compilation succeeded.
    5 Warning(s)
    0 Error(s)

Time elapsed 00:00:03.6269424

Hosting environment: Production
Content root path: C:\Users\igorsych\Conferences\MSDevCon2016\src\src\MyShuttl
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.
```

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



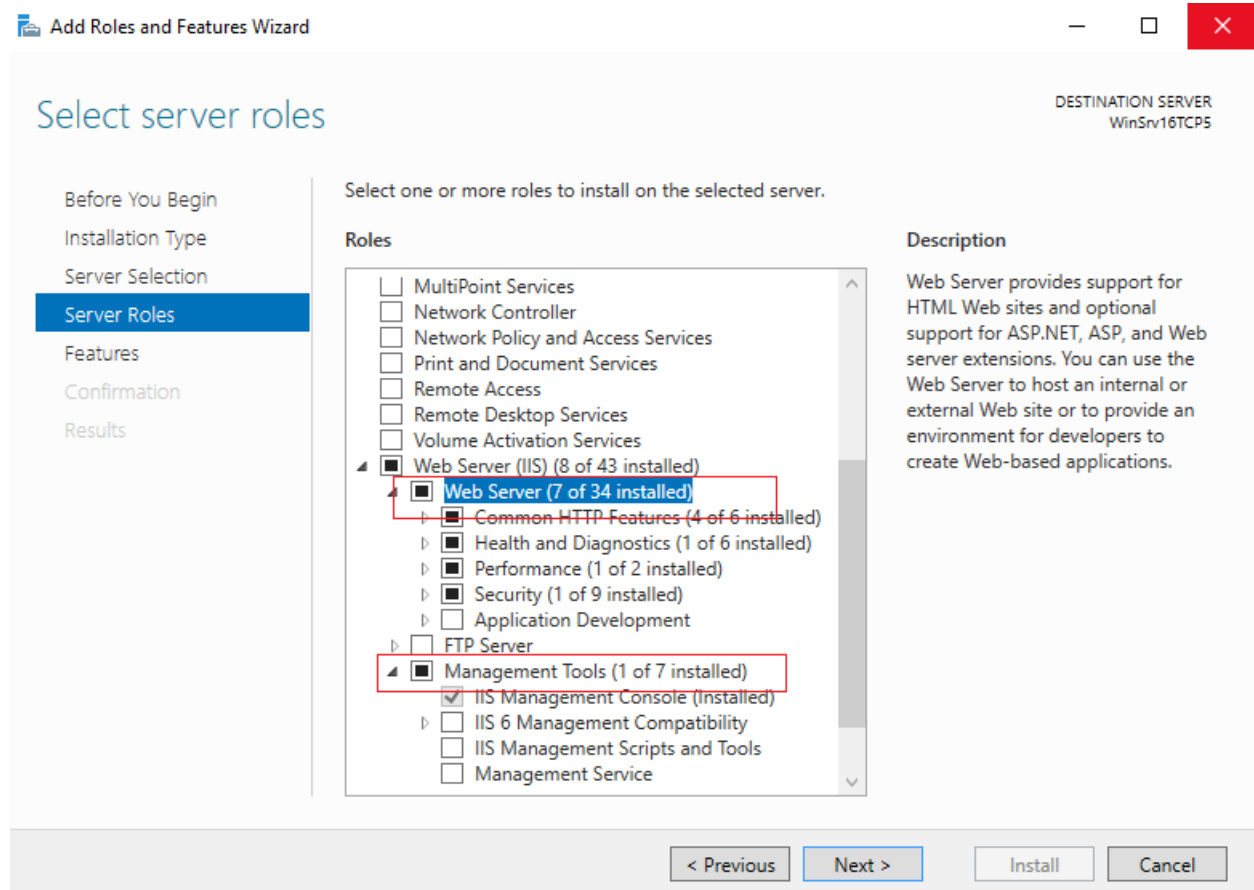
12. Press Ctrl-C to shut down the web server

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

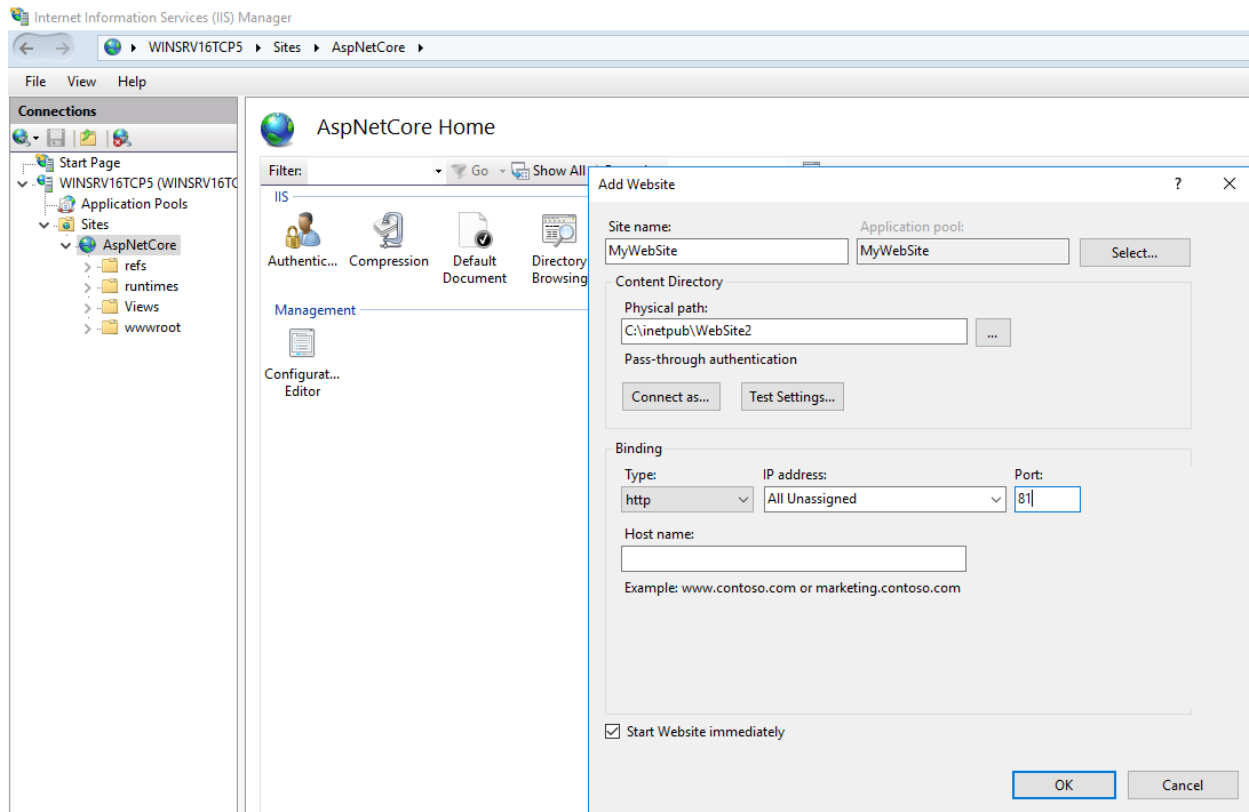
```
Select Administrator: Windows PowerShell
PS C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc> dnu publish
Microsoft .NET Development Utility Clr-x86-1.0.0-rc1-16609

Copying to output path C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output
Using Package dependency Microsoft.AspNet.Antiforgery 1.0.0-rc1-final
Using Package dependency System.Threading.Overlapped 4.0.0
Using Package dependency System.Runtime.InteropServices 4.0.0
Source C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\System.Threading.Overlapped\4.0.0
Target C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\System.Threading.Overlapped\4.0.0
Source C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\Microsoft.AspNet.Antiforgery\1.0.0-rc1-final
Target C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\Microsoft.AspNet.Antiforgery\1.0.0-rc1-final
Using Package dependency Microsoft.AspNet.Authorization 1.0.0-rc1-final
Source C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\Microsoft.AspNet.Authorization\1.0.0-rc1-final
Target C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\Microsoft.AspNet.Authorization\1.0.0-rc1-final
Using Package dependency System.Text.RegularExpressions 4.0.11-beta-23516
Source C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\System.Text.RegularExpressions\4.0.11-beta-23516
Target C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\packages\System.Text.RegularExpressions\4.0.11-beta-23516
Using Project dependency HelloMvc 1.0.0 for DNx,Version=v4.5.1
Copying source code from Project dependency HelloMvc
Source C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\project.json
Target C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\aproot\src\HelloMvc
Copying contents of Project dependency HelloMvc to C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\wwwroot
Source C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\wwwroot
Target C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc\bin\output\wwwroot
Using command 'web' as the entry point for web.config.
Time elapsed 00:00:22.0271838
C:\Users\igorsych\home\samples\1.0.0-rc1-update1\HelloMvc>
```

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



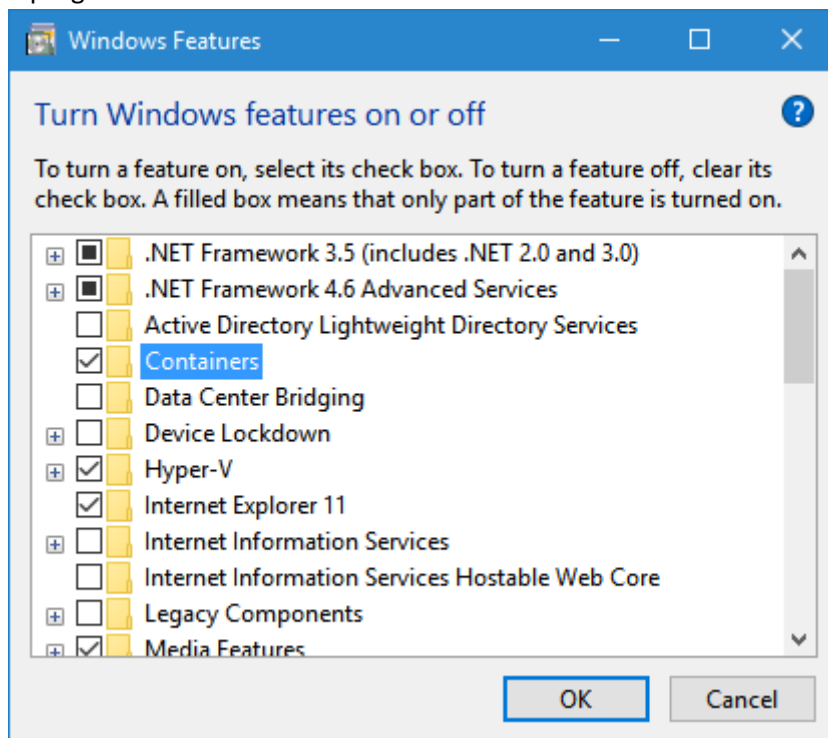
16. Start new app on IIS from this folder



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:

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\igorsych> Install-WindowsFeature containers

Success Restart Needed Exit Code      Feature Result
-----
True      Yes                SuccessRest... {Containers}
WARNING: You must restart this server to finish the installation process.
```

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
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\igorsych> Install-PackageProvider ContainerImage -Force

Name                Version      Source      Summary
----                -
ContainerImage       0.6.4.0      PSGallery   This is a PackageManagement provider module which h...

PS C:\Users\igorsych> Find-ContainerImage
PS C:\Users\igorsych> Find-ContainerImage

Name                Version      Source      Summary
----                -
NanoServer          10.0.14300.1010 ContainerImag... Container OS Image of Windows Server 2016 Technical...
WindowsServerCore   10.0.14300.1000 ContainerImag... Container OS Image of Windows Server 2016 Technical...

PS C:\Users\igorsych> Install-ContainerImage -Name WindowsServerCore
WARNING: Based on customer feedback, we are updating the Containers PowerShell module to better align with Docker. As part of that some cmdlet and parameter names may change in future releases. To learn more about these changes as well as to join in the design process or provide usage feedback please refer to http://aka.ms/windowscontainers/powershell
PS C:\Users\igorsych> Get-ContainerImage

Name                Publisher    Version      IsOSImage
----                -
WindowsServerCore   CN=Microsoft 10.0.14300.1000 True
```

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

5. Configure network nat/firewall

```
Administrator: Windows PowerShell
PS C:\Users\igorsych> New-NetFirewallRule -RemoteAddress "172.16.0.0/12" -Name "ContainerTCP5004" -DisplayName "ContainerTCP5004" -Protocol tcp -LocalPort 5004 -Action Allow -Enabled True -LocalAddress "172.16.0.0/12"

Name : ContainerTCP5004
DisplayName : ContainerTCP5004
Description :
DisplayGroup :
Group :
Enabled : True
Profile : Any
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : The rule was parsed successfully from the store. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local

PS C:\Users\igorsych> New-NetFirewallRule -Name "HostTCP50800" -DisplayName "HostTCP50800" -Protocol tcp -LocalPort 50800 -Action Allow -Enabled True

Name : HostTCP50800
DisplayName : HostTCP50800
Description :
DisplayGroup :
Group :
Enabled : True
Profile : Any
Platform : {}
Direction : Inbound
Action : Allow
EdgeTraversalPolicy : Block
LooseSourceMapping : False
LocalOnlyMapping : False
Owner :
PrimaryStatus : OK
Status : The rule was parsed successfully from the store. (65536)
EnforcementStatus : NotApplicable
PolicyStoreSource : PersistentStore
PolicyStoreSourceType : Local
```

- `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
- `New-NetFirewallRule -Name "HostTCP50800" -DisplayName "HostTCP50800" -Protocol tcp -LocalPort 50800 -Action Allow -Enabled True`

```
Select Administrator: Windows PowerShell
PS C:\Users\igorsych> New-ContainerNetwork -Name MyNatNetwork -Mode NAT -SubnetPrefix "172.16.0.0/12"

Name      Id                               Subnets      Mode SourceMac DNSServers DNSSuffix
-----
MyNatNetwork a363cfbe-7fc3-4646-acfd-33750df55b4e {172.16.0.0/12} NAT

PS C:\Users\igorsych> Get-NetNat

Name : Ha363cfbe-7fc3-4646-acfd-33750df55b4e
ExternalIPInterfaceAddressPrefix : 172.16.0.1/12
InternalIPInterfaceAddressPrefix : 172.16.0.1/12
IcmpQueryTimeout : 30
IcmpEstablishedConnectionTimeout : 1800
IcmpTransientConnectionTimeout : 120
TcpFilteringBehavior : AddressDependentFiltering
UdpFilteringBehavior : AddressDependentFiltering
UdpIdleSessionTimeout : 120
UdpInboundRefresh : False
Store : Local
Active : True

PS C:\Users\igorsych> Add-NetNatStaticMapping -NatName "Ha363cfbe-7fc3-4646-acfd-33750df55b4e" -Protocol TCP -ExternalIPAddress 0.0.0.0 -InternalIPAd
172.16.0.1 -InternalPort 50800 -ExternalPort 5004

StaticMappingID : 0
NatName : Ha363cfbe-7fc3-4646-acfd-33750df55b4e
Protocol : TCP
RemoteExternalIPAddressPrefix : 0.0.0.0/0
ExternalIPAddress : 0.0.0.0
ExternalPort : 5004
InternalIPAddress : 172.16.0.1
InternalPort : 50800
InternalRoutingDomainId : {00000000-0000-0000-0000-000000000000}
Active : True
```

- `Add-NetNatStaticMapping -NatName "Hb38e9905-d1c3-4ed0-8e67-3a9a35687ef3" -Protocol TCP -ExternalIPAddress 0.0.0.0 -InternalIPAddress 172.16.0.1 -InternalPort 50800 -ExternalPort 5004`

- Instead of NatName, use name from Get-NetNat output

Exercise 3 Create Container

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

```
$sid= $container.ContainerId
New-PSSession -ContainerId $sid
```

```
Administrator: Windows PowerShell
PS C:\Users\igorsych> Get-ContainerImage

Name          Publisher      Version      IsOSImage
-----
WindowsServerCore CN=Microsoft 10.0.14300.1000 True

PS C:\Users\igorsych> Get-ContainerNetwork

Name          Id          Subnets      Mode SourceMac DNSServers DNSSuffix
-----
MyNatNetwork a363cfbe-7fc3-4646-acfd-33750df55b4e {172.16.0.0/12} NAT

PS C:\Users\igorsych> New-Container -Name aspnetcore -ContainerImageName WindowsServerCore -NetworkName MyNatNetwork

Name          State Uptime      ParentImageName
-----
aspnetcore Off 00:00:00 WindowsServerCore

PS C:\Users\igorsych> Start-Container aspnetcore
PS C:\Users\igorsych> $container=Get-Container
PS C:\Users\igorsych> $sid= $container.ContainerId
PS C:\Users\igorsych> $container

Name          State Uptime      ParentImageName
-----
aspnetcore Running 00:01:07.3680000 WindowsServerCore

PS C:\Users\igorsych> New-PSSession -ContainerId $sid

Id Name          ComputerName ComputerType State ConfigurationName Availability
---
1 Session1      aspnetcore   Container    Opened          Available
```

2. Connect to container and install .net core 2

- a. Open PS Session to container using "Enter-PSSession -containerId \$sid"
- 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""

```
PS C:\temp\arch> cd C:\temp
PS C:\temp> dir

Directory: C:\temp

Mode                LastWriteTime         Length Name
----
d-----          5/25/2016   3:13 PM                arch
-a-----          5/25/2016   3:18 PM      33283361 dotnet-dev-win-x64.1.0.0-preview1-002702.zip
-a-----          5/25/2016   3:11 PM      25922856 dotnet-win-x64.1.0.0-rc2-3002702.zip
-a-----          5/25/2016   2:58 PM     26899568 DotNetCore.1.0.0.RC2-SDK.Preview1-x64.exe

PS C:\temp> 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.

```
Administrator: Windows PowerShell
PS C:\> Enter-PSSession -ContainerId $id
[aspnetcore]: PS C:\Users\ContainerUser\Documents> ipconfig /all

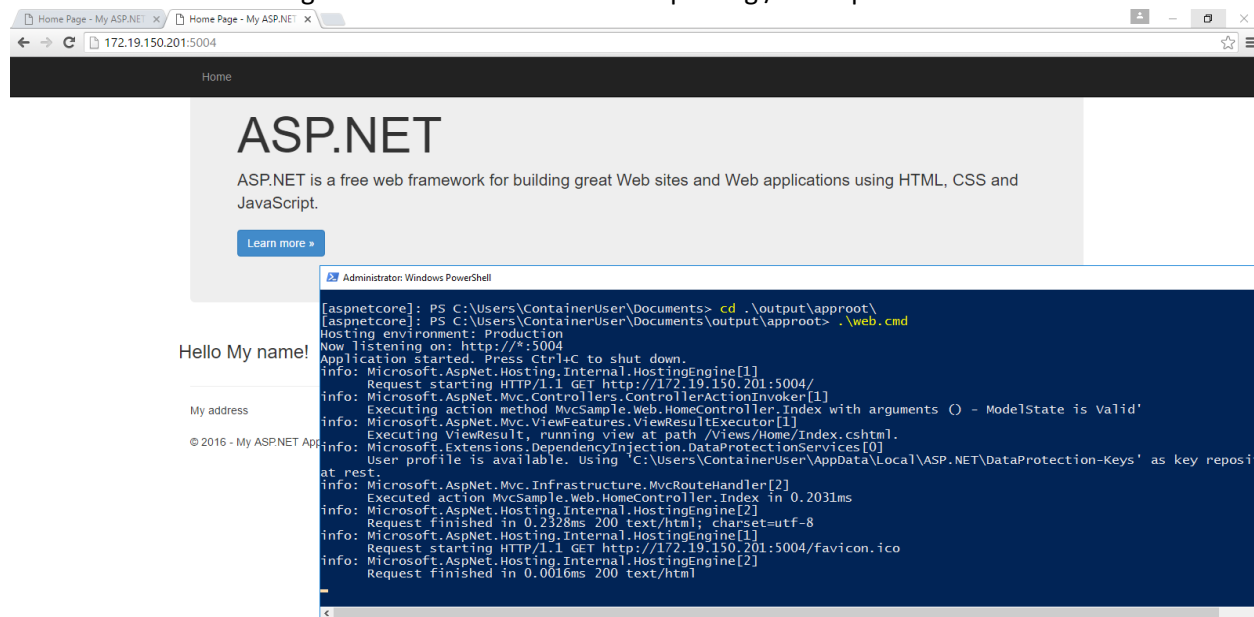
Windows IP Configuration

Host Name . . . . . : WIN-DE6U4068NAF
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : rfms0gm1hzmepeizm2f0k2e0oa.ax.internal.cloudapp.net

Ethernet adapter vEthernet (Temp Nic Name):

Connection-specific DNS Suffix . : rfms0gm1hzmepeizm2f0k2e0oa.ax.internal.cloudapp.net
Description . . . . . : Hyper-V Virtual Ethernet Adapter #2
Physical Address. . . . . : 00-15-5D-00-07-20
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::a49a:9495:e128:4da2%17(Preferred)
IPv4 Address. . . . . : 172.19.150.201(Preferred)
Subnet Mask . . . . . : 255.240.0.0
Default Gateway . . . . . : 172.16.0.1
DNS Servers . . . . . : 168.63.129.16
NetBIOS over Tcpip. . . . . : Disabled
[aspnetcore]: PS C:\Users\ContainerUser\Documents> cd .\output\aproot\
[aspnetcore]: PS C:\Users\ContainerUser\Documents\output\aproot> .\web.cmd
Hosting environment: Production
Now listening on: http://*:5004
Application started. Press Ctrl+C to shut down.
```

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://technet.microsoft.com/en-us/library/dn283352(v=wps.630).aspx)
- https://msdn.microsoft.com/en-us/virtualization/windowscontainers/management/manage_images