Dennis Tikhomirov. DevOps Spring’19.

Task 3. Hyper-V

1. Using the GUI, create the following VM without a network connection:

• Windows 7 (comp1)

• Windows Server 2016 (server)

• Ubuntu 18.04 (comp2)

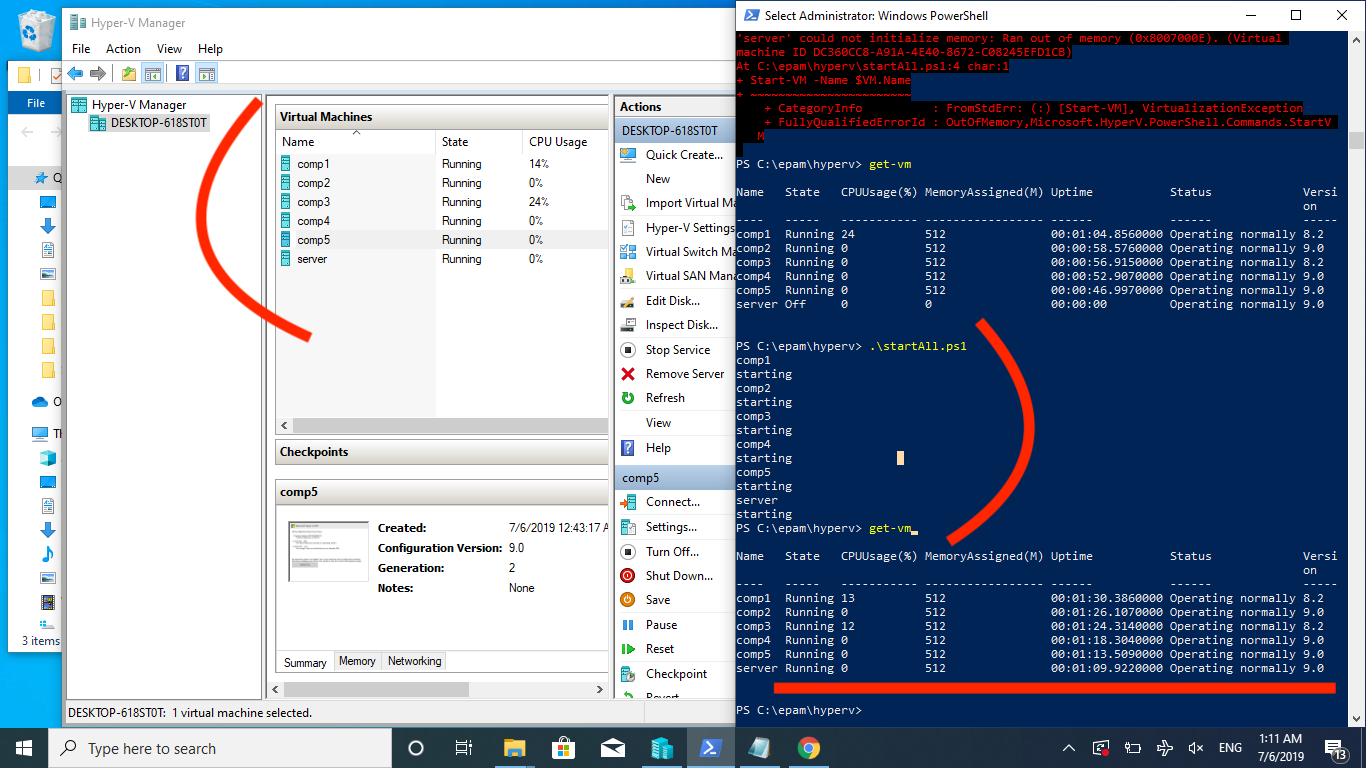
1. Using the PowerShell, create the following VM without a network connection:

• Windows 7 (comp3)

• Windows 10 (comp4)

• Ubuntu 19.04 (comp5)

Screenshot #1. Hyper-V manager, deployed VMs



I have created bunch of scripts for managing VMs on Hyper-V host.

#1. List of VMs <https://github.com/dennis00010011b/epam-devops-training/blob/master/Task3HyperV/VMs.xml>

<?xml version="1.0"?>  
<VMs>  
 <VM>  
 <Name>comp1</Name>  
 <SourcePath>C:\ISOs\IE11.Win7.HyperV\Virtual Machines\106A06B0-0DE9-4997-A87C-3760FFBEC837.vmcx</SourcePath>  
 <IsImport>true</IsImport>  
 <Memory>536870912</Memory>  
 <VHDPath>./</VHDPath>  
 <OS>Wni7</OS>  
 </VM>  
  
 <VM>  
 <Name>comp2</Name>  
 <SourcePath>C:\ISOs\ubuntu-18.04.2-desktop-amd64.iso</SourcePath>  
 <Memory>536870912</Memory>  
 <VHDPath>./</VHDPath>  
 <OS>Ubuntu18.04</OS>  
 </VM>  
  
 <VM>  
 <Name>comp3</Name>  
 <SourcePath>C:\ISOs\IE11.Win7.HyperV\Virtual Machines\106A06B0-0DE9-4997-A87C-3760FFBEC837.vmcx</SourcePath>  
 <IsImport>true</IsImport>  
 <Memory>536870912</Memory>  
 <VHDPath>./</VHDPath>  
 <OS>Wni7</OS>  
 </VM>  
  
 <VM>  
 <Name>comp4</Name>  
 <SourcePath>C:\ISOs\Win10\_1809Oct\_v2\_English\_x64.iso</SourcePath>  
 <Memory>536870912</Memory>  
 <VHDPath>./</VHDPath>  
 <OS>Wni10</OS>  
 </VM>  
  
 <VM>  
 <Name>comp5</Name>  
 <SourcePath>C:\ISOs\ubuntu-19.04-desktop-amd64.iso</SourcePath>  
 <Memory>536870912</Memory>  
 <VHDPath>./</VHDPath>  
 <OS>Ubuntu19.04</OS>  
 </VM>  
  
 <VM>  
 <Name>server</Name>  
 <SourcePath>C:\ISOs\Windows\_Server\_2016\_Datacenter\_EVAL\_en-us\_14393\_refresh.ISO</SourcePath>  
 <Memory>1036870912</Memory>  
 <VHDPath>./</VHDPath>  
 <OS>Ubuntu19.04</OS>  
 </VM>  
</VMs>

#2. PS script for creating VMs <https://github.com/dennis00010011b/epam-devops-training/blob/master/Task3HyperV/createVM.ps1>

$VMName = $args[0]  
$ISO = $args[1]  
$Memory = $args[2]  
$VHDPath = $args[3]  
 $VM = @{  
 Name = $VMName  
 MemoryStartupBytes = $Memory  
 Generation = 2  
 NewVHDPath = "$VHDPath$VMName.vhdx"  
 NewVHDSizeBytes = 53687091200  
   
 }  
 echo Creating $VMName $ISO $Memory $VHDPath  
 New-VM @VM  
  
 Add-VMDvdDrive -VMName $VMName -Path $ISO  
  
 $firmw = Get-VMFirmvare $VMName  
 Set-VMFirmvare -VMName $VMName -BootOrder $firmw.BootOrder[2]

#3. PS script for deploying VMs <https://github.com/dennis00010011b/epam-devops-training/blob/master/Task3HyperV/deploy.ps1>

[xml]$XmlDoc = Get-Content VMs.xml  
  
foreach ($VM in $XmlDoc.VMs.VM) {  
 if ($VM.isImport) {  
 Import-VM -Path $VM.SourcePath -VhdDestinationPath $VM.VHDPath -Copy -GenerateNewId  
 }  
 else{  
 .\createVM.ps1 $VM.Name $VM.SourcePath $VM.Memory $VM.VHDPath  
 }  
}

#4. PS script for removing VMs <https://github.com/dennis00010011b/epam-devops-training/blob/master/Task3HyperV/removeAll.ps1>

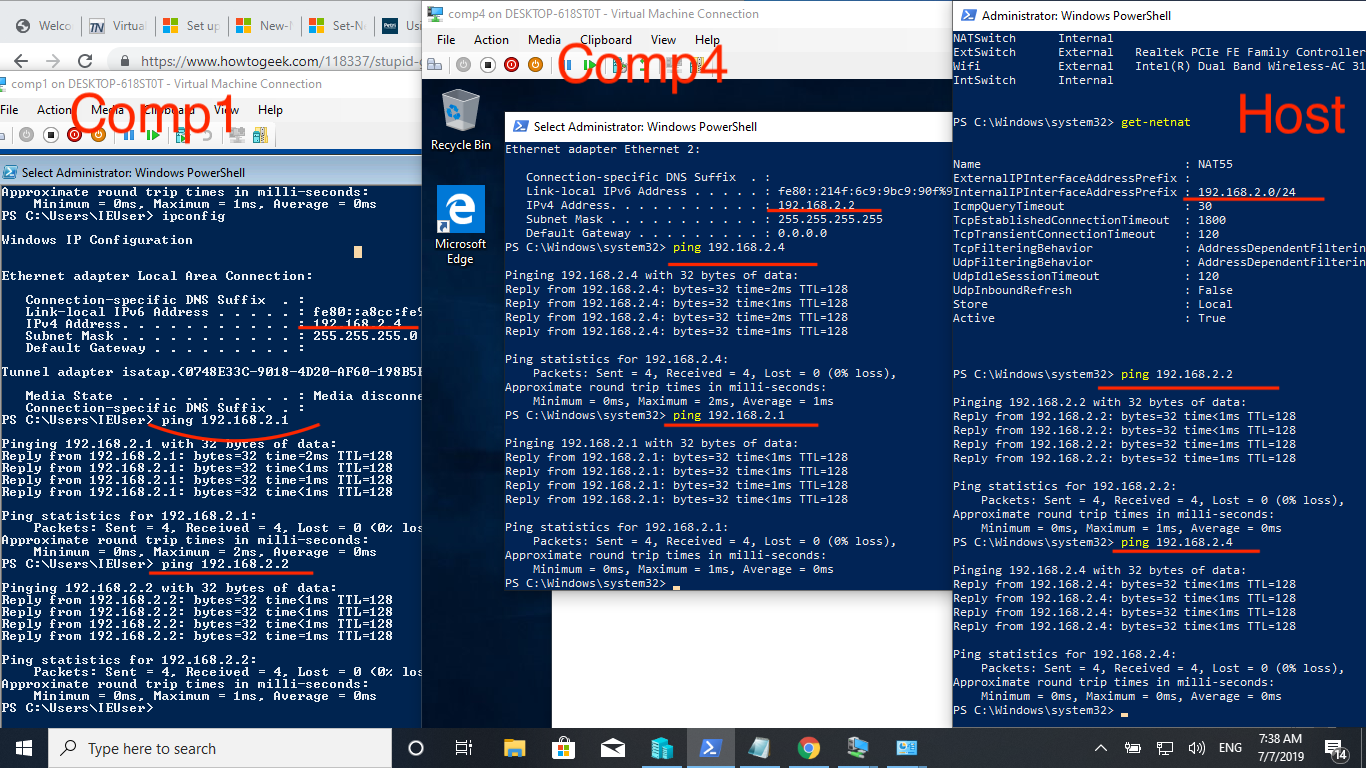
[xml]$XmlDoc = Get-Content VMs.xml  
foreach ($VM in $XmlDoc.VMs.VM) {  
 remove-vm $VM.Name -force  
 if (-Not($VM.isImport)) {  
 remove-item "$(Join-Path $VM.VHDPath $VM.Name).vhdx"  
 }  
}

1. Using the PowerShell for computers “comp1”, “comp4”, “comp5” configure NAT and Internet access

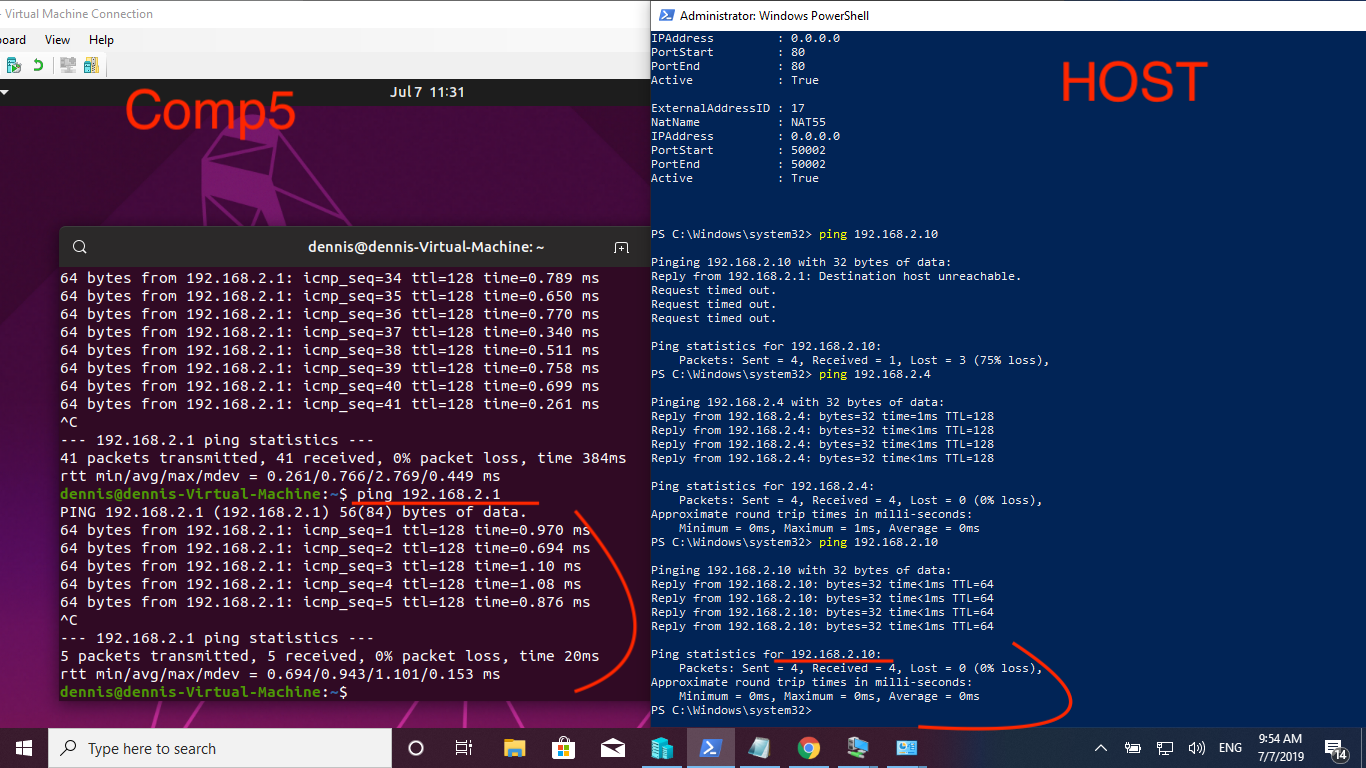
# Create internal vSwitch  
new-vmswitch -SwitchName "IntSwitch" -SwitchType Internal  
  
# Create IPGateway  
new-netipaddresstIPAddress 192.168.2.1 -PrefixLength 24 -InterfaceAlias "vEthernet (IntSwitch)"  
  
#Create NAT network  
new-netnat -name "NAT55" -InternalIPInterfaceAddressPrefix 192.168.2.0/24  
  
#Connect VMs to vSwitch  
connect-vmnetworkadapter -VMName comp1,comp4,comp5 -SwitchName IntSwitch  
  
# set manually IPAdress on each VM  
# for comp#4  
new-netipaddress 192.168.2.2 -InterfaceAlias "Ethernet"  
  
# for comp#1 (Windows7)  
$adapter = Get-WmiObject win32\_networkadapterconfiguration -filter “ipenabled = ‘true'”  
$adapter.EnableStatic(“192.168.2.4”, “255.255.255.0”)  
  
#for comp#5 (Ubuntu 19.04)  
sudo ifconfig eth0 add 192.168.2.10 netmask 255.255.255.0

1. Check your settings from the command line (terminal)

Screenshot #2. Connection between comp1,comp2,host via internal vSwitch



Screenshot #3. Connection between comp5 ,host via internal vSwitch

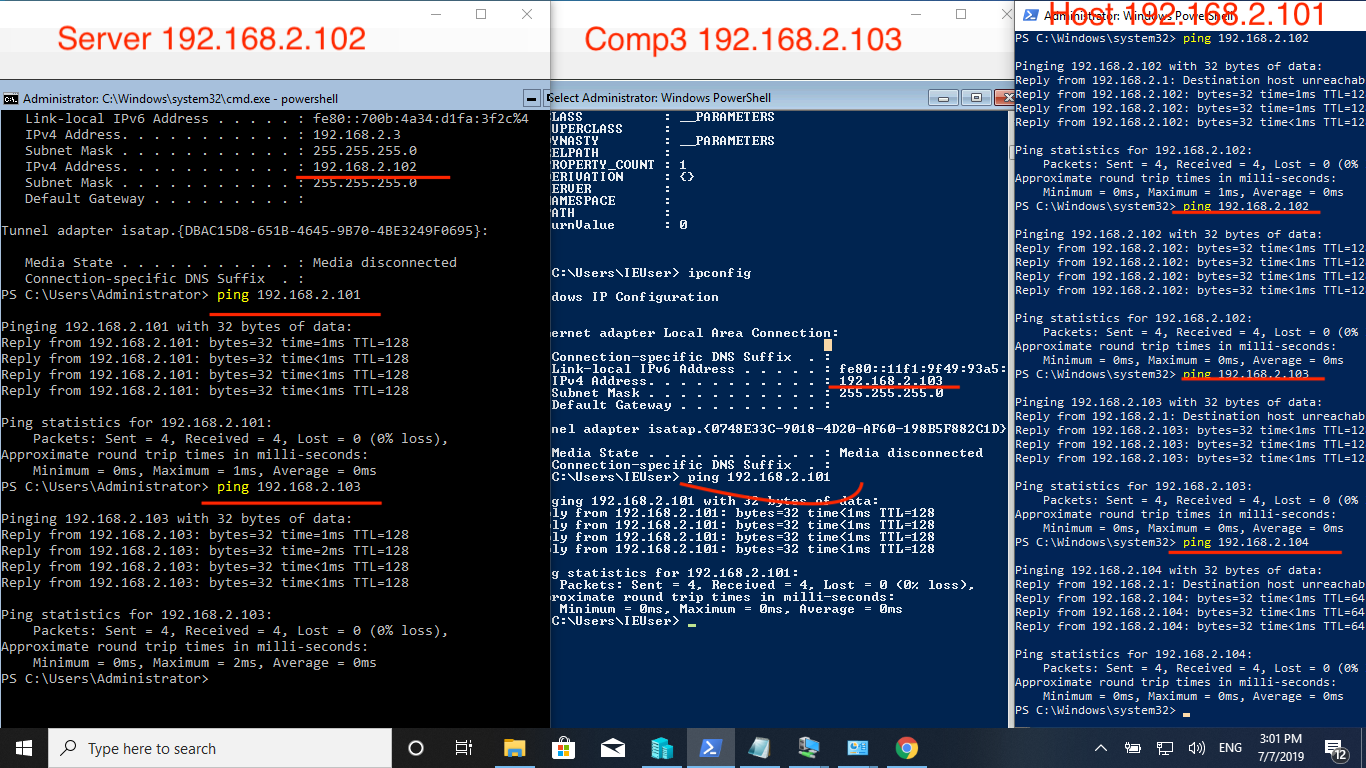


1. Using the PowerShell, add the remaining VM into a network by creating a new vSwitch

New internal vSwitch is created, 192.168.2.101

# One more vSwitch   
new-vmswitch -SwitchName "IntSwitch2" -SwitchType Internal  
new-netipaddress -IPaddress 192.168.2.101 -PrefixLength 24 -InterfaceAlias "vEthernet (IntSwitch2)"  
сonnect-vmnetworkadapter -VMName comp2,comp3,server -SwitchName IntSwitch2  
  
# set manually IPAdress on each VM  
# for 'server' (Windows Server 2016)  
new-netipaddress 192.168.2.102 -PrefixLength 24 -InterfaceAlias "Ethernet"  
  
#for comp#2 (Ubuntu 18.04)  
sudo ifconfig eth0 add 192.168.2.104 netmask 255.255.255.0  
  
# for comp#3 (Windows7)  
$adapter = Get-WmiObject win32\_networkadapterconfiguration -filter “ipenabled = ‘true'”  
$adapter.EnableStatic(“192.168.2.103”, “255.255.255.0”)

Screenshot #4. Connection between comp3,server, host via internal vSwitch



6. Configure DHCP on “server”

7. Configure the LAN and Internet access

8. Check your settings from the command line (terminal)

9. Configure remote desktop connection to each VM

10. On the VM "server" install Hyper-V, which install 2 VM with Windows 7 ("comp6" and "comp7”)

11. Demonstrate file transfer and editing from Host to Guest and back

12. Create a report with screenshots and attach script files that demonstrate the solution of tasks