

ITAS 233 LAB 06

Chapter 6

Submitted by

Jithin Jose

Student Id:655775815

Submitted to

Brandon Britt

Submitted on

10/09/2019

Table of Contents

INTRODUCTION.....	3
PART 2: GUI Storage Spaces.....	3
Activity 6-2	3
Activity 6-3	4
Activity 6-4	5
Activity 6-5	5
Activity 6-6	6
CONCLUSION.....	11

Introduction

The lab is basically about setting up **ServerVM1, InstallCore and ServerVM2** on the Hyper-v. We need to install server 2016 Desktop version on both server vm1 and serverVM2. However, we have to install the server core version on the install core vm. This lab also teaches us about making vm using the powershell, checkpoints in the hyper-v and accessing a hyper drive in the vm. This lab also teach us about making a PS session towards the serverVM1 and ServerVM2 using the poweshell of the **hyper-V**.

Part 2: Working with hyper V

Activity 6-2:Making a VMTest1 using powershell

Make a new vm with name VMtest1

Command used:

```
PS C:\Users\Administrator> New-VM VMTest1 -MemoryStartupBytes 1GB -NewVHDPATH  
c:\VMs\VMTest1\VMTest1.vhdx -NewVHDSIZEBytes 40GB
```

Connect the network adapter into this vm:

```
Connect-VMNetworkAdapter VMTest1 -Name "Network Adapter" -SwitchName JJPrivateNet
```

Link the dvd drive into the ios:

```
Set-VMdvdDrive VMTest1 -Path C:\iso\Windows_Server_2016_Datacenter_EVAL_en-  
us_14393_refresh.ISO
```

View the details of the VMtest1

```
Get-VM VMTEst1
```

```
Get-VM VMTEst1 | fl * // To get more details
```

Command to start the vmare which are turned

Get-VM | Where-Object {\$_.State -eq "Off"} | Start-VM

Force turn of the vmware

Get-VM | Where-Object {\$_.State -eq "running"} | Stop-VM

Removing Vmtest1 and vhdx

Remove-VM VMTest1 -Force

del C:\VMs\VMTest1\VMTest1.vhdx

Activity 6-3:Successful Checkpoint Production

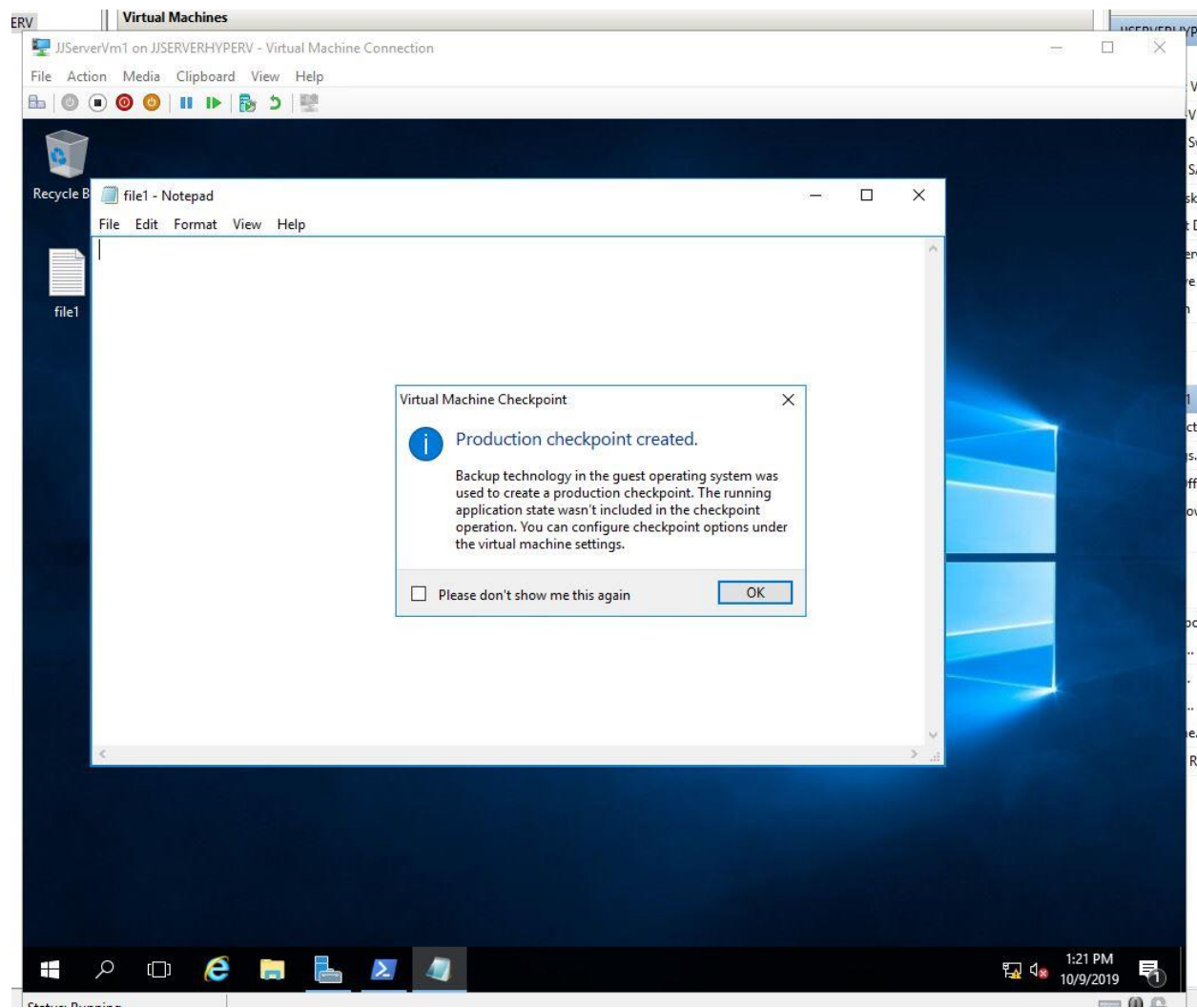


Figure 1: Screen shot of the successful production

Checkpoints are also like the snapshot in the vmware. We are able to make the snapshot in the hyper-v from the **action** toolbar in the vm screen.

Activity 6-4:Exporting InstallCore to the exports folder

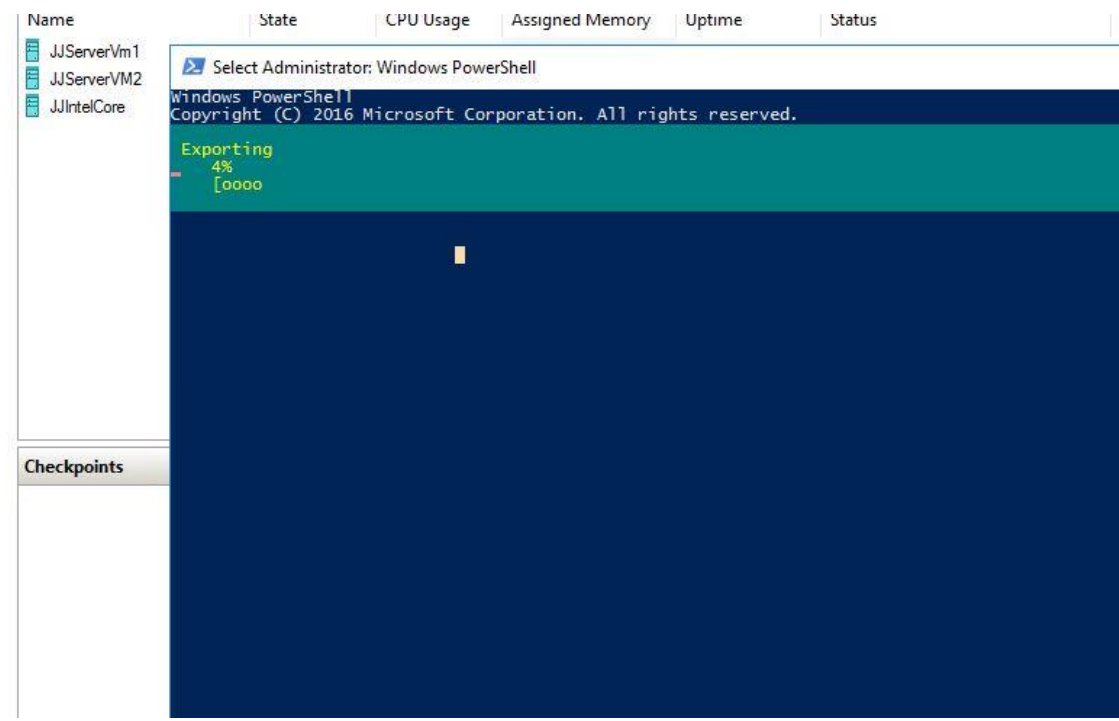


Figure 2: Exporting the install core on to the exports folder in the c drive

Exporting and importing the vm helps us to make a duplicate version of the vm. This also helps us to make a similar vm with the same features with the different name.

Activity 6-5: Drive Share to the Servervm1

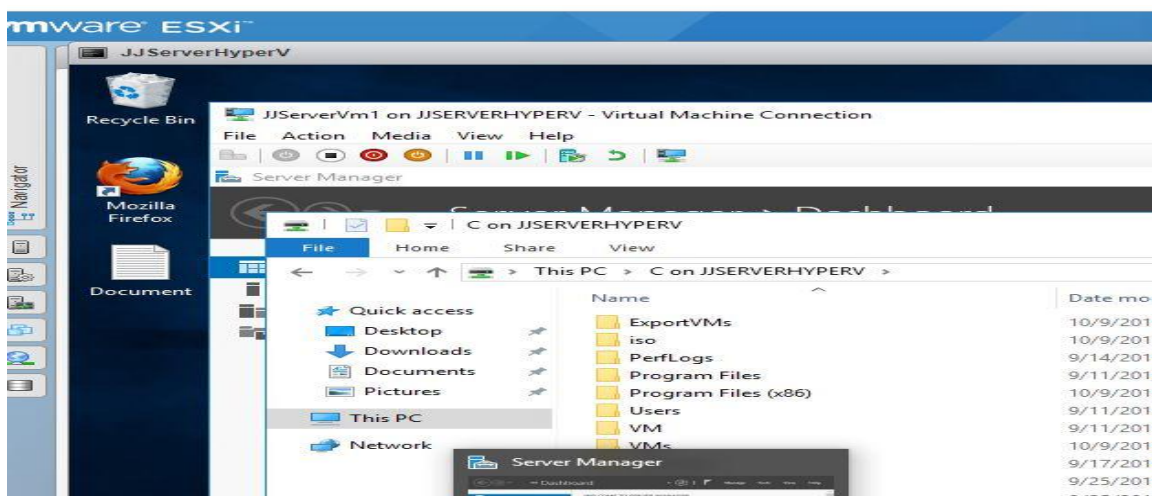


Figure 3: Accessing C drive of the hyper v in the server dm1

Enhanced mode in the **hyper-v settings** help us access the drive of the **Hyper V** . If the enhance mode is turned on, when we turn on the **servervm1** . it ask for the things that need to be shared.

Activity 6-6:PSsession to the Servervm1 from hyperV

Power shell output screen after typing all the commands in the activity 6.6.

This commands below are the commands that are required to start the vm of the hypervisor using the power shell. This commands also teach us about a start a powershell session from the powershell of the hypervisor. It also make us aware us how to shutdown the vm from the powershell.

Windows PowerShell

Copyright (C) 2016 Microsoft Corporation. All rights reserved

PS C:\Users\Administrator> Set-VMHost -EnableEnhancedSessionMode \$false

PS C:\Users\Administrator> Start-VM JJServerVm1

PS C:\Users\Administrator> Get-VM JJServerVM1

Name	State	CPUUsage(%)	MemoryAssigned(M)	Uptime	Status	Version
----	----	-----	-----	-----	-----	
JJServerVm1	Running	1	1024	00:00:33.0930000	Operating normally	8

PS C:\Users\Administrator> Enter-PSSession -VMName JJServerVM1

cmdlet Enter-PSSession at command pipeline position 1

Supply values for the following parameters:

Credential

Enter-PSSession : The credential is invalid.

At line:1 char:1

+ Enter-PSSession -VMName JJServerVM1

+ ~~~~~

+ CategoryInfo : InvalidArgument: (:) [Enter-PSSession], PSDirectException

+ **FullyQualifiedErrorId** :
CreateRemoteRunspaceForVMFailed,Microsoft.PowerShell.Commands.EnterPSSessionC
ommand

PS C:\Users\Administrator> Enter-PSSession -VMName JJServerVM1

cmdlet Enter-PSSession at command pipeline position 1

Supply values for the following parameters:

Credential

[JJServerVm1]: PS C:\Users\Administrator\Documents>

[JJServerVm1]: PS C:\Users\Administrator\Documents> Get-Disk

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus
Total Size	Partiti			

Style

-----	-----	-----	-----
0	Virtual HD	Healthy	Online 127 GB MBR

[JJServerVm1]: PS C:\Users\Administrator\Documents> Get-NetIPAddress

IPAddress : fe80::643d:c1df:b2cd:9630%13

InterfaceIndex : 13

InterfaceAlias : Ethernet

AddressFamily : IPv6

Type : Unicast

PrefixLength : 64

PrefixOrigin : WellKnown

SuffixOrigin : Link

AddressState : Preferred

ValidLifetime : Infinite ([TimeSpan]::MaxValue)

PreferredLifetime : Infinite ([TimeSpan]::MaxValue)

SkipAsSource : False

PolicyStore : ActiveStore

IPAddress : fe80::5efe:192.168.0.12%8

InterfaceIndex : 8
InterfaceAlias : isatap.{F39630F3-E388-46BF-8E13-CA989FB7A180}
AddressFamily : IPv6
Type : Unicast
PrefixLength : 128
PrefixOrigin : WellKnown
SuffixOrigin : Link
AddressState : Deprecated
ValidLifetime : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource : False
PolicyStore : ActiveStore
IPAddress : ::1
InterfaceIndex : 1
InterfaceAlias : Loopback Pseudo-Interface 1
AddressFamily : IPv6
Type : Unicast
PrefixLength : 128
PrefixOrigin : WellKnown
SuffixOrigin : WellKnown
AddressState : Preferred
ValidLifetime : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource : False
PolicyStore : ActiveStore
IPAddress : 192.168.0.12
InterfaceIndex : 13
InterfaceAlias : Ethernet

AddressFamily : IPv4
Type : Unicast
PrefixLength : 24
PrefixOrigin : Manual
SuffixOrigin : Manual
AddressState : Preferred
ValidLifetime : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource : False
PolicyStore : ActiveStore
IPAddress : 127.0.0.1
InterfaceIndex : 1

InterfaceAlias : Loopback Pseudo-Interface 1

AddressFamily : IPv4
Type : Unicast
PrefixLength : 8
PrefixOrigin : WellKnown
SuffixOrigin : WellKnown
AddressState : Preferred
ValidLifetime : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource : False
PolicyStore : ActiveStore

[JJServerVm1]: PS C:\Users\Administrator\Documents> Stop-Computer

Stop-Computer : Privilege not held.

+ **CategoryInfo** : InvalidOperation: (WIN-HAKP20CJRE4:String) [Stop-Computer],
ManagementException

+ **FullyQualifiedErrorId** :
 StopComputerException,Microsoft.PowerShell.Commands.StopComputerCommand

[JJServerVm1]: PS C:\Users\Administrator\Documents> shutdown /s /t 0

[JJServerVm1]: PS C:\Users\Administrator\Documents>

```

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

PS C:\Users\Administrator> Set-VMHost -EnableEnhancedSessionMode $false
PS C:\Users\Administrator> Start-VM JJServerVm1
PS C:\Users\Administrator> Get-VM JJServerVm1

Name      State      CPUUsage(%) MemoryAssigned(M) Uptime          Status           Version
----      -
JJServerVm1 Running 1          1024             00:00:33.0930000 Operating normally 8.0

PS C:\Users\Administrator> Enter-PSSession -VMName JJServerVm1
cmdlet Enter-PSSession at command pipeline position 1
Supply values for the following parameters:
Credential
Enter-PSSession : The credential is invalid.
At line:1 char:11
+ Enter-PSSession -VMName JJServerVm1
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Enter-PSSession], PSArgumentException
+ FullyQualifiedErrorId : CreateRemoteRunspaceForVMFailed,Microsoft.PowerShell.Commands.EnterPSSessionCommand

PS C:\Users\Administrator> Enter-PSSession -VMName JJServerVm1
cmdlet Enter-PSSession at command pipeline position 1
Supply values for the following parameters:
Credential
JJServerVm1]: PS C:\Users\Administrator\Documents> Get-Disk
JJServerVm1]: PS C:\Users\Administrator\Documents>

Number Friendly Name Serial Number          HealthStatus          OperationalStatus      Total Size Partition
-----
0 Virtual HD                               Healthy               Online                 127 GB MBR

JJServerVm1]: PS C:\Users\Administrator\Documents> Get-NetIPAddress

IPAddress      : fe80::643d:c1df:b2cd:9630%13
InterfaceIndex : 13
InterfaceAlias : Ethernet
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 64
PrefixOrigin   : WellKnown
SuffixOrigin    : Link
AddressState    : Preferred
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource    : False
PolicyStore     : ActiveStore

IPAddress      : fe80::5efe:192.168.0.12%8
InterfaceIndex : 8
InterfaceAlias : isatap.{F39630F3-E388-46BF-8E13-CA989FB7A180}
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 128
PrefixOrigin   : WellKnown
SuffixOrigin    : Link
AddressState    : Deprecated
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource    : False
PolicyStore     : ActiveStore

IPAddress      : ::1
InterfaceIndex : 1
InterfaceAlias : Loopback Pseudo-Interface 1
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 128
PrefixOrigin   : WellKnown
SuffixOrigin    : Preferred
AddressState    : Preferred
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)

```

Figure 4: Screenshot of the powershell activity 6.6

Conclusion

This lab helped me to know more about using vm on the hyper-v . We created serverVM1, ServerVM2 and InstallCore using this lab in the hyper-v. We also created a vm named VMTes1 using the power shell in the hyper-v. We not only learned how to make vm using the powershell, but also learned to make checkpoints, PS session using the powershell and access hyper v drive using the EnhancedSessionMode in the Hyper-v Settings.