



SETTING UP A MULTIPLE VIRTUAL MACHINE LABORATORY ENVIRONMENT FOR MICROSOFT SQL SERVER ALWAYS ON, AVAILABILITY GROUP FAILOVER CLUSTERS, USING ORACLE VIRTUALBOX

In 30 Easy Steps. Maybe more like 87 though...

Abstract

Using Oracle VirtualBox to set up several Windows Server 2016 servers with four of them hosting SQL Server 2019 instances – to demonstrate AlwaysOn Availability Groups with Windows Failover Clustering

Rick Bellows
rgbellows@yahoo.com
Validien Solutions

Contents

A little bit of background.....	2
Step 0) Verify your Host System:.....	4
Step 1) Install Oracle VirtualBox software.....	4
Step 2) Set up some folders on the Host OS:	5
Step 3) Get the files you need on the Host OS for:	6
Step 4) Create the first Windows Server 2016 Virtual Machine	8
Step 5) From VirtualBox, create the two Shared Folders	12
Step 6) From Virtual Box, set up two Network Adapters:	15
Step 7) Start the Virtual Machine.....	17
Step 8) Install Vbox Guest Additions	21
Step 9) Configure the Network Adapters in the Virtual Machine	27
Step 10) Server Manager – Change the Name of this Windows Server	28
Step 11) Make this machine a domain controller:.....	28
Step 12) Add an Active Directory domain user account.	30
Step 13) Join this User account to the Domain Admins group.	30
Step 14) Install another Windows Server for our 1 st SQL Server	31
Step 15) Add this computer to the domain.....	31
Step 16) Re-Start this server.....	32
Step 17) Install SQL Server 2019 to this machine.	32
Step 18) Install SQL Server Management Studio	32
Step 19) Add second and third and fourth SQL Server machines	33
Step 20) Install Failover Clustering on the first of the Server 2016 machines.	33
Step 21) On WinSvr16-SQL1, Create the Cluster.	34
Step 22) Configuring Quorum and File Share Witness.....	34
Step 23) Confirm settings	35
Step 24) Turn off the Firewall.....	35
Step 25) Set up a SQL Server Availability Group	36
Step 26) Setting up fo Availability Groups.....	36
Step 27) Choose the Databases to join the Availability Group.	37
Step 28) Set up the Listener	37
Step 29) Synch the databases	37
Step 30) Verify connectivity to the Listener from the Domain Controller	38

A little bit of background

Since the beginning of SQL Server, several problems have prompted an evolution of Windows and Database Engine technology. Some of these problems can be asked as questions like:

1. How do you take a running SQL Server installation down for security patching or maintenance, and keep the applications that rely on the databases up without interruption? (High Availability: HA)
2. If corruption happens on a database, or the running SQL Server installation, what can we have in place that minimizes downtime and prevents data loss? (High Availability: HA)
3. What options do we have in replicating databases across different geographical regions, in case of a large scale disaster at a primary data center? (Disaster Recovery: DR)
4. What options do we have for increasing the ability to handle more load or traffic? (High Performance: HP)

High Availability has been addressed by the use of Clusters. This is where the Database Administrators can have applications "talk" to a central Listener server, which routes the database traffic to one of several SQL Server engines. The one currently in use is the "Primary" node. Other nodes are in standby and known as Secondary replicas. When you need to patch or do high-impact maintenance on SQL-A, you "failover" the cluster so the Listener points to a secondary node: SQL-B (which then becomes the primary node.) In the past, this has been performed manually, and the Windows Server (hosting the SQL Server instances) didn't participate in the failover, except that the subnet routing had changed. It is through the network (a class D network: e.g. 10.10.10.1) that the Listener knows which machine to route the application's database traffic to. This addresses point 1.

Windows Failover Clustering – where the Windows server plays a more active role in automatically handling this failover as it senses issues with a Primary node, and gives us a greater availability of the databases due to unforeseen problems, and adds automation in that process. This addresses point 2.

Log Shipping is a well-established approach to asynchronously pushing transaction changes on a primary SQL instance to another computer – even if that backup system is very far away; across the country or on a different continent. This can usually accommodate a small amount of data loss, since it's meant for if a meteor hits the primary datacenter. There are newer technologies that make this more streamlined and cut down the possibility of data loss. This addresses point 3.

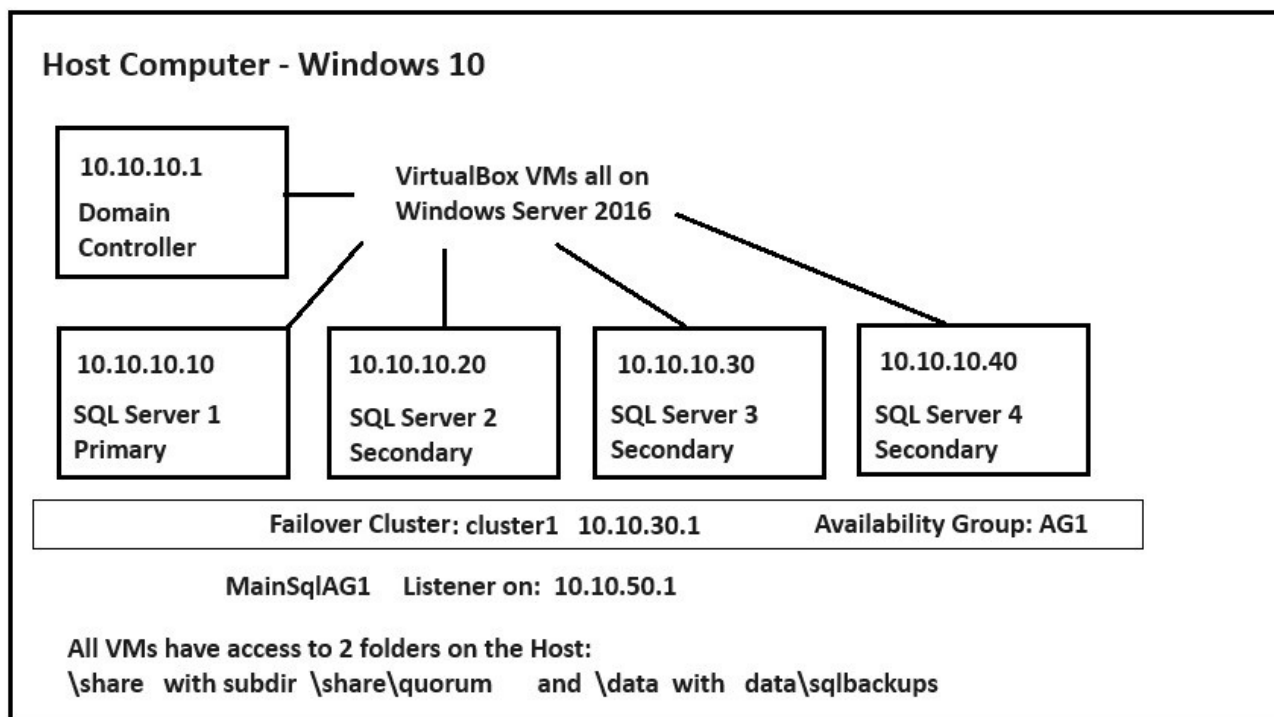
Increasing the capacity of SQL Server to accommodate ever-increasing amounts of data, is done by either Scaling Up (upgrading hardware,) or Scaling Out (adding servers to spread out the load.) Scaling Up has always been an expensive option, and while always something we could do; we're usually tasked with increasing capacity in the most economical way. Partitioning databases across more (cheaper) machines – is usually more desirable, but presents problems that have recently been addressed with Microsoft's offerings with Availability Groups. This addresses point 4.

There has been a steady evolution in these technologies, since the earliest versions of SQL Server. Even version 6.5 had clustering. Version 7 had log shipping. Increasing technology offerings in Windows Server (the Host Operating System) has had similar transformational changes that, when combining SQL Server on Windows Server – in the right configurations – has bolstered the IT footprint of many, many companies in such a way that they minimize the time required to recover (expressed as the Recovery Time Objective: RTO,) and minimize potential data loss (expressed as the Recovery Point Objective: RPO). Windows Server 2008 and SQL Server 2008 were pivotal releases. SQL Server 2012 was also a very important version, which provided technologies known as AlwaysOn, and Availability Groups. At this time, in August 2023, Windows Server 2016 is in widespread use. SQL Server 2019 is the most common version of SQL Server running on it; and it is that combination that will be the topic of this presentation.

There is a preview release of Windows Server 2022, and SQL Server 2022 – but these are not yet in large scale adoption yet. Most CTO's believe in not being on the "bleeding edge", but on the versions that are on Long Term Support.

For myself, I've been keeping up on a lot of this technology, but have only limited experience in managing these Clustering technologies. A few years ago I purchased the SQL Server 2012 Training Kit – which is for a Microsoft certification (which is no longer offered,) as I have found these to be some of the most authoritative books, since they are published by Microsoft Press.

A weeks ago I picked up the "Administering SQL Server 2012" book, and the first section starts off by saying **"You will need administrative privileges on six machines...."** I rolled my eyes because I only have 4 computers in my home office. But I got to thinking: "I've used VMWare to do this in the past – but on the machine I had, I could only get 2 virtual machines. What about Oracle's VirtualBox I was hearing about? I've got a much better development machine now. I'm sure I can do this...." Here's what I'm shooting for:



Step 0) Verify your Host System:

Your Desktop machine should have plenty of SSD space, memory, and cores.

My Host System Specs:

Windows 10 Professional

AMD Ryzen 5 1600 6 core

3200 mHz 12 logical processors (cores)

32 GB DDR4 RAM

SSD: 350 GB available C:

HDD: Two 1 TB drives >500 GB available on each as drives D: and E:

We're going to need to create at least 4 virtual machines:

Each with Windows Server 2016.

- One set up as a Domain Controller
 - Three with SQL Server 2019 Developer Edition (I'll avoid installing SSAS, and SSRS)
- 4 x 50 GB SSD = 200 GB SSD (not pre-allocated) I have plenty of space on the spinners too.
4 x 2 GB RAM = 8 GB on guest OS's. That's 1/4 of my RAM. Should be fine.

I'm sure I could get more installed, as I have 12 cores.

It might get slow when they're all running, but that's a problem I'm looking forward to seeing for myself.

Also:

Your Desktop machine must have Virtualization turned on in the BIOS and Hyper-V turned on in Windows 10.

Step 1) Install Oracle VirtualBox software.

The most current version of Oracle's VirtualBox (7.0.10 as of late August, 2023) is at:

<https://download.virtualbox.org/virtualbox/7.0.10>

You can just enter this into your browser:

<https://download.virtualbox.org/virtualbox/7.0.10/VirtualBox-7.0.10-158379-Win.exe>

and it should place this executable file in your Downloads folder:

VirtualBox-7.0.10-158379-Win.exe

So run it to install VirtualBox. It's very easy. All it needs to know is where to install it to.

You can find the User Manual for this version at:

<https://download.virtualbox.org/virtualbox/7.0.10/UserManual.pdf>

That pdf file is also included locally in the installation folder off Program Files

Step 2) Set up some folders on the Host OS:

We will need a common place to store files. We have two options.

- a) Place them on this Domain Controller
- b) Place them on the Host OS (i.e. Windows 10)

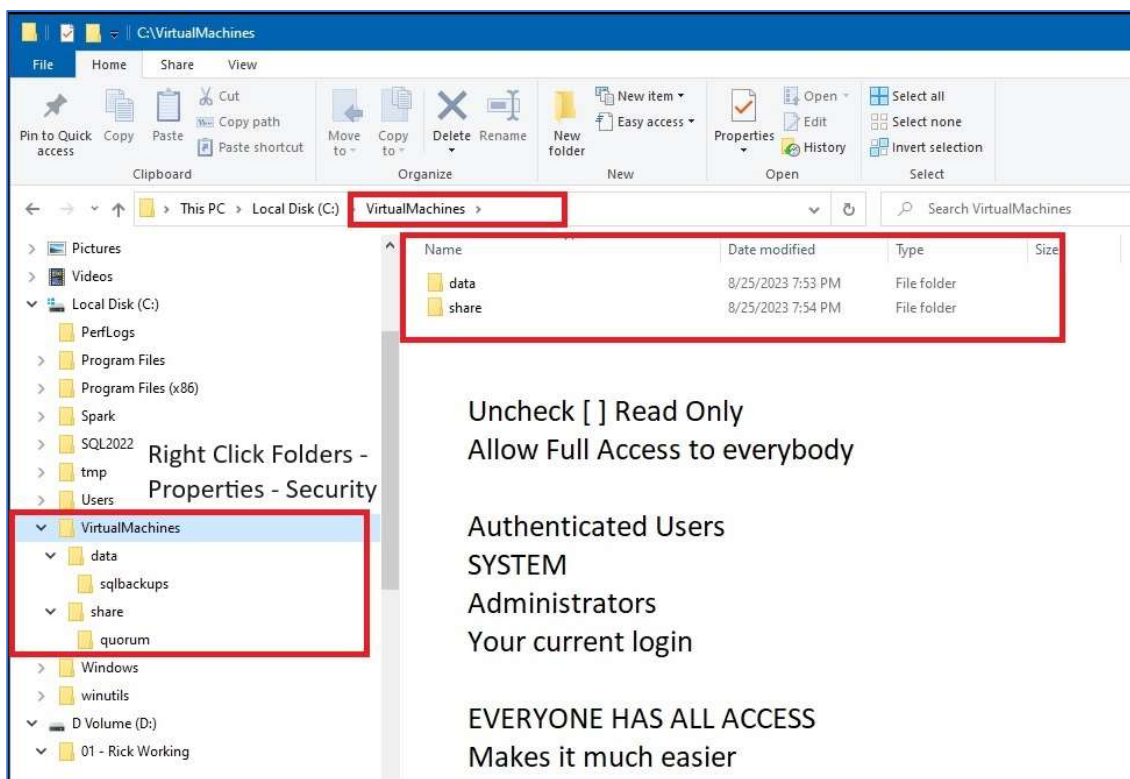
I have chosen to put these on the Host computer, in:

C:\VirtualMachines\share\Quorum

C:\VirtualMachines\data\SQLBackups

So first, let' create those folders in the Windows 10 host OS.

I had issues when creating these off my C: drive. I mean, I could create the folders, but then when I copied some files to the \share directory, I couldn't see the files. When I tried copying them again, it asked did I want to replace them. So they were obviously there, and I'm pretty sure it's some kind of permissions issue. You may need to open up the permissions on these folders. I chose to give all the "users" Full Control.



Step 3) Get the files you need on the Host OS for:

Windows Server 2016 ISO image
SQL Server 2019 Installer
SQL Server Management Studio

Windows Server 2016 ISO image:

Windows_Server_2016_Datacenter_Eval_en-us_14393_refresh.ISO

You will have to get this from Microsoft, in their Evaluation area. Navigate through this link:

<https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2016>

<https://info.microsoft.com/ww-landing-windows-server-2016.html>

You may need to supply your name, email and phone number - but when you do, it will let you download the 180 day trial:

This is: 6,808,810 kb (6.8 GB) in size

SQL Server 2019 Installer is at:

<https://download.microsoft.com/download/5/A/7/5A7065A2-C81C-4A31-9972-8A31AC9388C1/SQLServer2017-SSEI-Dev.exe>

Filename is: SQL2019-SSEI-Dev.exe 5,811 kb

SQL Server Management Studio download site:

<https://aka.ms/ssmsfullsetup>

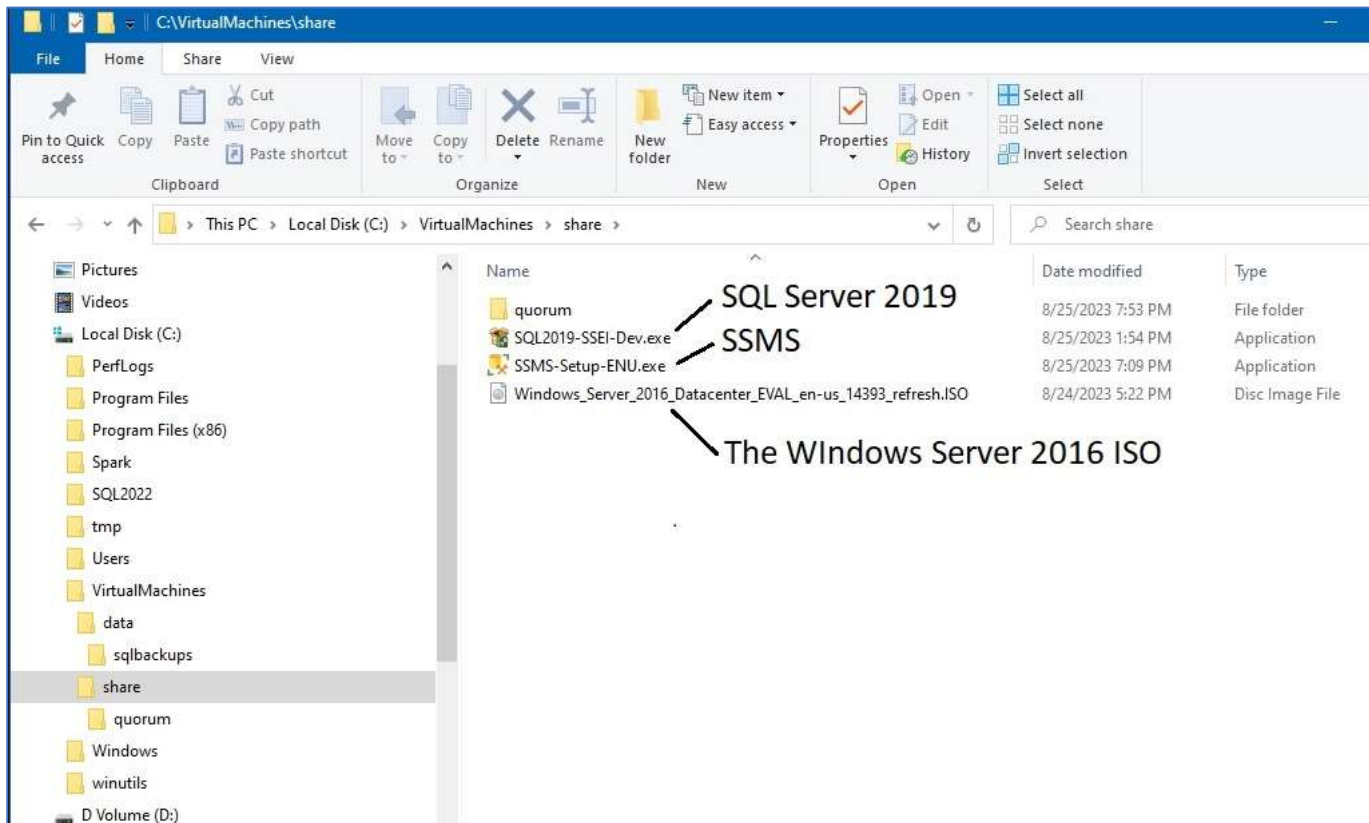
Filename is: SSMS-Setup-ENU.exe 644,163 kb

So now, on the Host OS, I placed these 3 files in this folder:

C:\VirtualMachines\share

A short note about Microsoft Windows Server 2016: Microsoft recommends that installations use the Server Core version of the operating system in production. This is also known as a "headless" installation that does not include the "Desktop Experience" GUI of Windows. "Real" administrators manage operating systems at the command-line level, and it is a much smaller footprint of the operating system. I am not using this. Even though that Training Kit recommended one of the OS installations be on the Core version – I may not do that yet.

So let's make sure we have these files where we want them:



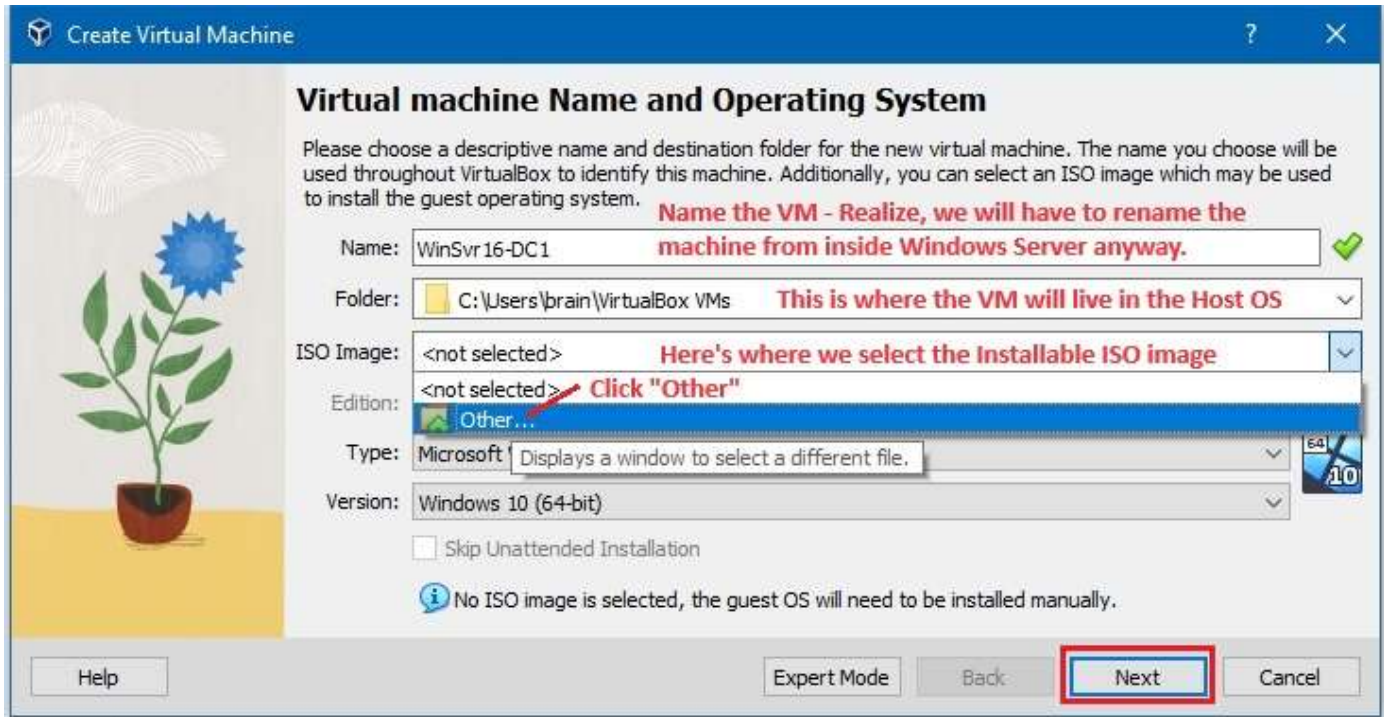
This will make thing easier later, in case your Virtual Machines have problems connecting to the Internet - which is really not necessary to do the Availability Group setup and demos later.

I've been having issues on the Virtual Machines, in that I can get them to either get on the Static IP's on the class D network (10.10.10.x) where I don't have access to the internet; or have the network adapter set up to obtain IP's automatically (DHCP) – and then it can see the internet. It's getting it to do both at the same time that's the trick. We're going to TRY to address that by having these machines assigned 2 network adapters.

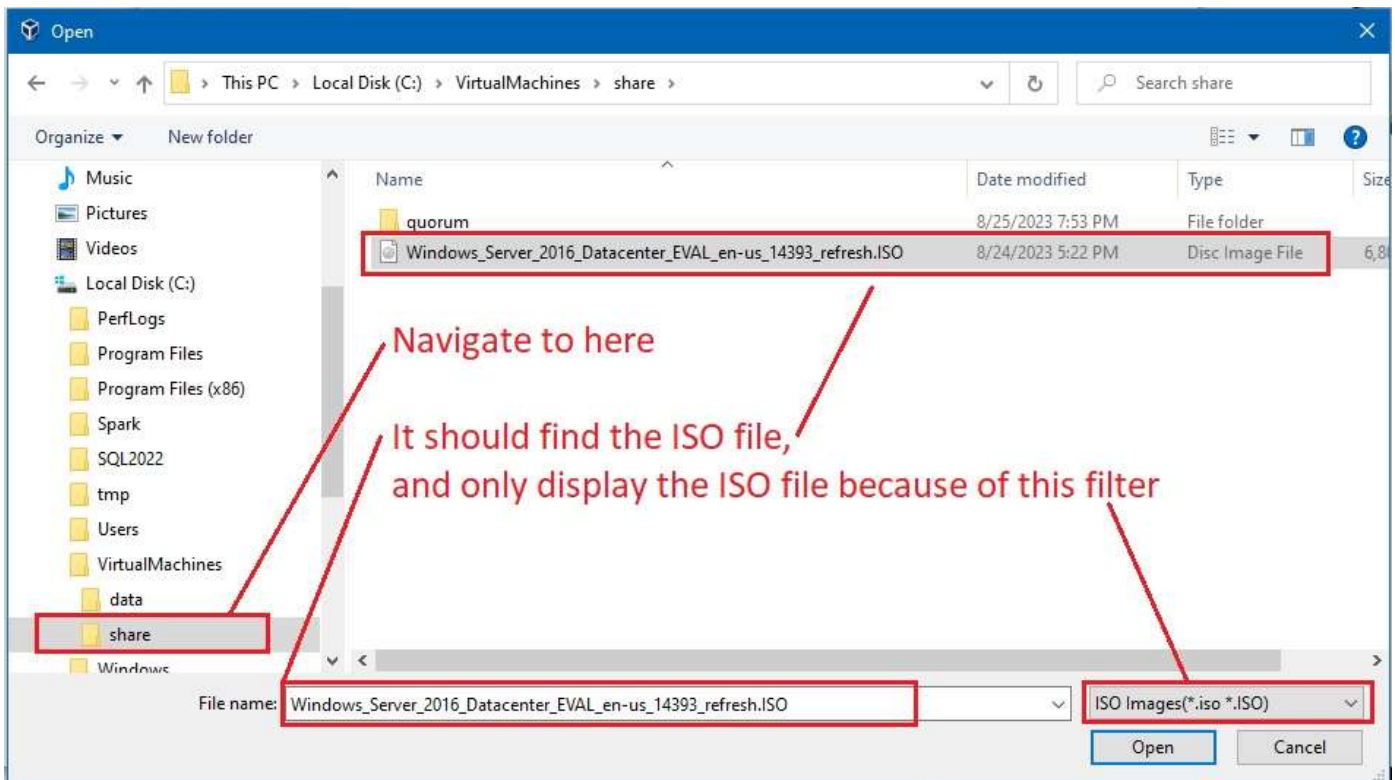
Fut for now, I'm going to assume that these Virtual Machines will be set up where they cannot get to the Internet – and having the installables available on this common share – will make it easier than having to download them each time I need them.

Step 4) Create the first Windows Server 2016 Virtual Machine

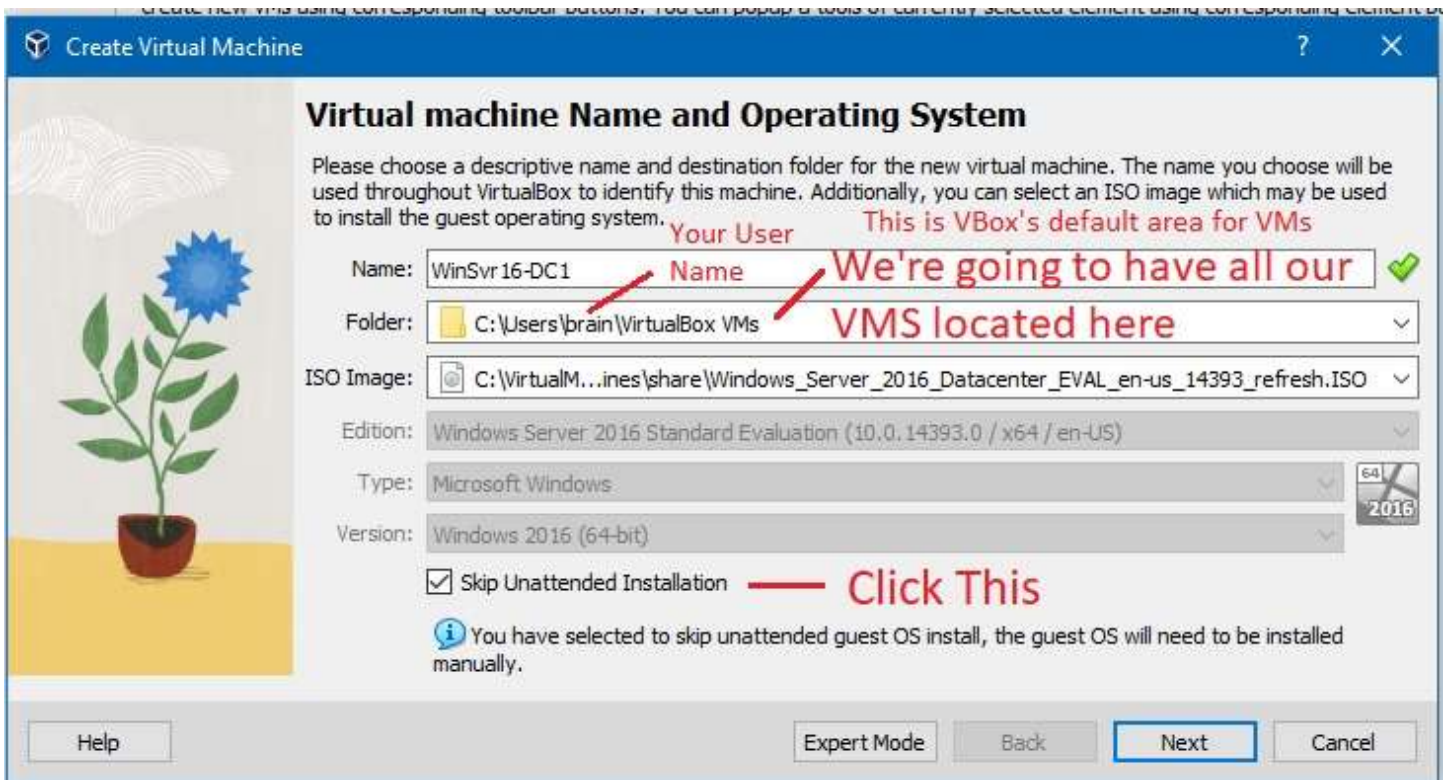
In VirtualBox, click Add a new Virtual Machine:



We have to tell it where to find the ISO file for the operating system we want: Windows Server 2016:



When it detects that this is a Windows Server ISO



!!IMPORANT!!

When doing the VirtualBox install of a Windows Server machine, it is important that you check the box for:

[x] Ignore Unattended Install.

If you do an unattended install, you will not get the chance to accept the Microsoft License Terms of Service (EULA). Even though the templates for VirtualBox unattended installs have that checked to "true" - The install will ignore those. If you do an unattended install, you'll get to a point where the virtual Windows Server begins a Setup - and will warn you that it can't find the terms of service. The only option will be to click "OK" - and it will re-start, complaining about the same thing. JUST DON'T DO AN UNATTENDED INSTALL. It's a hassle.

HOWEVER: There IS a way to do the unattended install, but it requires you to install the machine from a command line, using an MSI (microsoft installable) of the Windows Server.

This is documented on page 57 of the VirtualBox user manual.

For this demo - we're NOT going to use that Unattended install.

One other minor annoyance is that regardless of what we name the Virtual Machine in VirtualBox, the windows server install will ignore that, and give it its own automatically generated name. Have you noticed tht it does that since Windows 10? I'm not sure when they started generating machine names off of the OEM information, but it's a little annoying. After our installation, the first thing we're going to do is go to Server Manager and change the name to one of our choosing.

Virtual Box now wants to know how much memory and cores you want allocated to this Virtual Machine:

The screenshot shows the 'Create Virtual Machine' dialog box with the 'Hardware' tab selected. On the left is a decorative illustration of a potted plant with a blue flower. The main area contains the following settings:

- Base Memory:** A slider set to 2048 MB, with a range from 4 MB to 32768 MB.
- Processors:** A slider set to 1 CPU, with a range from 1 CPU to 12 CPUs.
- ☐ Enable EFI (special OSes only)

Handwritten red text annotations are present:

- I will accept these defaults** (in the center)
- 2 GB RAM** (below the memory slider)
- 1 core as the CPU** (below the processor slider)
- Look at that AMD Ryzen with 12 cores...** (with an arrow pointing to the 12 CPUs mark on the processor slider)

At the bottom are buttons for 'Help', 'Back', 'Next', and 'Cancel'.

And how much drive space it will allocate. I'm not going to pre-allocate this though.

The screenshot shows the 'Create Virtual Machine' dialog box with the 'Virtual Hard disk' tab selected. On the left is the same decorative illustration of a potted plant. The main area contains the following settings:

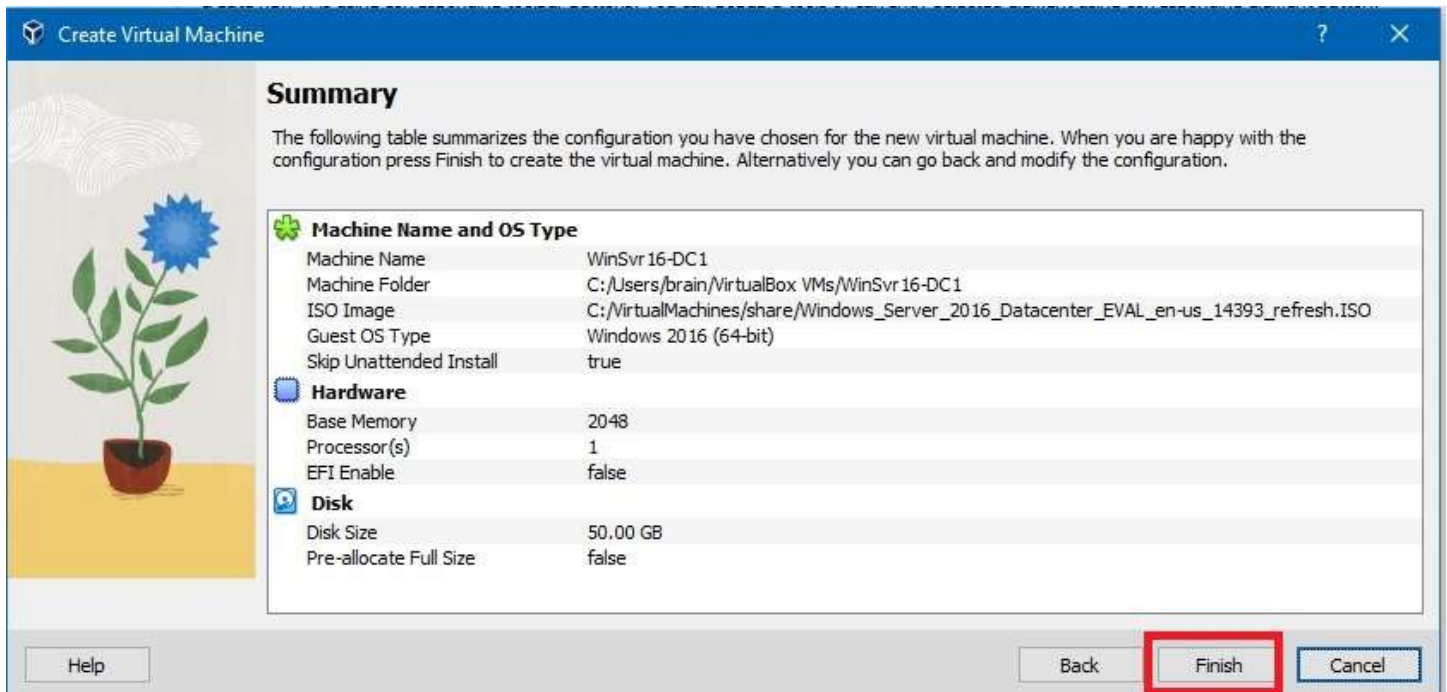
- ☒ **Create a Virtual Hard Disk Now**
- Disk Size:** A slider set to 50 GB, with a range from 4.00 MB to 2.00 TB.
- ☐ Pre-allocate Full Size
- ☐ Use an Existing Virtual Hard Disk File (with a dropdown menu showing 'Empty')
- ☐ Do Not Add a Virtual Hard Disk

Handwritten red text annotations are present:

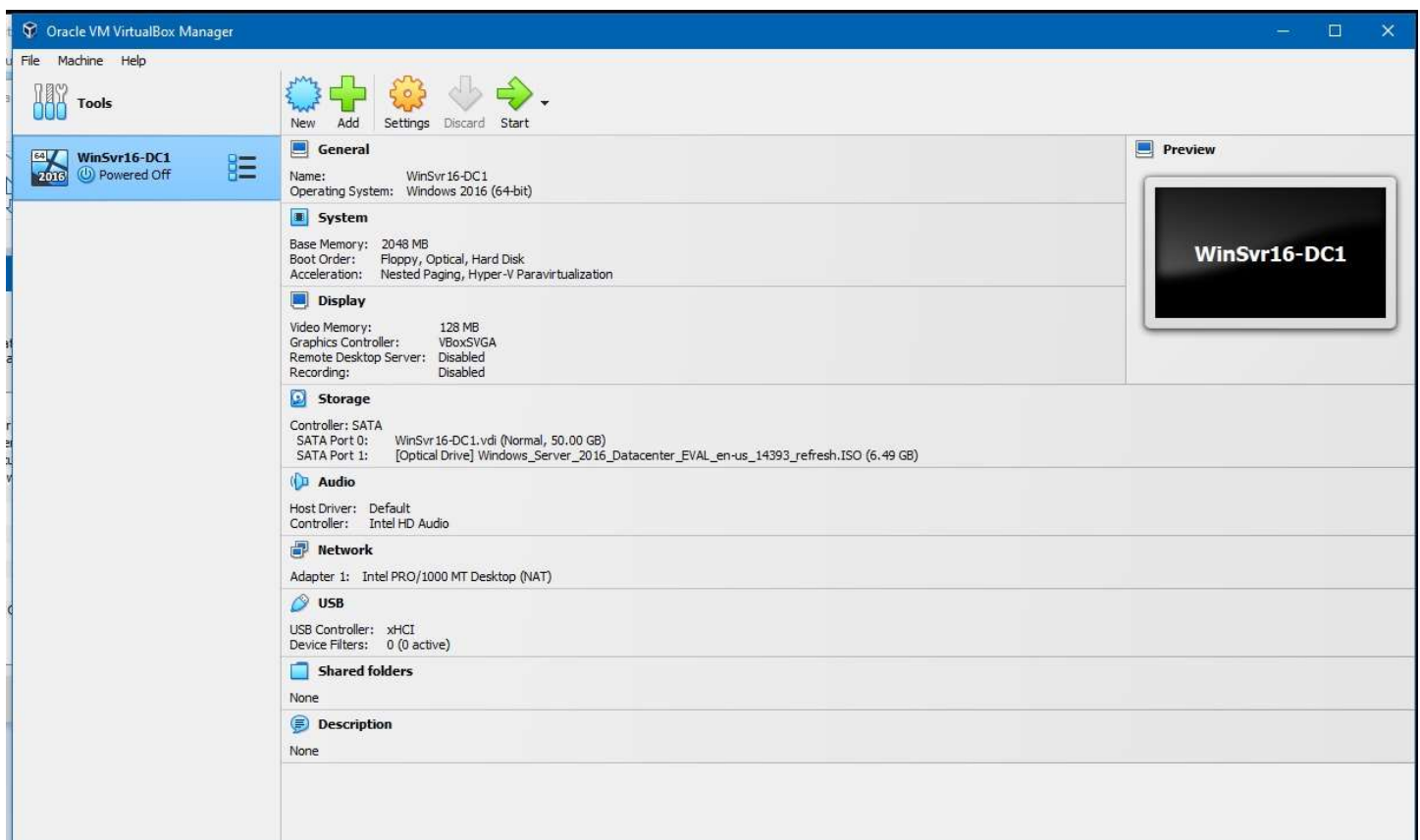
- It defaults to 50 GB which I will accept** (with an arrow pointing to the 50 GB value on the disk size slider)
- But it's not pre-allocating that much** (with an arrow pointing to the 'Pre-allocate Full Size' checkbox)

At the bottom are buttons for 'Help', 'Back', 'Next', and 'Cancel'.

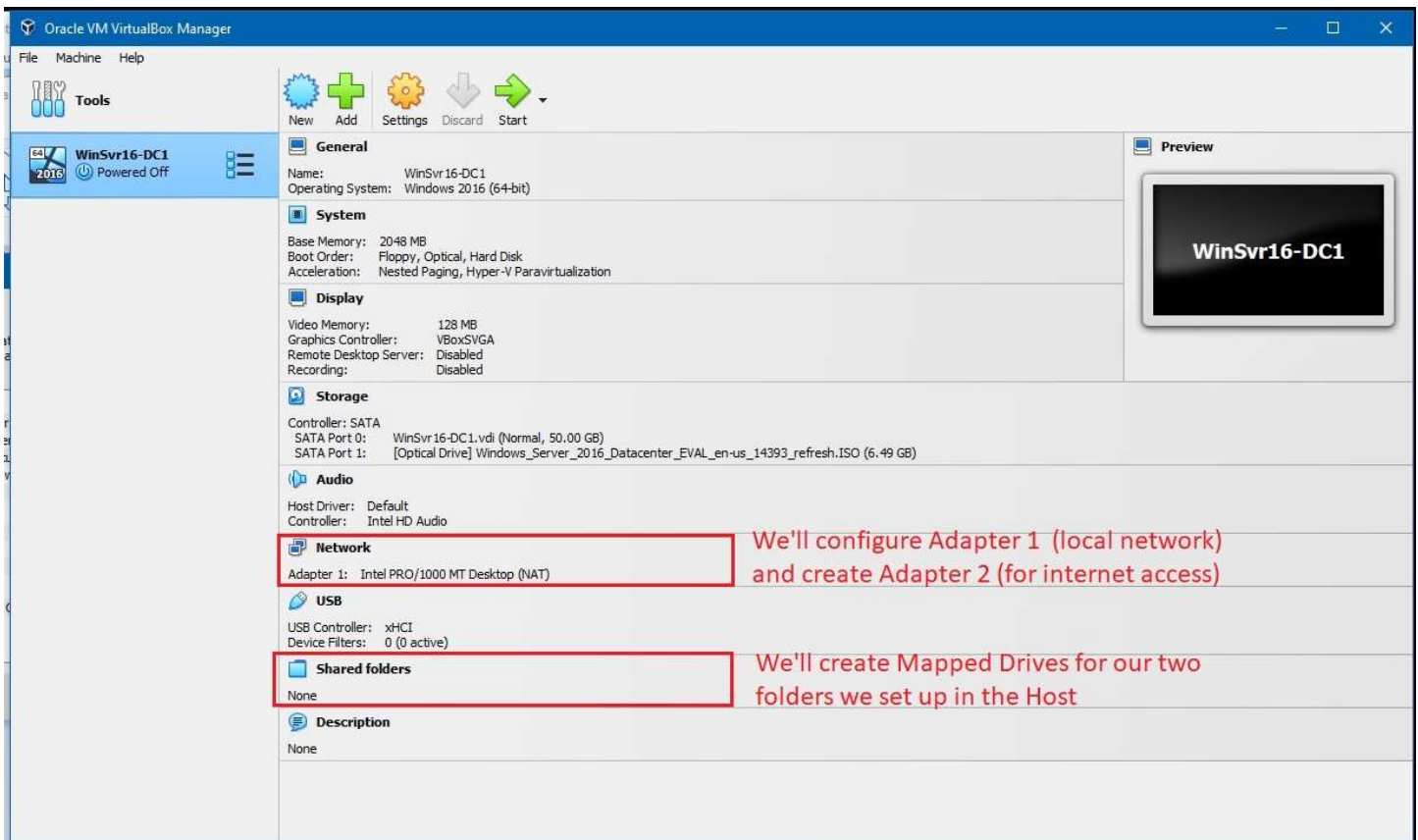
Summary



Click on Settings to see how the VM looks:



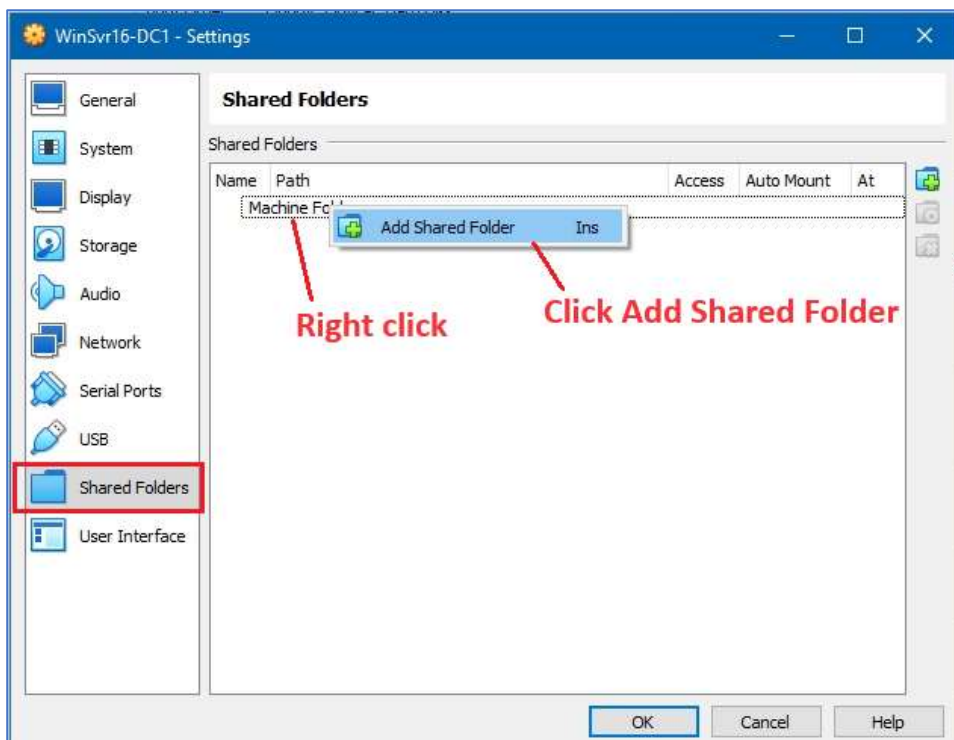
Here's 2 things we're going to change. Network Adapters and Shared Folders:

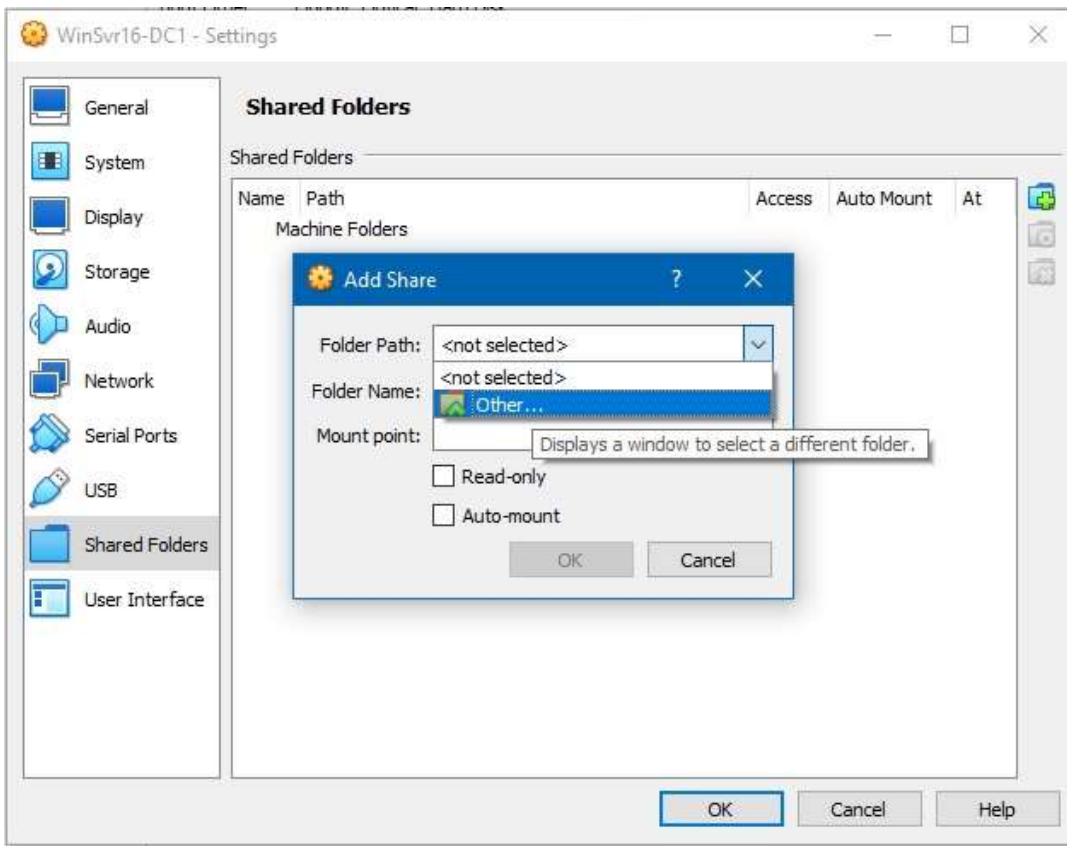


Step 5) From VirtualBox, create the two Shared Folders

In Virtual Box, click on the machine we just created.

Settings - click on Shared Folders. "Add new shared folder" button in top right corner.



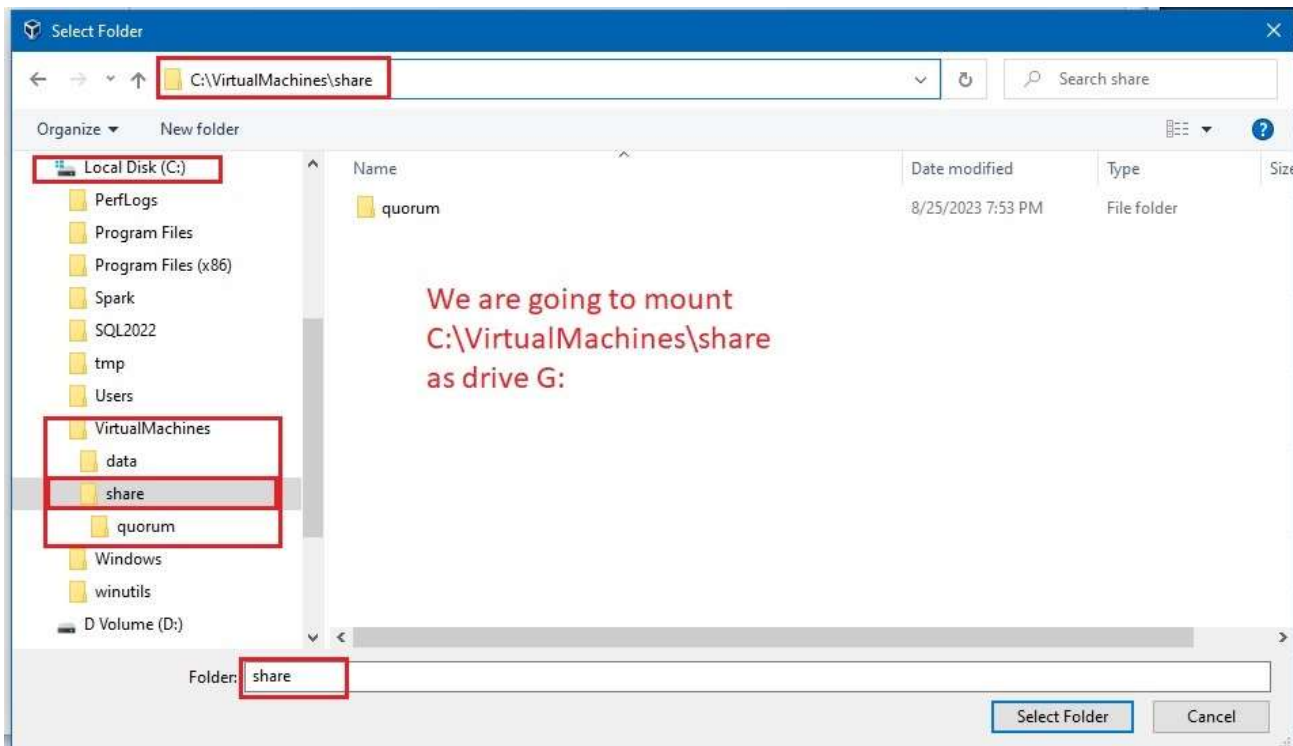


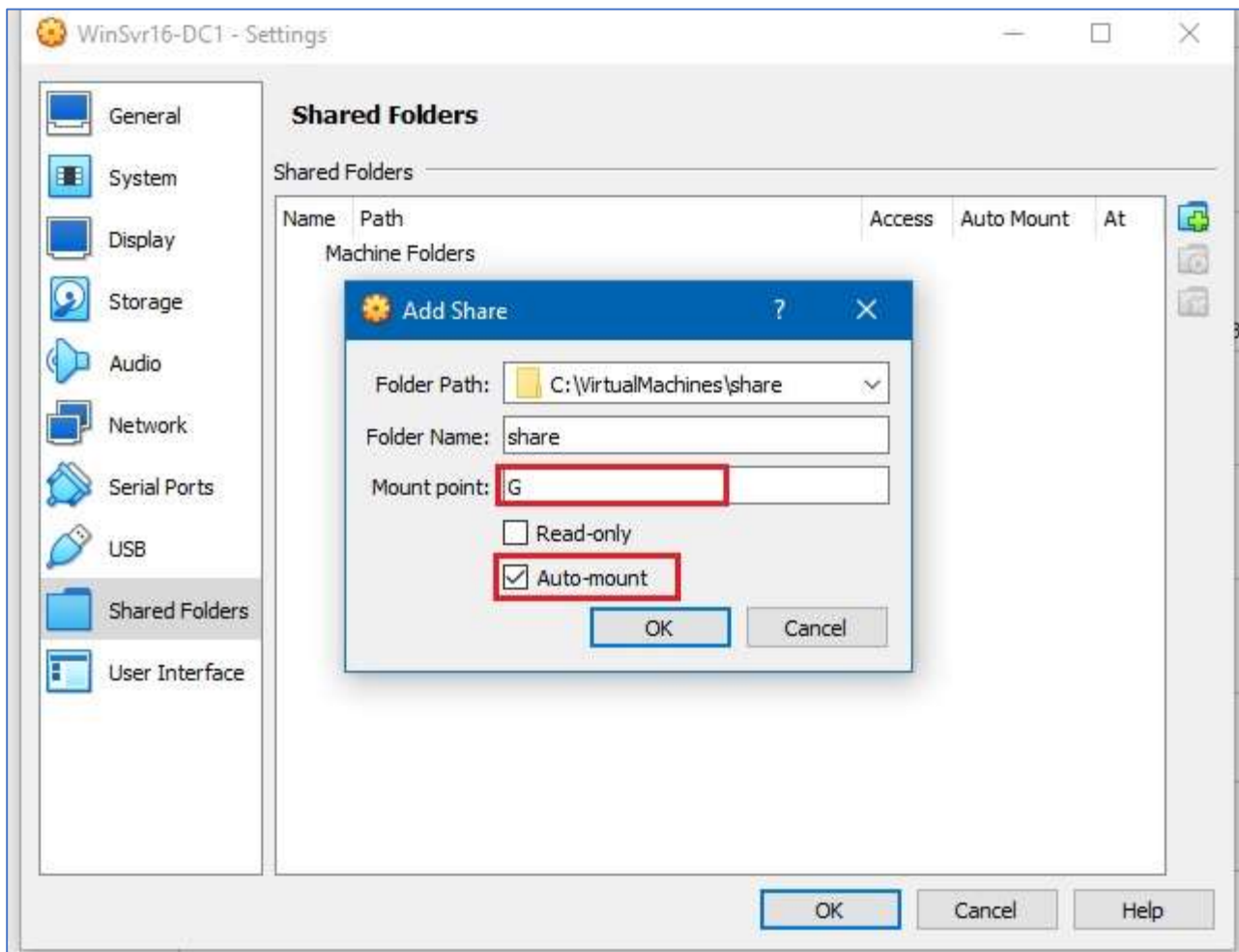
In the "Folder path" field, enter this path: C:\VirtualMachines\share

Check Auto-mount

Mount Point: G:

Check Make Permanent





Do the same thing for \data

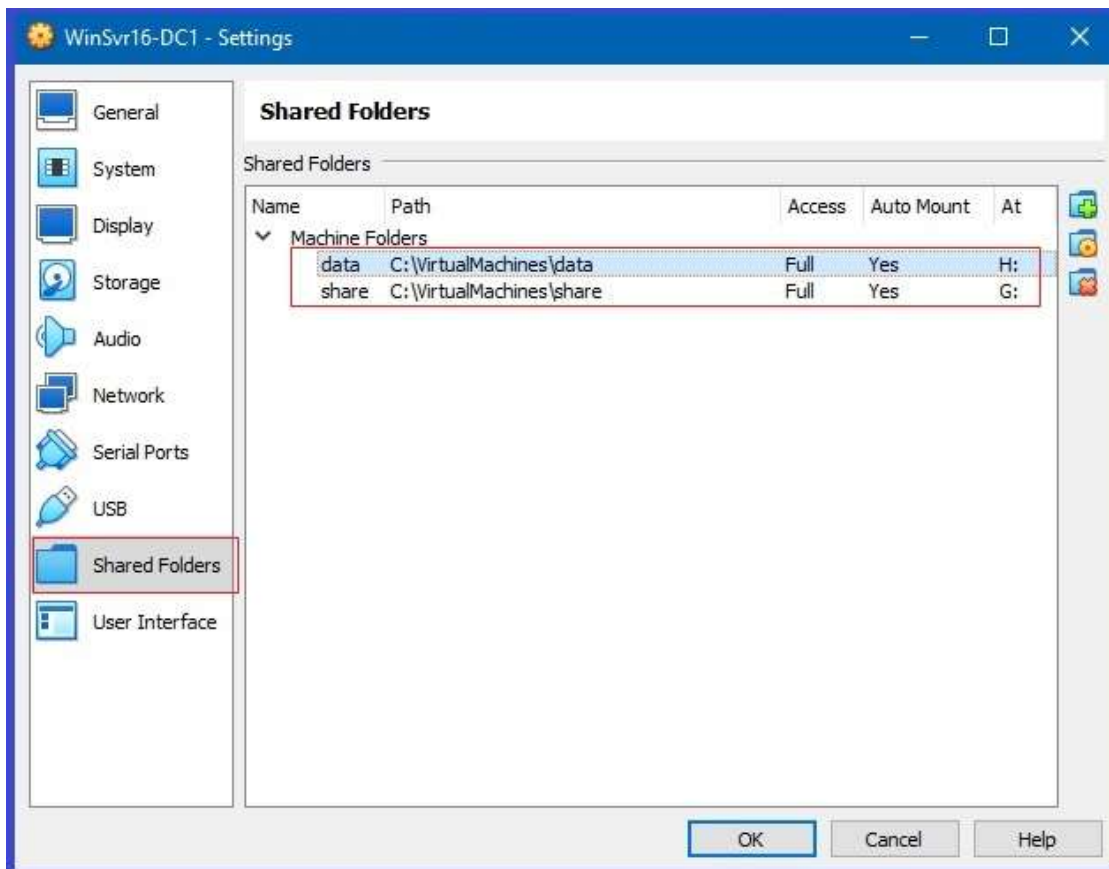
Folder Name: C:\VirtualMachines\data

Check Auto-mount

Mount Point: H:

Check Make Permanent

Now, under the Shared Folders, we should have:



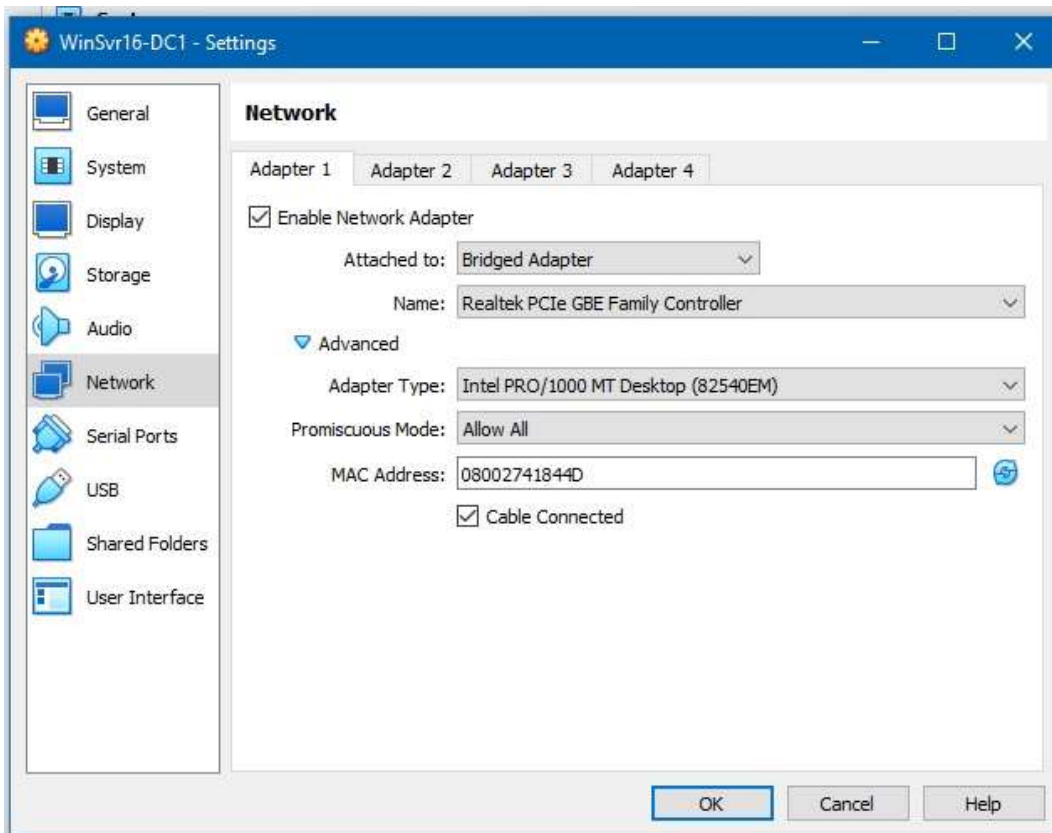
Step 6) From Virtual Box, set up two Network Adapters:

For all the virtual machines we're going to set up, each one of them will need to have 2 Adapters configured in Virtual Box:

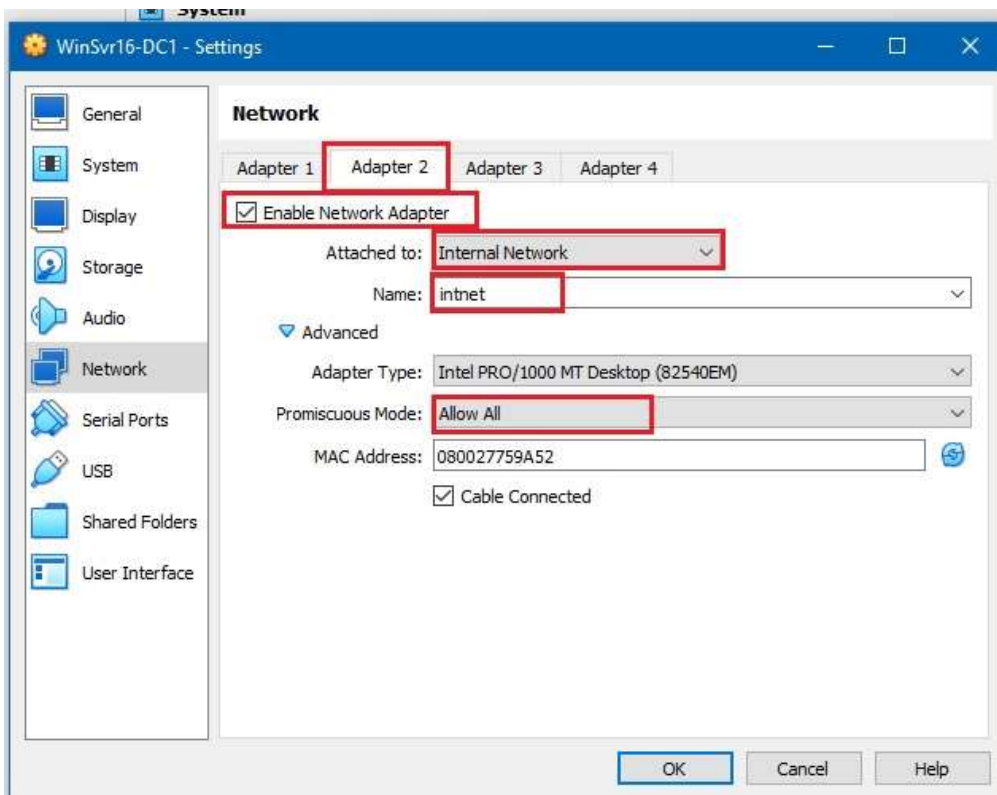
Adapter 1: For access to the domain's small class D network (the 10.10.10.x one).
This should be 'NAT' or 'Bridged-Adapter'

Adapter 2: For outside internet access. This should be 'Host-Only' -OR- 'Internal Network'

Let's configure Network Adapter 1



And now Network Adapter 2



Step 7) Start the Virtual Machine

So here's a summary of this first Windows Server:

Host Name:	WinSvr16-DC1
Folder:	C:\Users\brain\VirtualBox VMs\
Virtual HD:	50 GB SSD
RAM Allocated:	2 GB RAM (2048 MB)
Pre-Allocated:	No
Network Connection:	Ethernet V4 (TCP/IP)
Network Type:	NET
IP Address:	10.10.10.1
Subnet Mask:	255.0.0.0
Default Gateway:	192.168.1.1
Preferred DNS:	127.0.0.1
Alternate DNS:	leave blank

Start the Virtual Machine (for the first time)

It will now go into the Installation of Windows Server 2016.

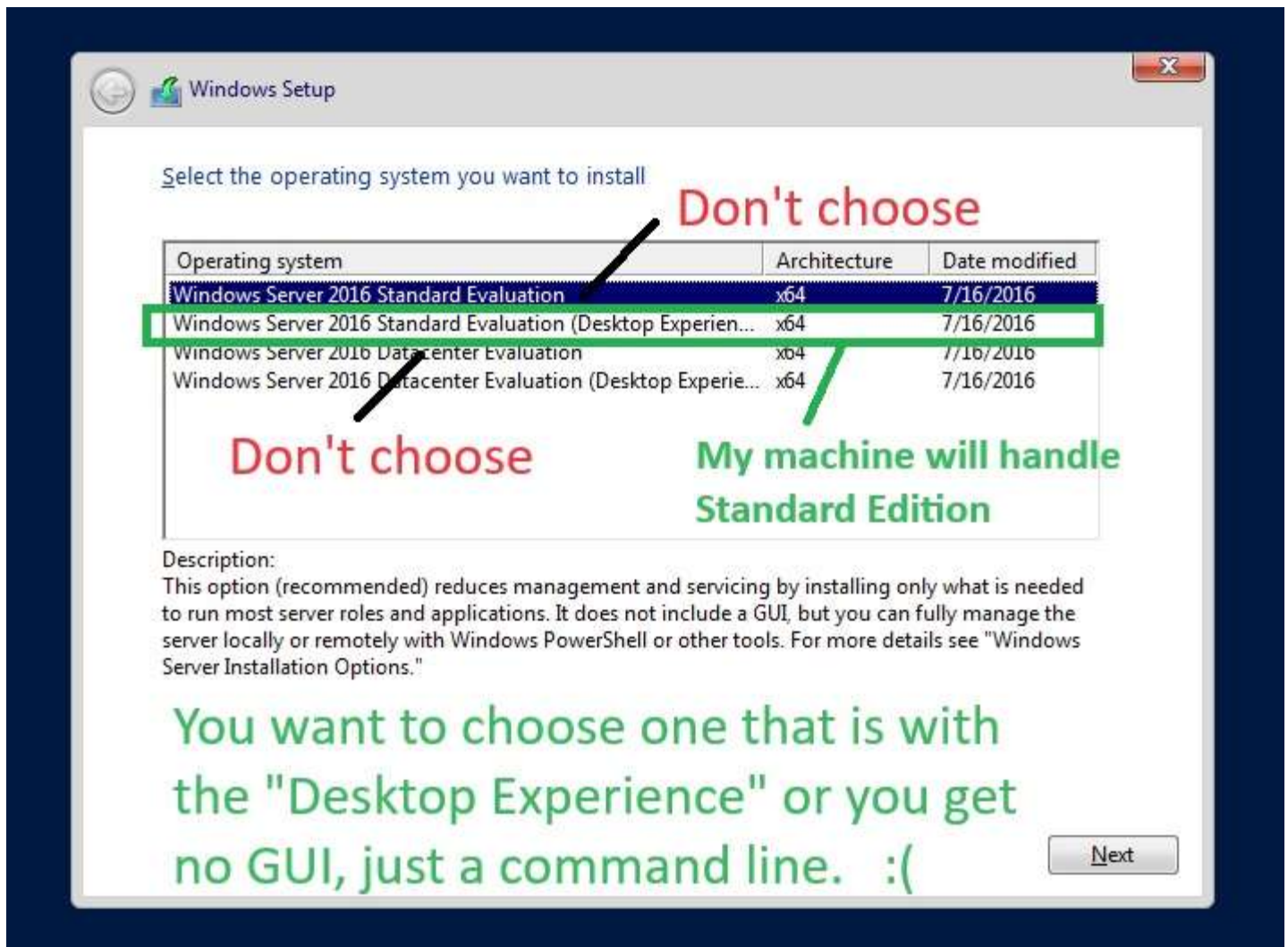
First it confirms your language and timezone



You will get warnings about the usage of the Mouse as you click inside the Virtual Machine, and how it will "capture" your mouse and keyboard. When you find your mouse captured within the virtual OS, you can always do CTRL-ALT-DEL and hit ESCAPE to return to your Windows 10 host.

To issue the CTRL-ALT-DEL command to the Virtual Machine itself, there is a menu item of "Input" on the top - select "Keyboard" - and you will get a selection of things to issue. CTRL-ALT-DEL is the first one listed. You will need this to Log into the guest operating system (Windows Server 2016).

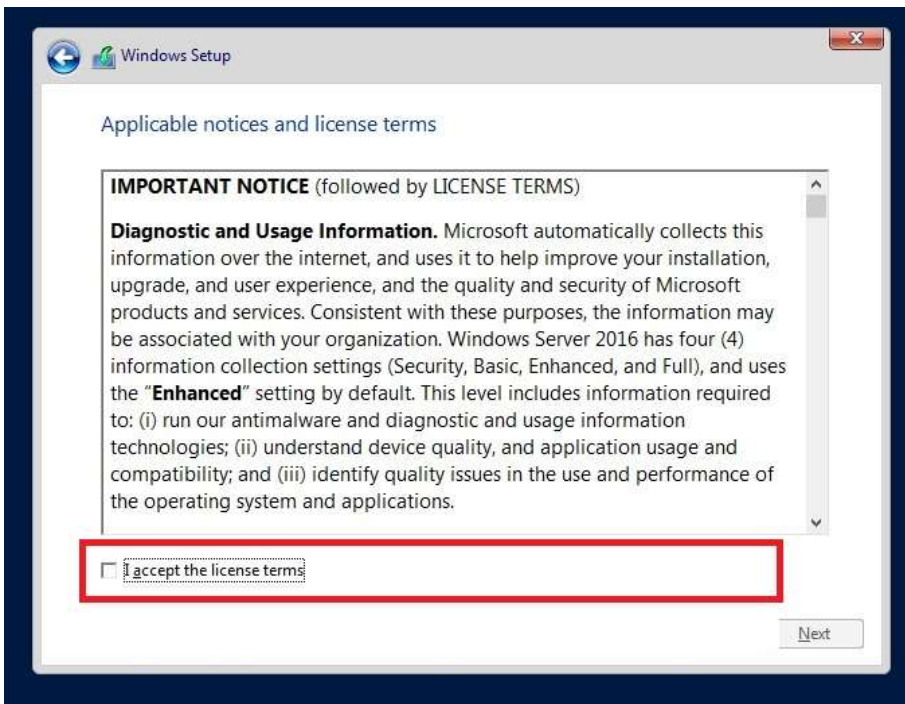
It now asks you to select the version of this operating system you want to install:



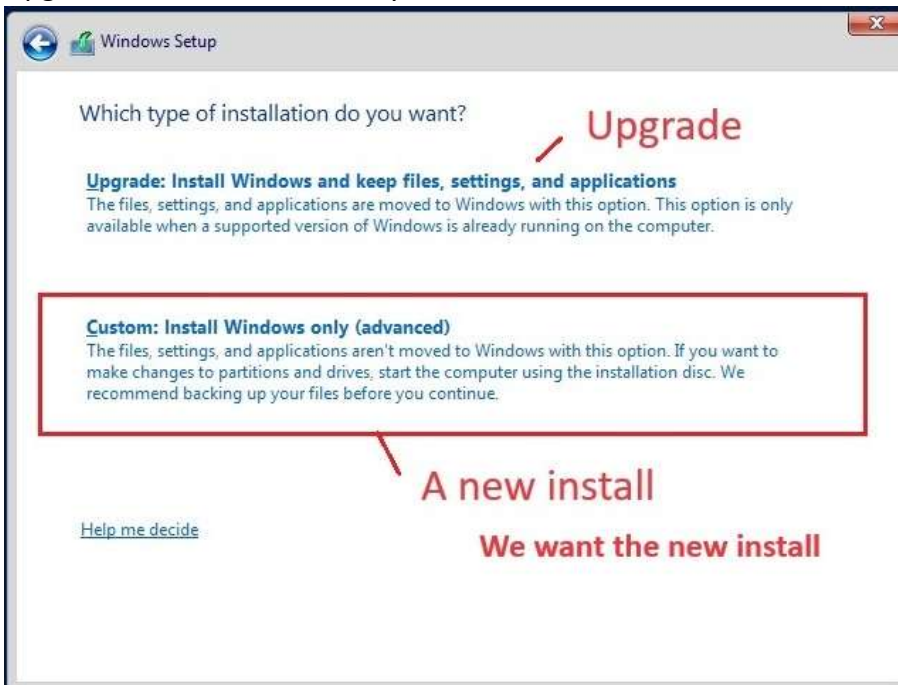
We will choose **Windows Server 2016 Standard Edition (Desktop Experience)** There are additional features in Data Center, but my hardware might not be able to use them (such as allocating min 2 cpus per node, SAN access, etc. Storage Replica sounds like something I'd like to test with - but for now, I'm thinking Standard edition will have to do.

MAKE SURE that you choose (Desktop Experience) - otherwise you will get a "headless install" where the machine is running, but with no user interface. For most SQL guys, that's pretty frustrating. To fix it, shut down the machine from VBox - and when re-starting, you'll get the DOS prompt of Hit Any Key to choose startup from CD or DVD (you'll chose that, because it will start a setup from the ISO again) - it's the only way to re-install the OS.

Accept the Licensing Terms.

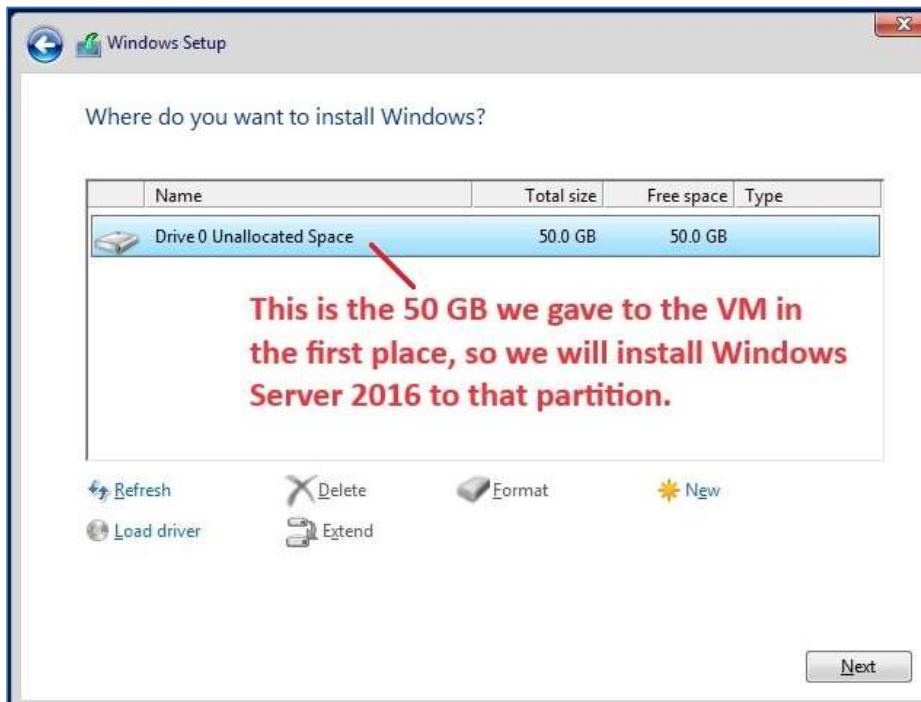


Upgrade or Custom Install only.



Choose the second one. Full install on new machine.

Specify where to install Windows Server: Use Drive 0 Unallocated space

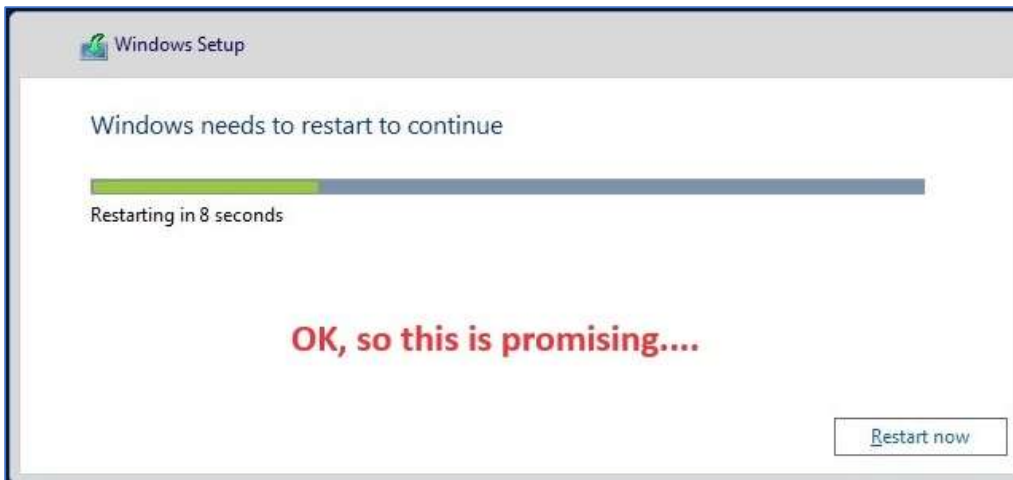


It will now continue with the installation of Windows Server:

- Copying Windows
- Getting files ready for installation
- Installing features
- Installing updates
- Finishing up



Windows needs to re-start now:



NOTE:

When it Re-starts Windows, you will get a "DOS" style window that prompts:

Press any key to start from CD or DVD Don't do that.

The reason it's asking is because in the "CD-ROM" of the Virtual Machine – is still mounted with the Windows Server installation ISO. If we hit a key in those 5 seconds that it waits – it will try to Re-Install windows, like as if we were booting from the CD-ROM with the Windows installation media in it. We won't need to do that.

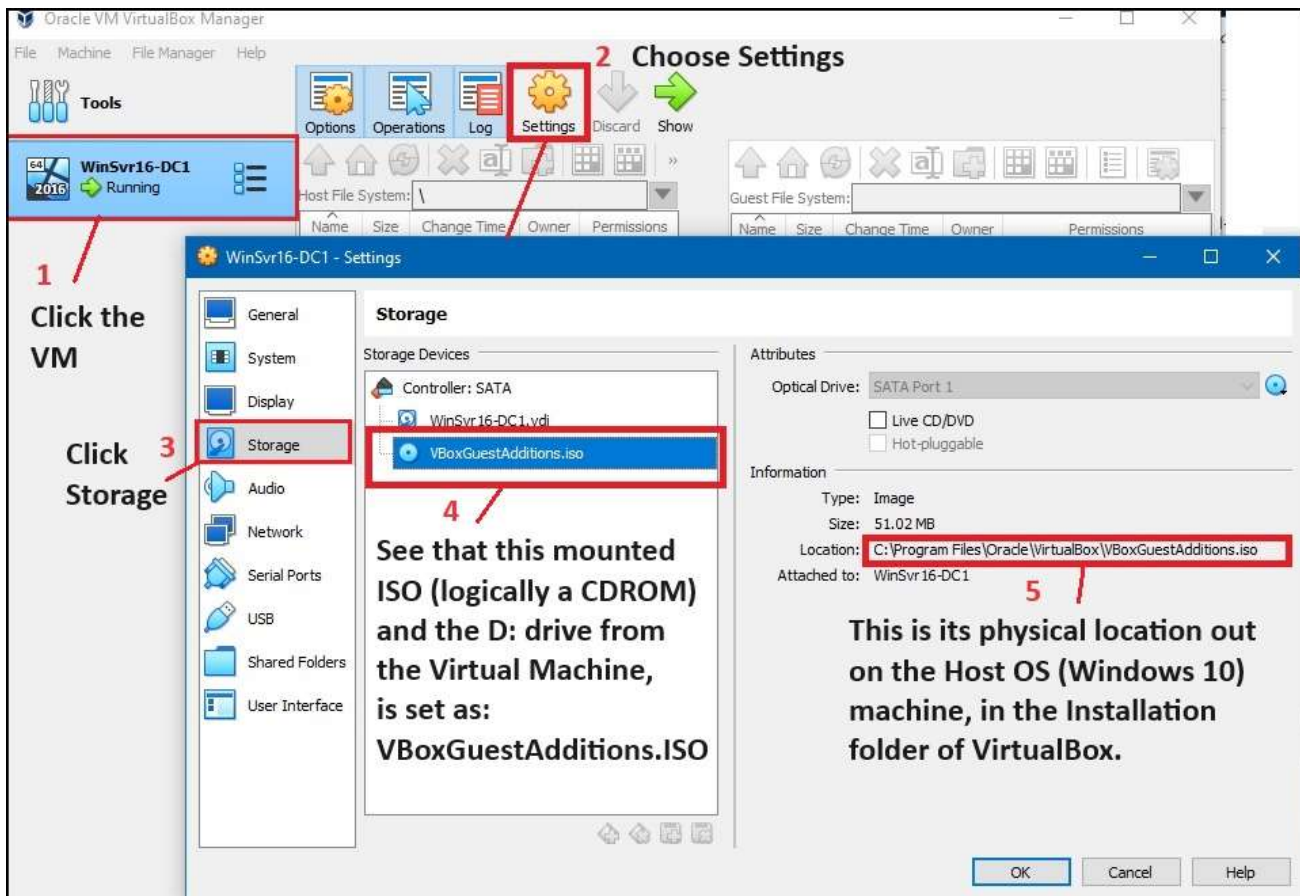
Step 8) Install Vbox Guest Additions

For the Virtual Machine to see the Shared Folders we set up in VirtualBox, as they are on the Host machine, we need to run a special utility named: **VBox Guest Additions**. This comes as an ISO file with the installation of VirtualBox. It's the only way to get the Virtual Machines to see Shared Folders that exist on your Host.

In Virtual Box on this Virtual Machine: WinSvr16-DC1, go to Details - Storage.

It will show that the "CD/DVD" (of which we only get one), is currently set to the Windows Server 2016 ISO image. Remove the current mapping, and change it to this file:

C:\Program Files\Oracle\VirtualBox**VBoxGuestAdditions.iso**



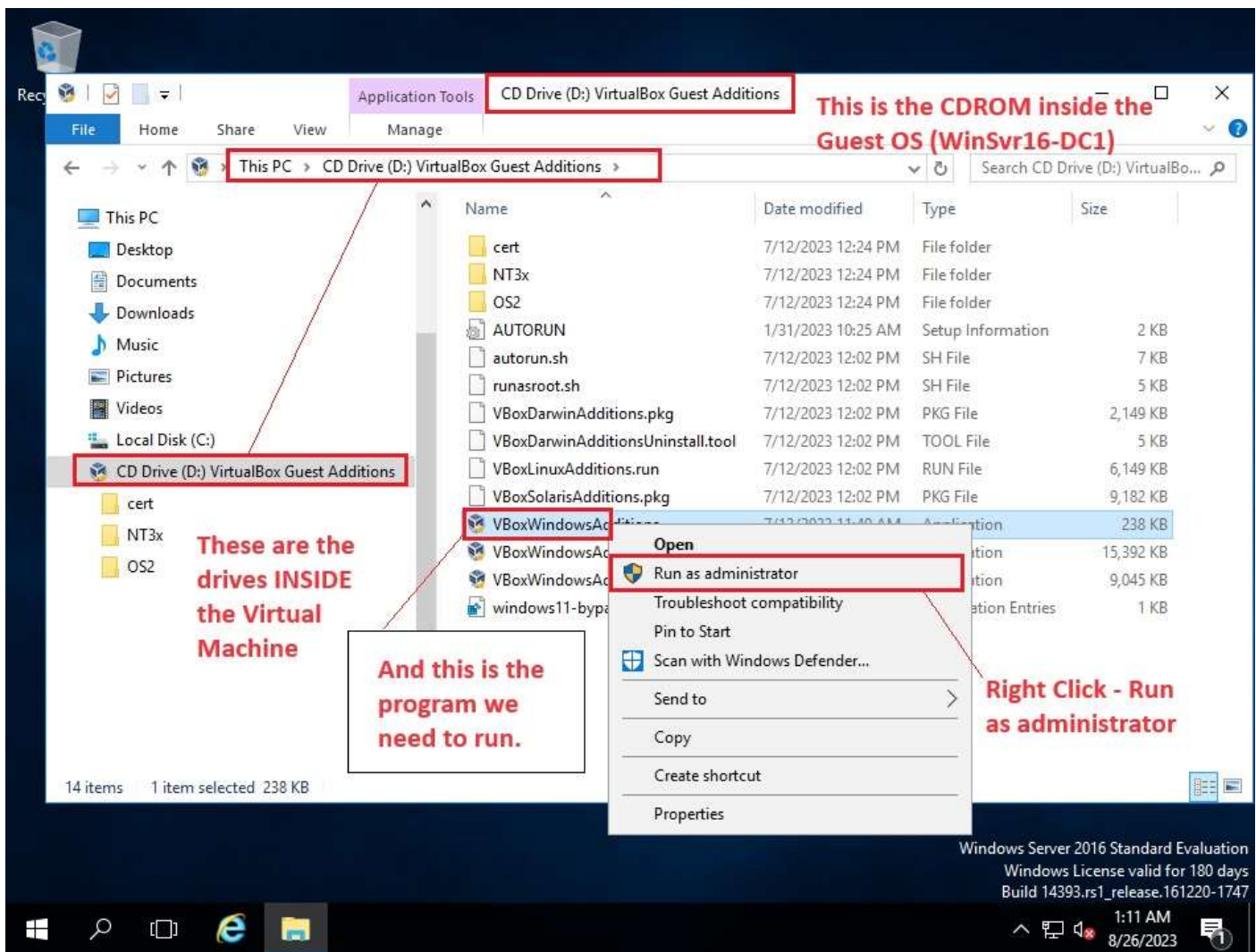
Then, after the "CD-ROM" is re-mounted to this ISO file:

Do we have to Re-Start the VM for it to see this?

Go back into the Virtual Machine and open the File Manager. It should now show D: as
CD Drive (D:) VirtualBox Guest Additions

Open this, and in the main folder, find the file inside it that is VBox Guest Additions Windows as an Application.

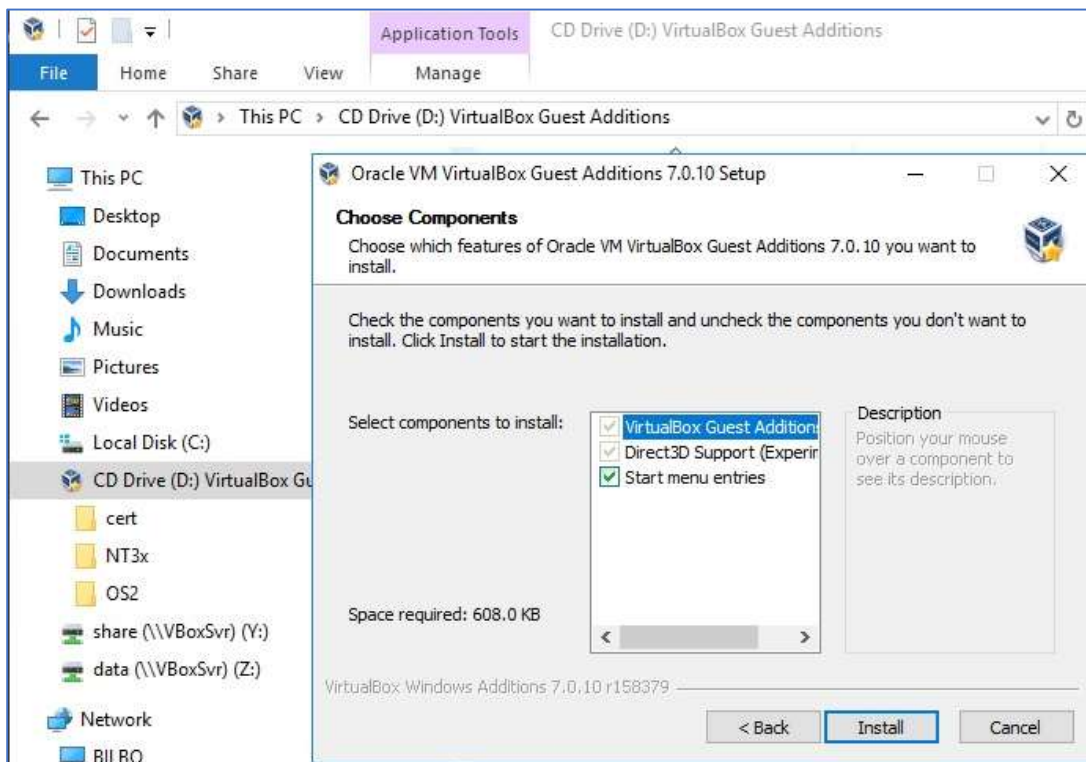
(It won't show it as a .exe unless you change the Windows Explorer setting to show the extensions of known file types - but that's not necessary at this point. You can run this from that mounted ISO.



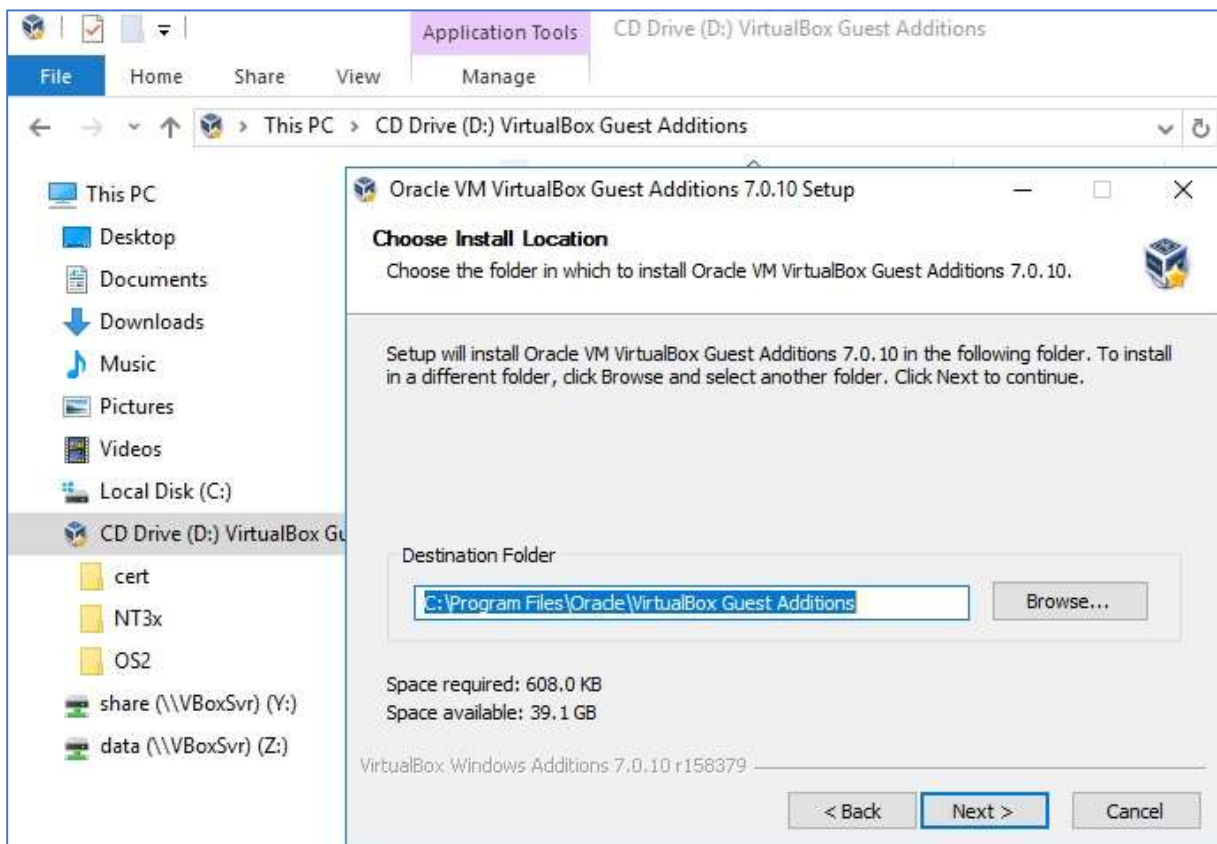
It will do the install of the utility.



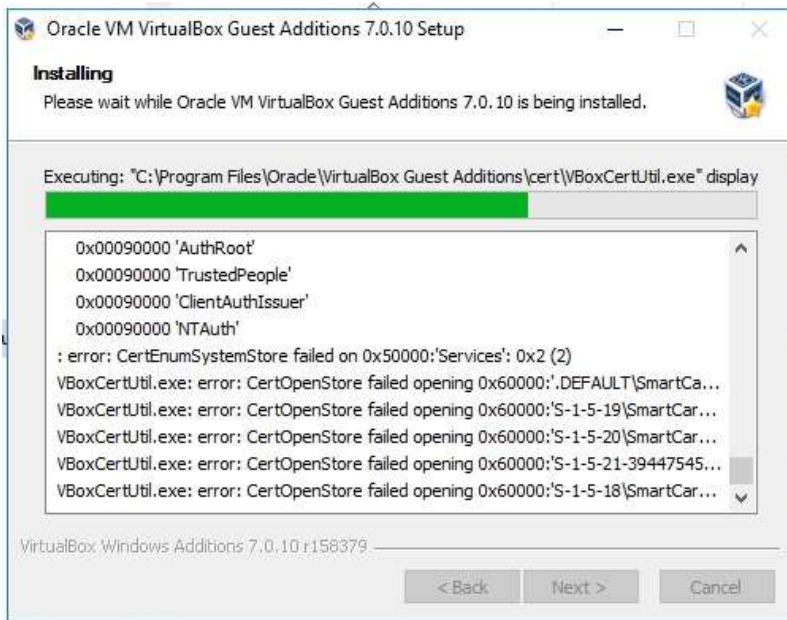
It will prompt to choose the components to install. Accept the defaults here:



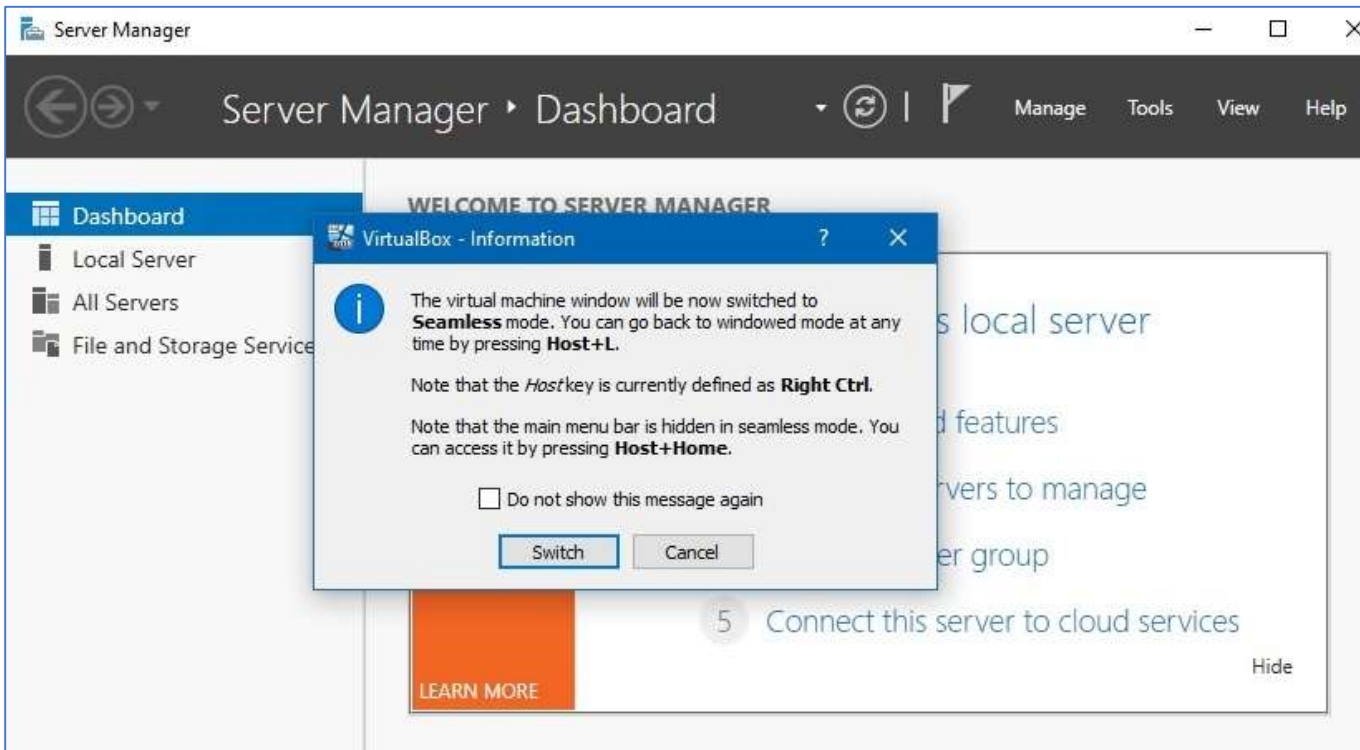
It will ask where to install the program. Accept its default. It now does the install.



And it will do the installation:



At the end of the run, it may say that its changing your display settings to be "**Seamless Mode**" This will change your user experience with respect to the Virtual Machine!

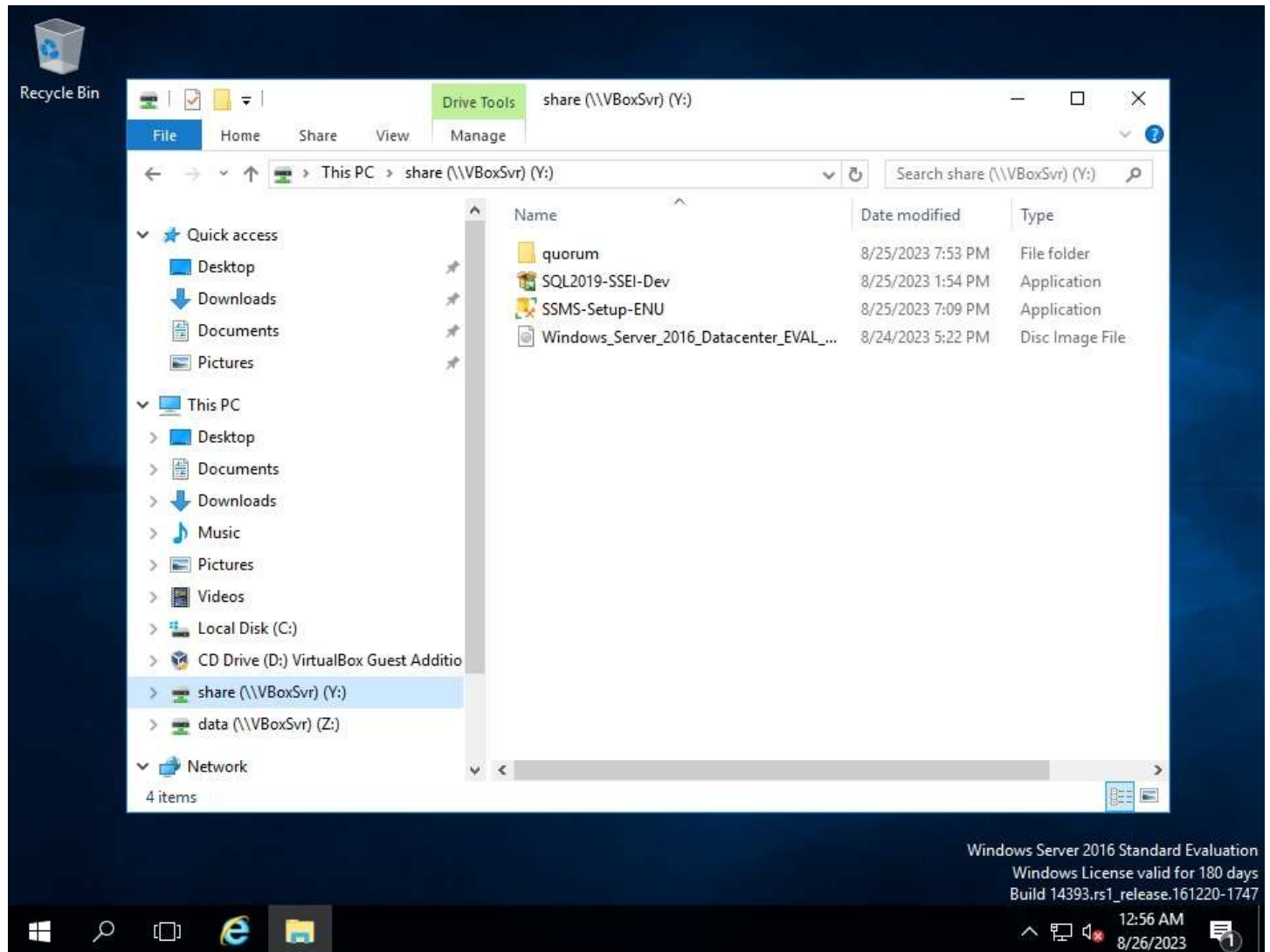


You may get the ability to drag program windows "out" of the Windows box it was in. You may also be able now to use the Print Screen button - and get screenshots of that Windows VM - which before, we needed to click on the top menu bar on "View" - and "Take Screenshot" - which was kind of annoying.

But the BIG THING is that it has now mapped those two shared folders for us, but it seems to have ignored our drive mappings, and made them:

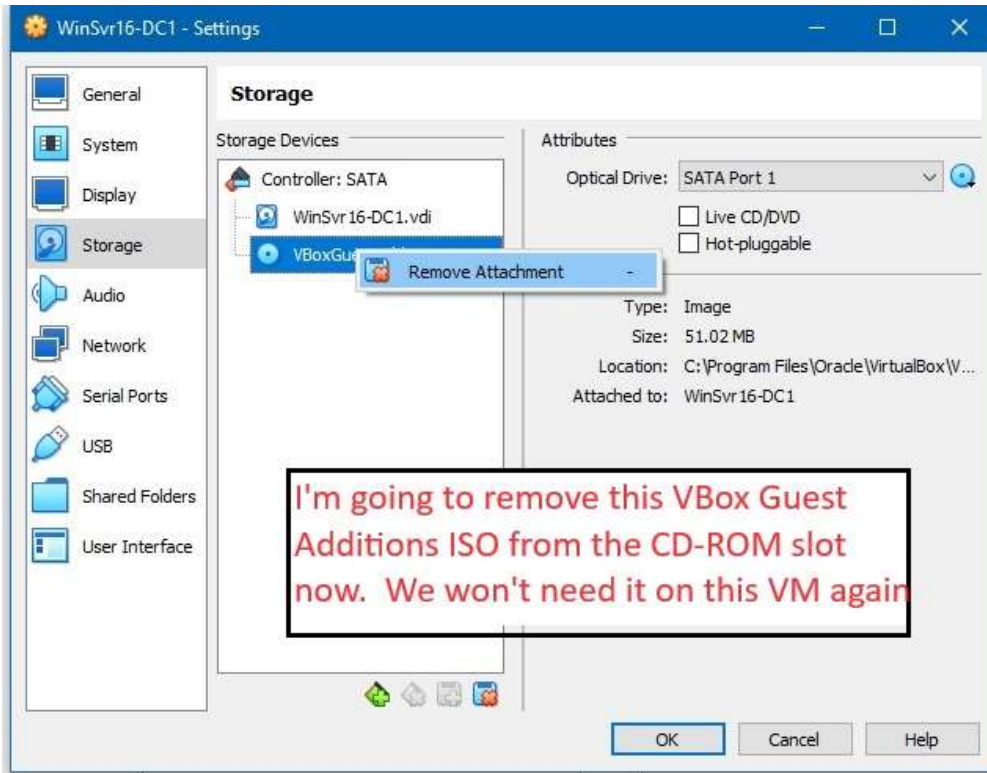
Y: is \share (\\VBoxSvr\share)

Z: is \data (\\VBoxSvr\data)



I don't care what the drive letters are, but I do kind of want them to be consistent across all the VMs. I guess we could re-map them to G: and H: But at this point, I really don't care where they are, just as long as they are mouted to drive letters.

You may want to re-map the Storage for SATA1, the CD ROM - back to the ISO image for Windows Server - just in case we need to install additional features of Windows Server.



Realize though that if that Windows Server 2016.ISO is the CD ROM - Every time Windows boots, you will have a small prompt for a few seconds that says "Press Any Key to boot from CD-ROM" - which is not a great annoyance, until we no longer need to install new features. After we're done with it - I would un-map that altogether (in VBox, Details, Storage). At some point, we won't need anything mounted as a CDROM inside the Virtual Machines.

Re-Start the Virtual Machine: WinSvr16-DC1.

Step 9) Configure the Network Adapters in the Virtual Machine

Back in Virtual Box, we made two Network Adapters available to this machine.

We still may need to make some changes inside Windows Server on that.

Control Panel – Network Settings – Right click the primary Ethernet Adapter
Right Click Ethernet TCP/IP v4 in the list – go to Properties:

Network Connection: Ethernet V4 (TCP/IP)

Let's make sure they are set to Static Ips:

Network Type:	NET	
IP Address:	10.10.10.1	
Subnet Mask:	255.0.0.0	
Default Gateway:	192.168.1.1	This is my router's address
Preferred DNS:	127.0.0.1	
Alternate DNS:	leave empty	

On the second Ethernet Adapter – we may want to try to see if we can get those to point to the Host machine, and get it so it can reach the Internet. We don't really have to do that though.

Step 10) Server Manager – Change the Name of this Windows Server

Go to Server Manager. Click the computer name in the upper left-hand side.

You are probably going to have to re-name this computer. Even though we named it in VirtualBox, Windows seems to have given it a name of its own. We really want it to have a name we decided on, since this is going to become our domain controller.

Change the computer name to: **WinSvr16-DC1**

This will require a re-start

You *might* be able to get to the Internet on Adapter 2. It will not really matter to much if you can't, as we already retrieved the installable files for SQL Server 2019 and SSMS ahead of time. We can play around with that later if we want.

Step 11) Make this machine a domain controller:

Server Manager: Add Roles and Features

(*) Role-based or feature-based installation

(*) Select a server from the server pool (this should be the only one available) - choose it. Next.

In Select Server Roles, check: Active Directory Domain services.

In Select Features, keep the defaults.

Confirmation:

Active Directory Domain services
Group Policy Management

Remote server Administration Tools

Role Administration Tools:

AD DS and AD LDS Tools

Active Directory module for Windows PowerShell

AD DS Tools

Active Directory Administrative Center

AD DS Snap-ins and Command-Line Tools

Install

Installation progress:

Under Active Directory Domain services, click the link to:

"Promote this server to a domain controller"

Active Directory Domain Services Configuration Wizard.

Deployment Configuration

(*) Add a new forest

Root Domain name: validien.local

← Use whatever domain name you want.

Validien Solutions is the name of my LLC 😊

Domain Controller Options:

☒ Domain Name System (DNS) server

☒ Global Catalog (GC)

Password: Pa\$\$word123!!

Confirm password: Pa\$\$word123!!

Next

Select the defaults on the next few screens. It will go through a Prerequisites Check - and will provide some warnings. We can usually ignore these. There will probably be a re-boot. When it comes back, we should have a good new domain controller set up.

Step 12) Add an Active Directory domain user account.

While logged into the domain controller machine, go to Server Manager - Active Directory Users

Click on the domain: validien.local

Right click Users - New - User: For me, I'm just going to use my name.

New Object – User

First Name: Rick

Last Name: Bellows

User logon name:

Rick @validien.local

User login name (pre-Windows 2000)

VALIDIEN\ Rick

Next

Set a password: Pa\$\$word123!!

☒ Password never expires

Finish

Step 13) Join this User account to the Domain Admins group.

Under Active Directory Users and Computers - choose the domain - click Users.

Locate the new user alphabetically in the list, right click it - Properties:

Click the "Member Of" tab: Right now it will just be in "validien.local/Users"

Click the "Add" button:

Make it a member of "Domain Users"

Make sure Object Types is "Groups or Built-in security principals"

From this location: validien.local

Enter the object names to select:

Type in "Domain Users" to locate that group. Make sure it is selected (underlined) and click the "OK" button.

OK – we have our first Virtual Machine, and it is a Domain Controller set up with a User in an Active Directory.

Step 14) Install another Windows Server for our 1st SQL Server

You're going to need to Steps 4 through 7 for each of the SQL Servers.

Windows_Server_2016_Datacenter_Eval_en-us_14393_refresh.ISO

Host Name:	WinSvr16-SQL1	
Virtual HD:	50 GB SSD	
RAM Allocated:	2 GB RAM (2048 MB)	
Pre-Allocated:	No	
Admin Password:	Pa\$\$word123!!	Use whatever password you want
Network Connection:	(TCP/IP) V4	
Network Type:	NET	
IP Address:	10.10.10.10	SQL1= 10 , SQL2 will be 20, SQL3 will be 30
Subnet Mask:	255.0.0.0	
Default Gateway:	10.10.10.1	- The domain controller's IP
Preferred DNS:	10.10.10.1	- The domain controller's IP
Alternate DNS:	leave blank	

From steps 4 through 7, after you have:

- Installed Windows Server 16
- Configured the Shared Folders in Virtual Box
- Configured the Network Adapters
- Installed the Operating System
- Re-named it appropriately
- Run the Vbox Guest Additions

You will be ready to continue.

This is why the title of this document says "in 30 steps, but more like a hundred"

Since those steps will need to be done on SQL-1, SQL-2, SQL-3, and however many replicas we want.

Step 15) Add this computer to the domain

Log back in to WinSvr16-SQ1 and then add this computer to the domain:

Click on this server name.

Click "Change"

"To rename this computer or change its domain or workgroup, click Change". Click Change button.

Member of

(*) Domain

validien.local

OK

You will have to enter the user name and password we set up before:

Rick

Pa\$\$word123!!

After clicking OK, this computer should be joined to the domain, and say:

Computer Name/Domain Changes

Welcome to the validien.local domain

Step 16) Re-Start this server.

When starting back up, you should log in under:

Rick

Pa\$\$word123!!

Step 17) Install SQL Server 2019 to this machine.

Locate the file at: Y: named:

SQL2019-SSEI-Dev.exe

Installing SQL Server – Need to detail that. Will we be creating stand-alone instances, or using the "create an instance to be made a member of a cluster – option? Need to experiment with this and take screenshots.

Step 18) Install SQL Server Management Studio

Locate the file at: Y: named:

SSMS-Setup-ENU.exe

This should be pretty easy. It's very straight forward.

Install it on this first machine.

(And on each SQL Server later)

Step 19) Add second and third and fourth SQL Server machines

Create the VMs as shown for the second and third machines, just like the first SQL Server machine.

Recall Steps 4 through 7.

The machine names will be slightly different

The IPs will be slightly different.

Node:

WinSvr16-SQL1	10.10.10.10
WinSvr16-SQL2	10.10.10.20
WinSvr16-SQL3	10.10.10.30
WinSvr16-SQL4	10.10.10.40

Install SQL Server 2019 Developer edition, on each machine.

Installing SQL Server – Need to detail that. Will we be creating stand-alone instances, or using the "create an instance to be made a member of a cluster – option? Need to experiment with this and take screenshots.

Install SQL Server Management Studio on each machine.

This should be pretty easy. It's very straight forward.

The reason we are doing 4 nodes for SQL Server is because most of the exercises on the Training Kit use up to 4 separate instances.

Step 20) Install Failover Clustering on the first of the Server 2016 machines.

This is where we will set up the clusters, and SQL1 will be our primary replica.

But yes – we will end up doing this on each of the SQL Server machines

Server Manager - Manage/Add Roles and Features

Click "Next" until we get to "Select Features" - click the checkbox for "Failover Clustering" - Add Features

Add Roles and Features Wizard

Add Features that are required for Failover Clustering?

Check the "Include management tools" if applicable. - Add Features.

Confirm Installation selections

Don't check "Restart the destination server automatically, if required"

It should read: Failover Clustering

Remote Server Administration Tools

Failover Clustering Tools

Failover Cluster management tools

Failover Cluster Module for Windows PowerShell

Click "Install"

We'll need this set up on each of the SQL Server (Windows Server 2016) machines.

Step 21) On WinSvr16-SQL1, Create the Cluster.

Go to the first SQL Server machine

Server Manager - Tools (dropdown) - choose: Failover Cluster Manager

Create Cluster (on right hand side) - Add the servers that are going to be part of the cluster.

Under "Enter Server Name" box - type in the SQL servers one at a time.

Run through the Create Cluster Wizard. Observe the warnings. They should not be fatal.

Choose (*) Yes - when I click Next, run configuration validation tests and then return to the process of creating the cluster.

Access Point for Administering the Cluster.

Cluster Name: cluster1

IP Address: 10.10.30.1

Confirm the "Networks" selection.

Node: WinSvr16-SQL1

WinSvr16-SQL2

WinSvr16-SQL3

WinSvr16-SQL4

Step 22) Configuring Quorum and File Share Witness

From Cluster Manager, you should see the nodes as "Up"

We should have an odd number of votes.

If we have an even number of servers, create a File Share Witness.

Remember, we have a File Share accessible to the domain controller. Make ure that all the SQL Servers have full rights to it. We could say "EVERYONE" has full access to it (probably not a great idea in a non-Lab environment, but for here it's ok.)

The fully qualified name for this will be Y:\Quorum

Cluster Manager - More Actions - Configure Cluster Quorum Settings.

(*) Select the Quorum witness.

Next

Configure Cluster Quorum Wizard

Select Quorum Witness

(*) Configure a file share witness.

File Share Path:

Y:\Quorum

Next - Finish

Step 23) Confirm settings

Cluster Resources - should show Cluster Core Resources"

Server Name:

Name: cluster1	Online
----------------	--------

File Share Witness:

File Share Witness (G:\Quorum)	Online
--------------------------------	--------

Step 24) Turn off the Firewall.

We could open up just the ports needed (which would be preferred) - or just turn if off entirely.

So on the SQL Servers - Server Management - Tools - Windows Firewall with Advanced Security
- Windows Firewall Properties:

Go through each tab and change the Firewall State to "Off".

Domain Profile

Private Profile

Public Profile

IPsec Settings

Step 25) Set up a SQL Server Availability Group

In each SQL Server Windows machine, go to SQL Server Configuration Manager

SQL Server Properties - go to tab "AlwaysOn High Availability" - click 'Enable AlwaysOn Availability Groups'

Also, while in SQL Server Configuration Manager - Network Configuration - we must enable Protocol: TCP/IP

Double click TCP/IP - go to its properties, and choose dropdown Enabled: Yes

Re-Start the SQL Service.

Step 26) Setting up fo Availability Groups

Create some databases.

(Should it be Full Recovery mode?)

Take an initial full backup of those databases.

You must have at least one full backup to set up the Availability Groups.

In SSMS (SQL Server Management Studio) -

Which we should have installed on each SQL machine by this point.

Object Explorer – Always On High Availability - Availability Groups - Right click: New Availability group Wizard.

Specify Availability Group Options:

Availability Group Name: AG1

Cluster Type: Windows Server Failover Cluster

Next

Step 27) Choose the Databases to join the Availability Group.

Specify Replicas.

Replicas:

SQL-1 should be Primary

Others as Secondary

All set with Automatic Failover [x]

All with Synchronous Mode

Readable Secondary - No on all

BTW: If we were paying for all this, having Readable Secondaries would increase the number of core needing paid licenses.

Step 28) Set up the Listener

This will be the connection that applications will connect to, regardless of which replica is the primary.

Give this a name like:

Listener DNS Name:	MainSqlAG1	
Port:	1433	
Ip Address:	10.10.50.1	Notice the 50 in the 3 rd octet

Verify this Listener is active:

In any one of the SQL Server machines, in SSMS - try to connect to: MainSqlAG1

This would be the location that any client should be able to reach, and should point to whatever is the Primary replica that is Up.

Step 29) Synch the databases

Remember how we had a central File Share out on Host OS? We're going to want a folder on that just for SQL Backups. It doesn't have to be off the DC.

We're using the location on the Host that we mapped to drive Z:\SQLBackups

So, off of that New Availability Group - we go to Select **Initial Data Synchronization**

(*) "Full Database and Log Backup"

Specify the file share path in Windows format:

Z:\SQLBackups

The Validation will then happen on the new AG.

Hopefully everything is a Success.

Next - We get a summary

Click "Finish"

[Step 30\) Verify connectivity to the Listener from the Domain Controller](#)

Install SSMS on the Domain Controller

Make sure we can connect to MainSqlAG1. Try this as a final check.