Cookbook to transform a Teradata VMware Virtual Machine into a Hyper-V one

1. Introduction

Many vendors offer their products for demo purposes as VMware virtual machines. However, we may need to use Hyper-V hypervisor instead.

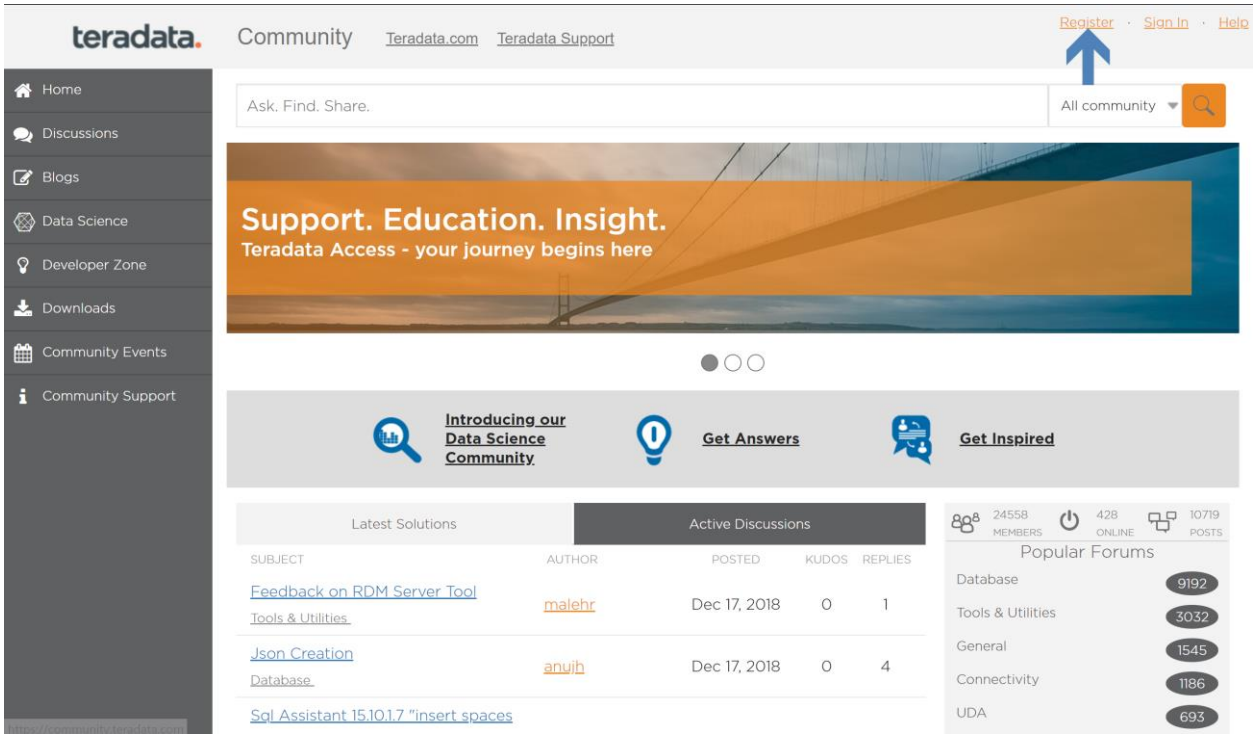
This post is a cookbook to transform a VMware virtual machine to a Hyper-V one. We are going to use a Teradata on VMware virtual machine as an example.

The detailed documentation about Teradata is here, and about the Teradata on VMware and its clients can be found here.

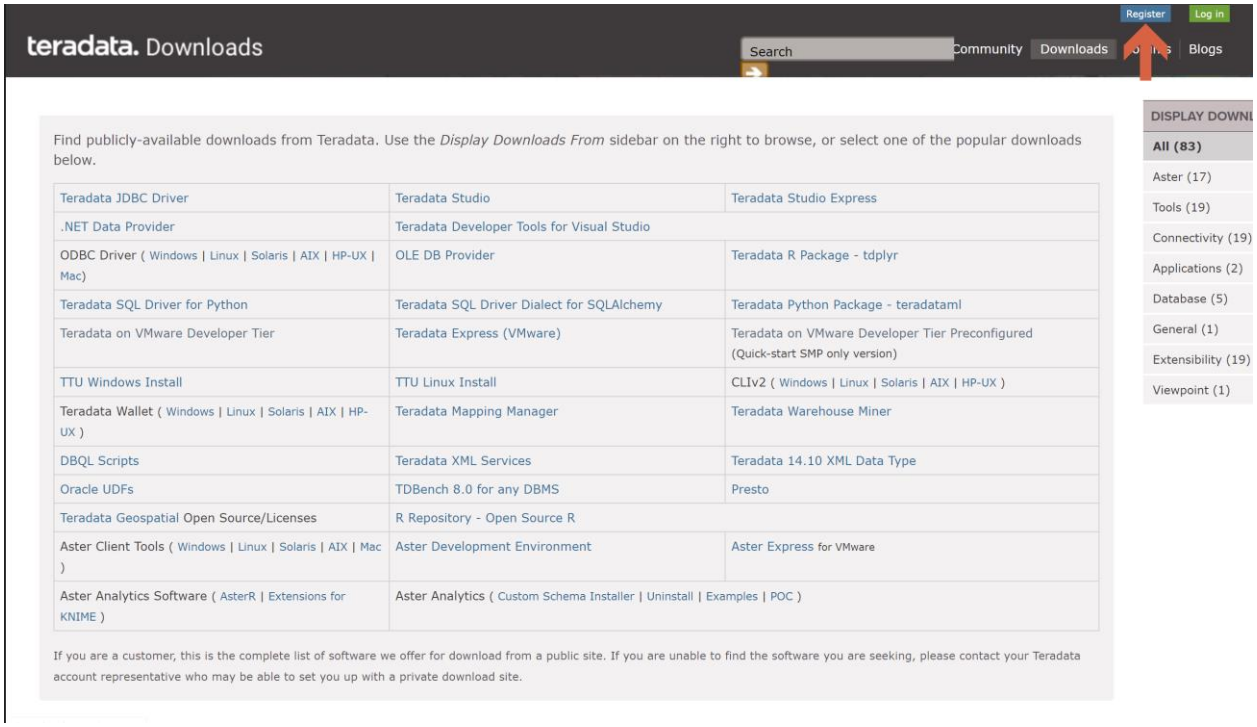
To illustrate this quick guide, we are going to use one of the virtual machines Teradata offers for free trials. These virtual machines can be found in Teradata Downloads. At the time of writing this document (October 2019), there are two Teradata Database flavours for trial. We are going to use the Teradata on VMware Developer Tier Preconfigured, released in May 2018.

2. Getting the Teradata VMware virtual machine

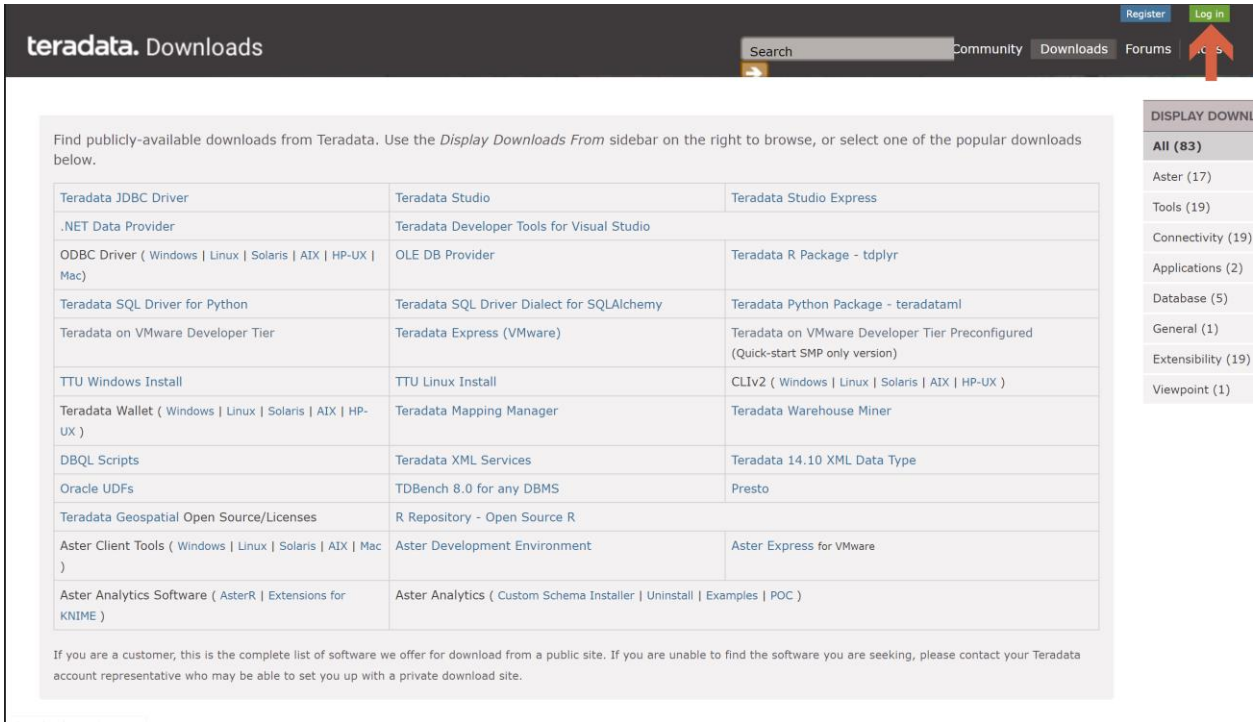
Before downloading the Teradata virtual machine, you'll have to log in Teradata Downloads. If you don't have a user for Teradata Community yet, register in it. In your laptop, open a browser and navigate to the Teradata Community.

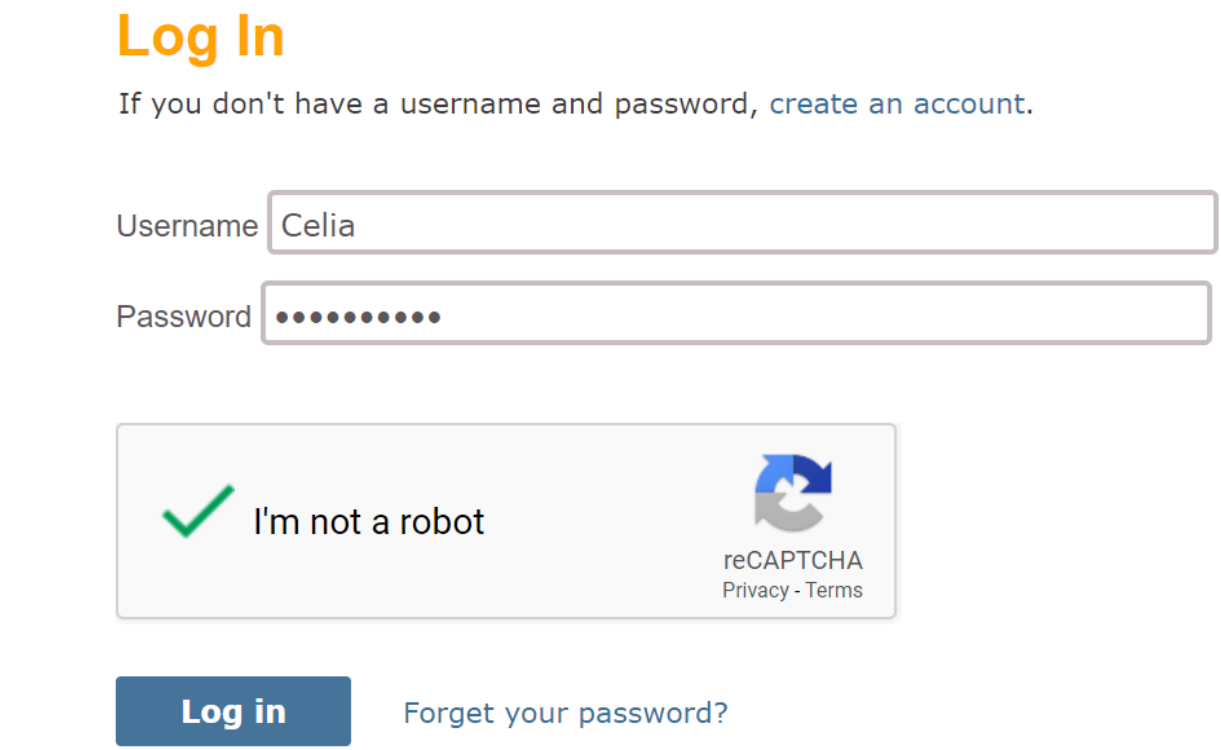


You can also register in Teradata Downloads.

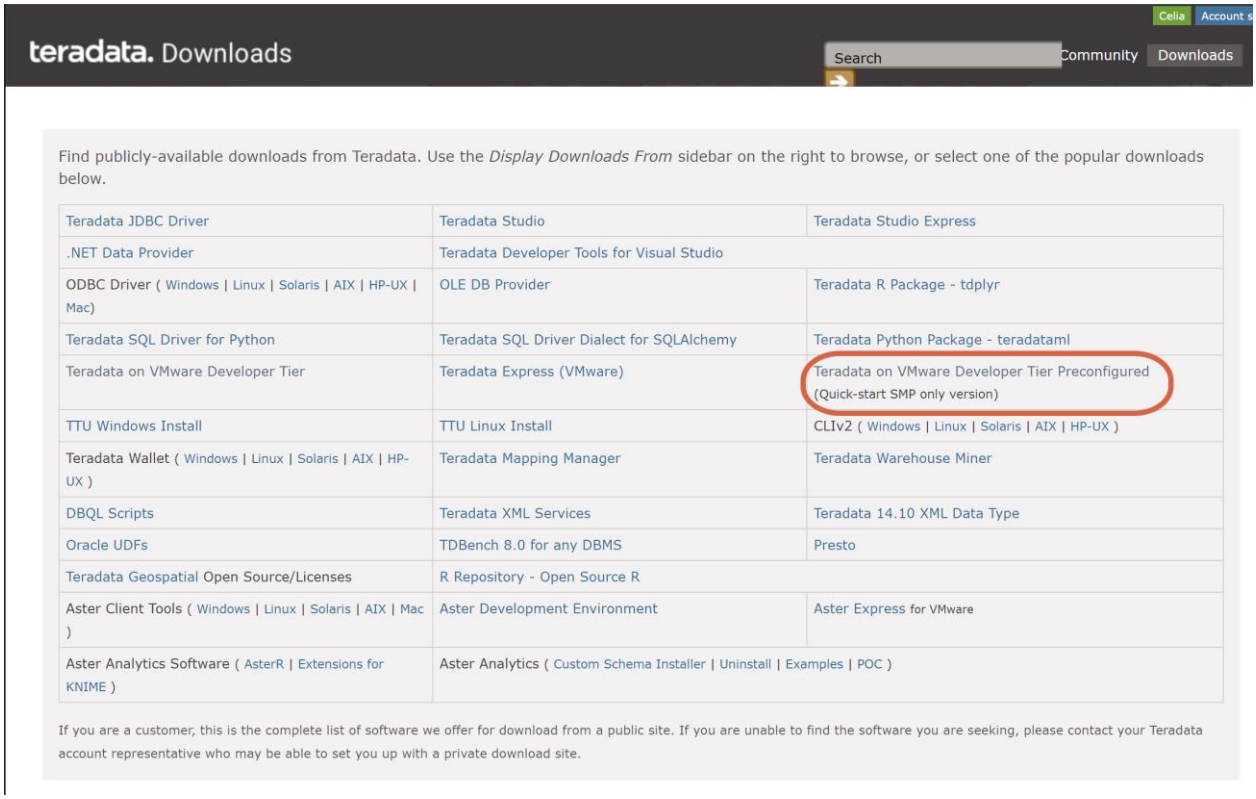


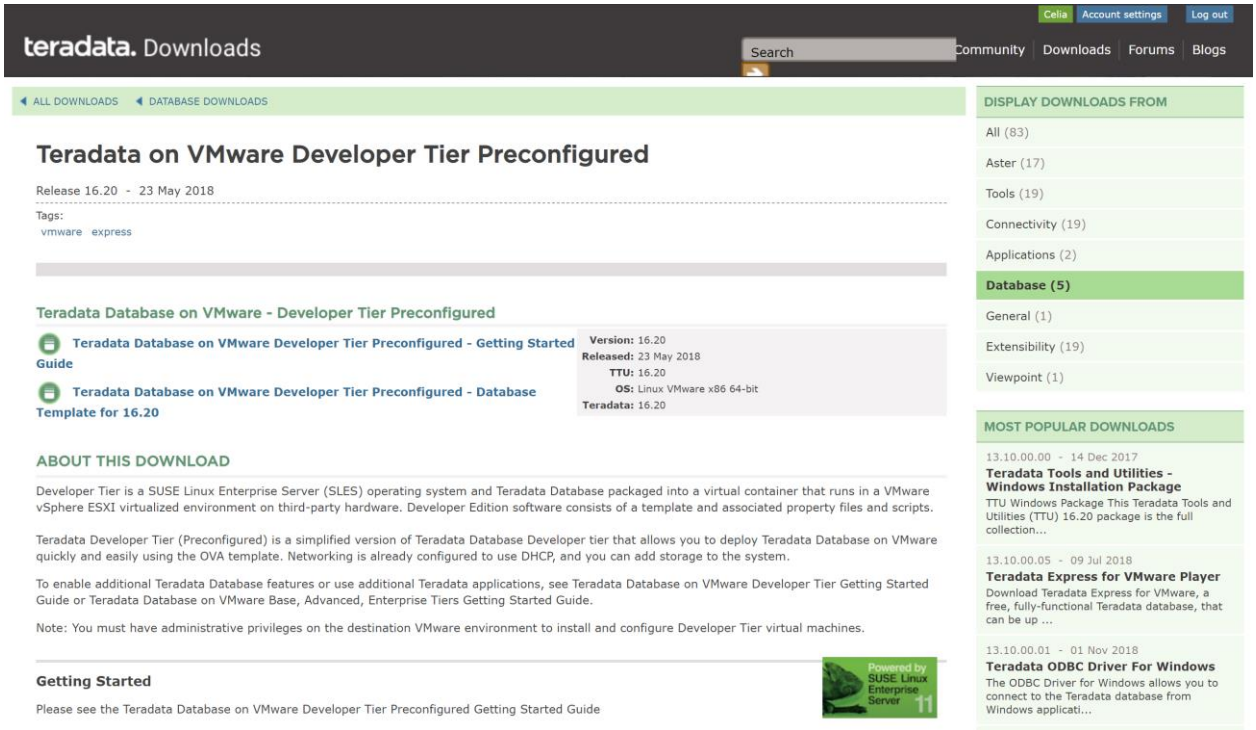
Once you have a user, log in [Teradata Downloads](#).



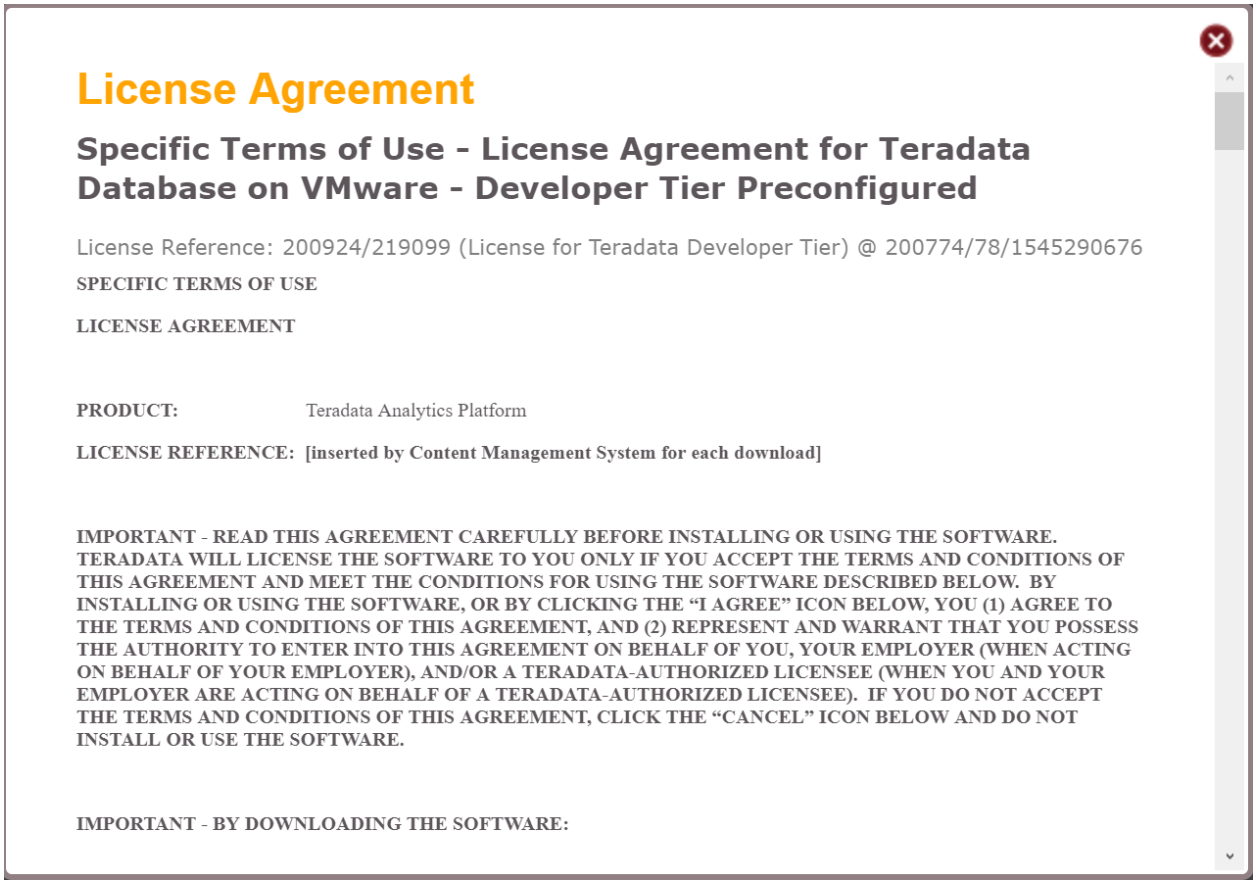


In Teradata Downloads, click on Teradata on VMware Developer Tier Preconfigured.





Click on the Getting Started Guide. Scroll down to the bottom of the License Agreement, and click on "I Agree".



License Agreement

Specific Terms of Use - License Agreement for Teradata Database on VMware - Developer Tier Preconfigured

License Reference: 200924/219099 (License for Teradata Developer Tier) @ 200774/78/1545290676

SPECIFIC TERMS OF USE

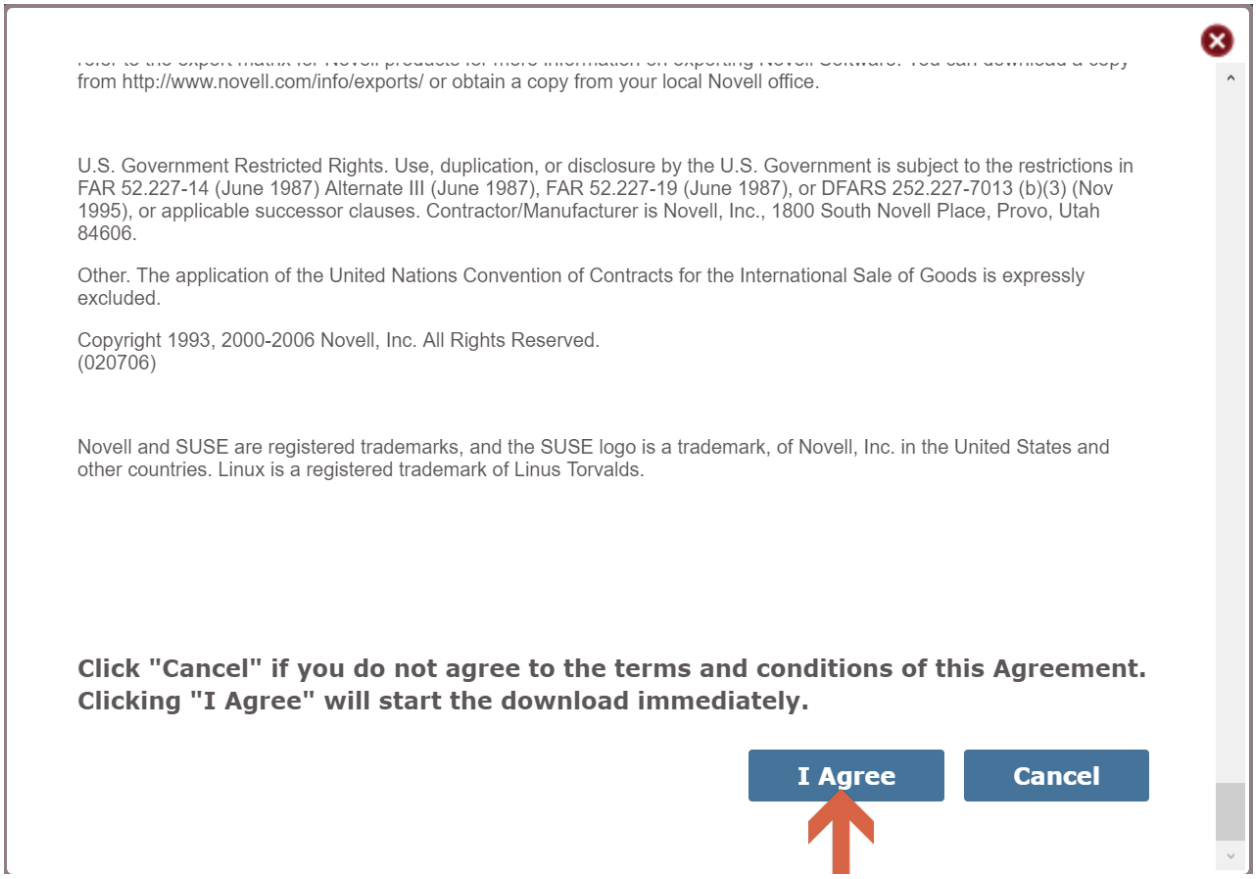
LICENSE AGREEMENT

PRODUCT: Teradata Analytics Platform

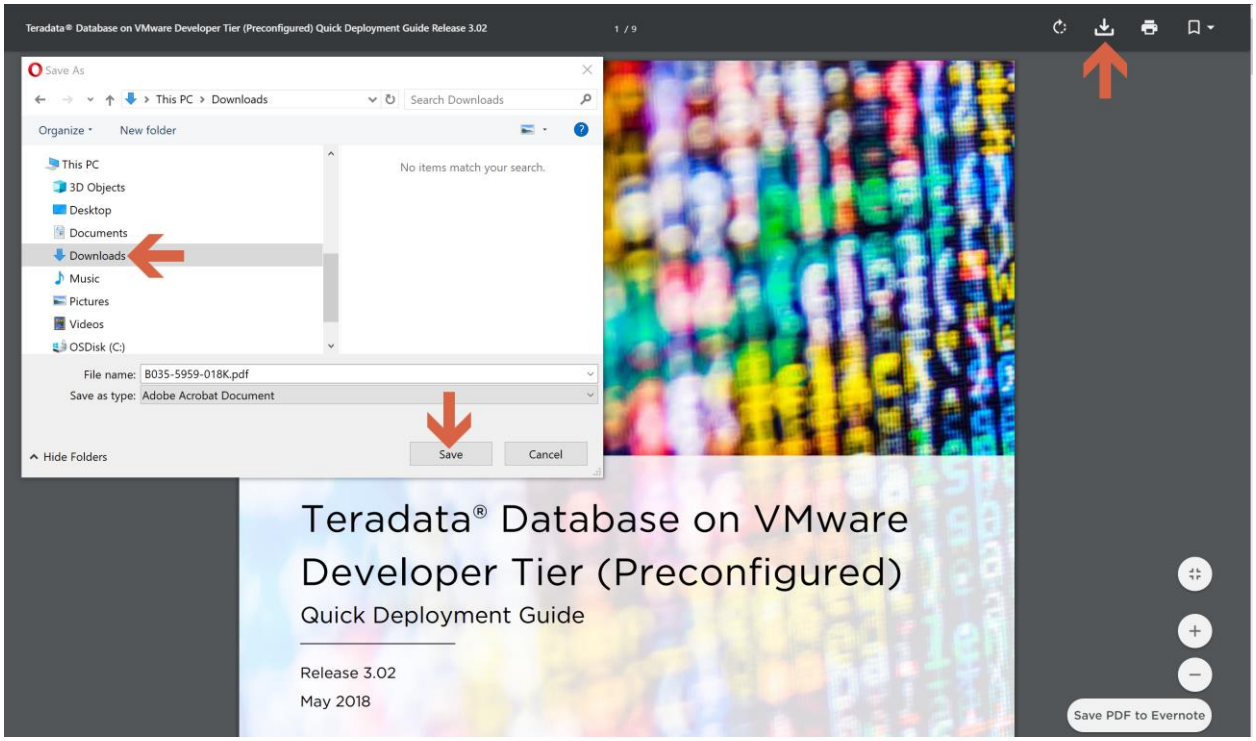
LICENSE REFERENCE: [inserted by Content Management System for each download]

IMPORTANT - READ THIS AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE SOFTWARE. TERADATA WILL LICENSE THE SOFTWARE TO YOU ONLY IF YOU ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT AND MEET THE CONDITIONS FOR USING THE SOFTWARE DESCRIBED BELOW. BY INSTALLING OR USING THE SOFTWARE, OR BY CLICKING THE "I AGREE" ICON BELOW, YOU (1) AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, AND (2) REPRESENT AND WARRANT THAT YOU POSSESS THE AUTHORITY TO ENTER INTO THIS AGREEMENT ON BEHALF OF YOU, YOUR EMPLOYER (WHEN ACTING ON BEHALF OF YOUR EMPLOYER), AND/OR A TERADATA-AUTHORIZED LICENSEE (WHEN YOU AND YOUR EMPLOYER ARE ACTING ON BEHALF OF A TERADATA-AUTHORIZED LICENSEE). IF YOU DO NOT ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT, CLICK THE "CANCEL" ICON BELOW AND DO NOT INSTALL OR USE THE SOFTWARE.

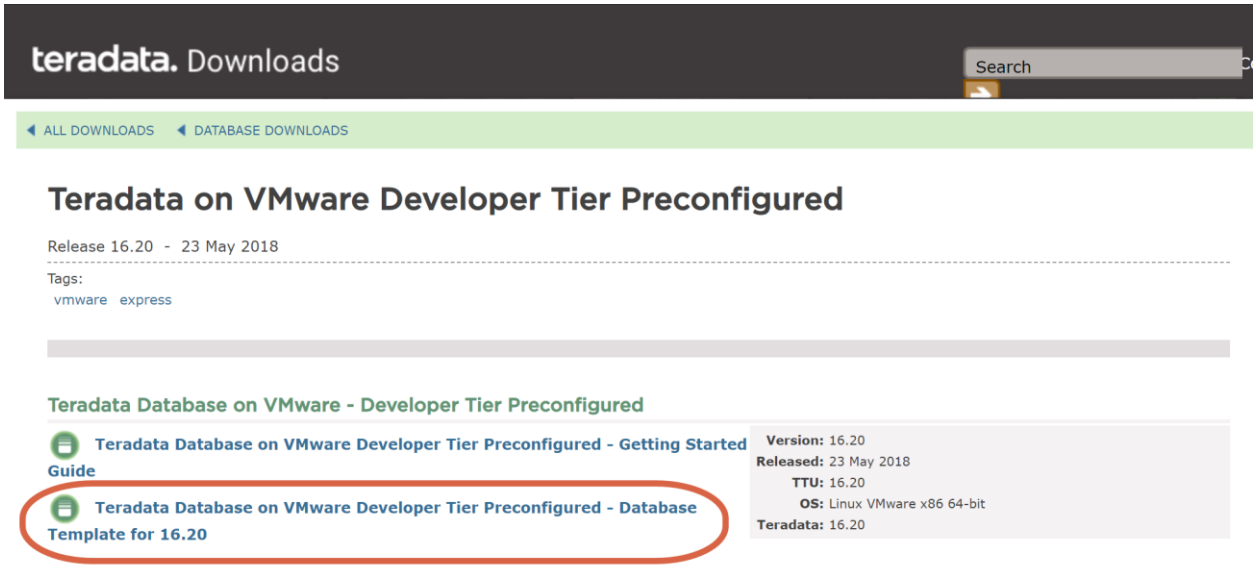
IMPORTANT - BY DOWNLOADING THE SOFTWARE:



Save the file to a location on your computer.



Back on the [Teradata on VMware Developer Tier Preconfigured] page, click on the Database Template.



Scroll down to the bottom of the License Agreement, and click on “I Agree”.

License Agreement

Specific Terms of Use - License Agreement for Teradata Database on VMware - Developer Tier Preconfigured

License Reference: 200924/219099 (License for Teradata Developer Tier) @ 200774/78/1545290987

SPECIFIC TERMS OF USE

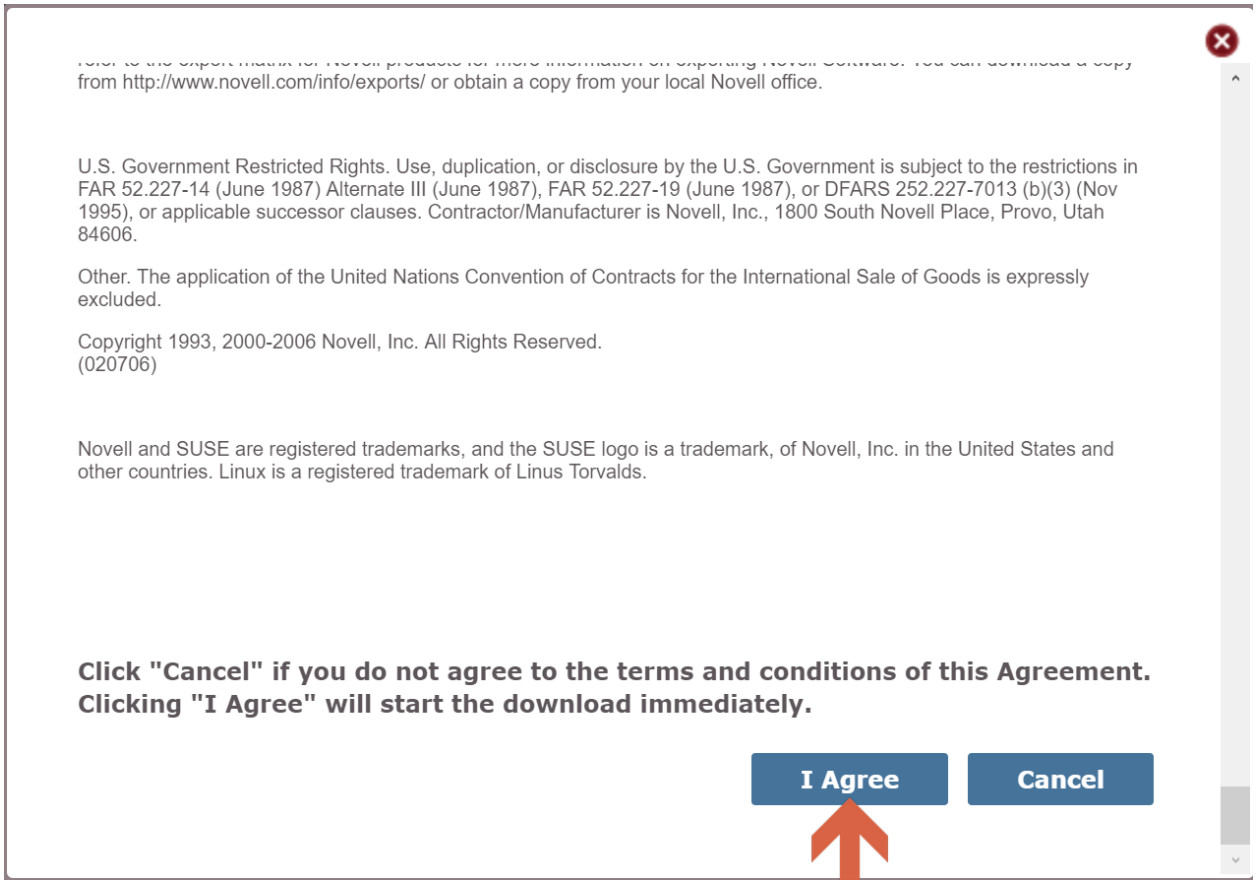
LICENSE AGREEMENT

PRODUCT: Teradata Analytics Platform

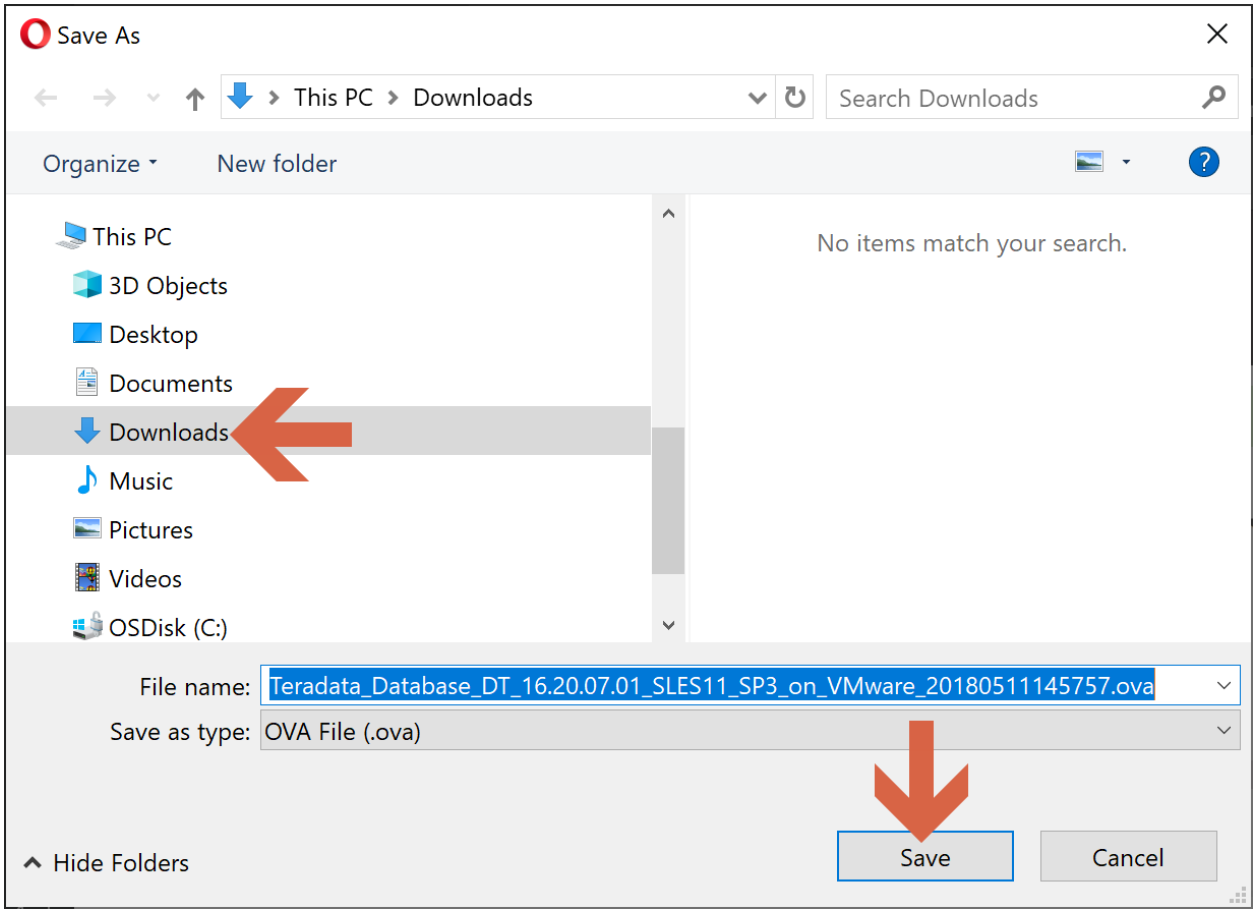
LICENSE REFERENCE: [inserted by Content Management System for each download]

IMPORTANT - READ THIS AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE SOFTWARE. TERADATA WILL LICENSE THE SOFTWARE TO YOU ONLY IF YOU ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT AND MEET THE CONDITIONS FOR USING THE SOFTWARE DESCRIBED BELOW. BY INSTALLING OR USING THE SOFTWARE, OR BY CLICKING THE “I AGREE” ICON BELOW, YOU (1) AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, AND (2) REPRESENT AND WARRANT THAT YOU POSSESS THE AUTHORITY TO ENTER INTO THIS AGREEMENT ON BEHALF OF YOU, YOUR EMPLOYER (WHEN ACTING ON BEHALF OF YOUR EMPLOYER), AND/OR A TERADATA-AUTHORIZED LICENSEE (WHEN YOU AND YOUR EMPLOYER ARE ACTING ON BEHALF OF A TERADATA-AUTHORIZED LICENSEE). IF YOU DO NOT ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT, CLICK THE “CANCEL” ICON BELOW AND DO NOT INSTALL OR USE THE SOFTWARE.

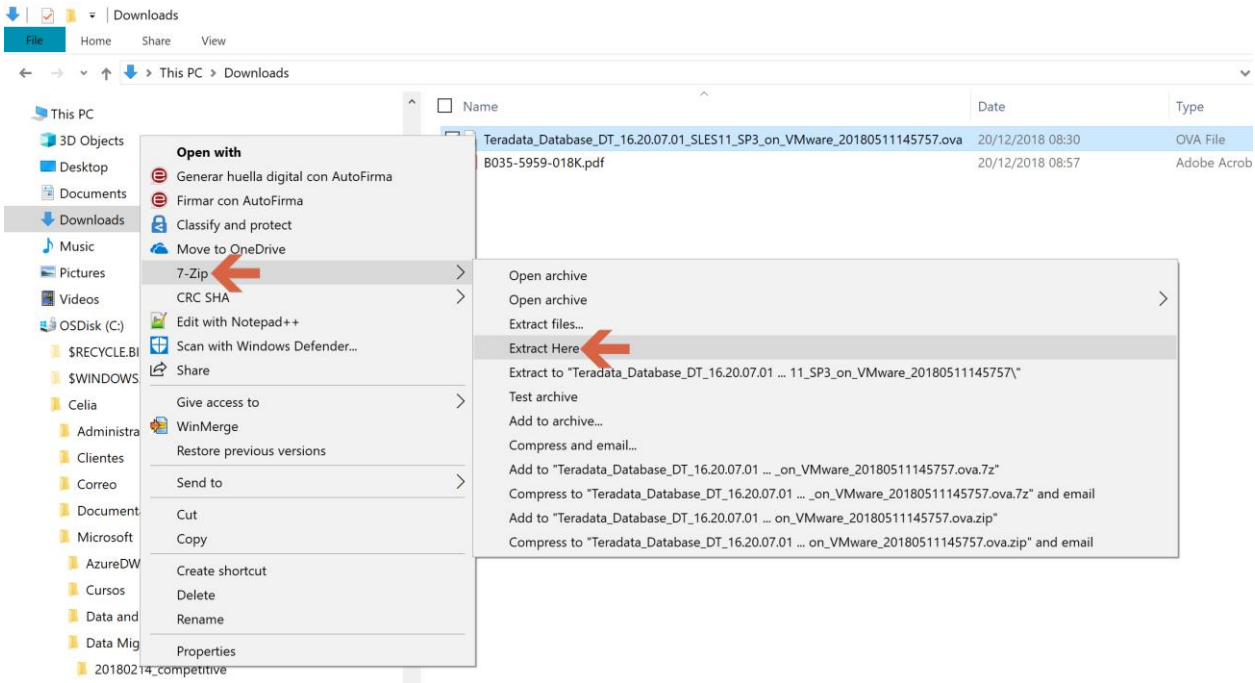
IMPORTANT - BY DOWNLOADING THE SOFTWARE:

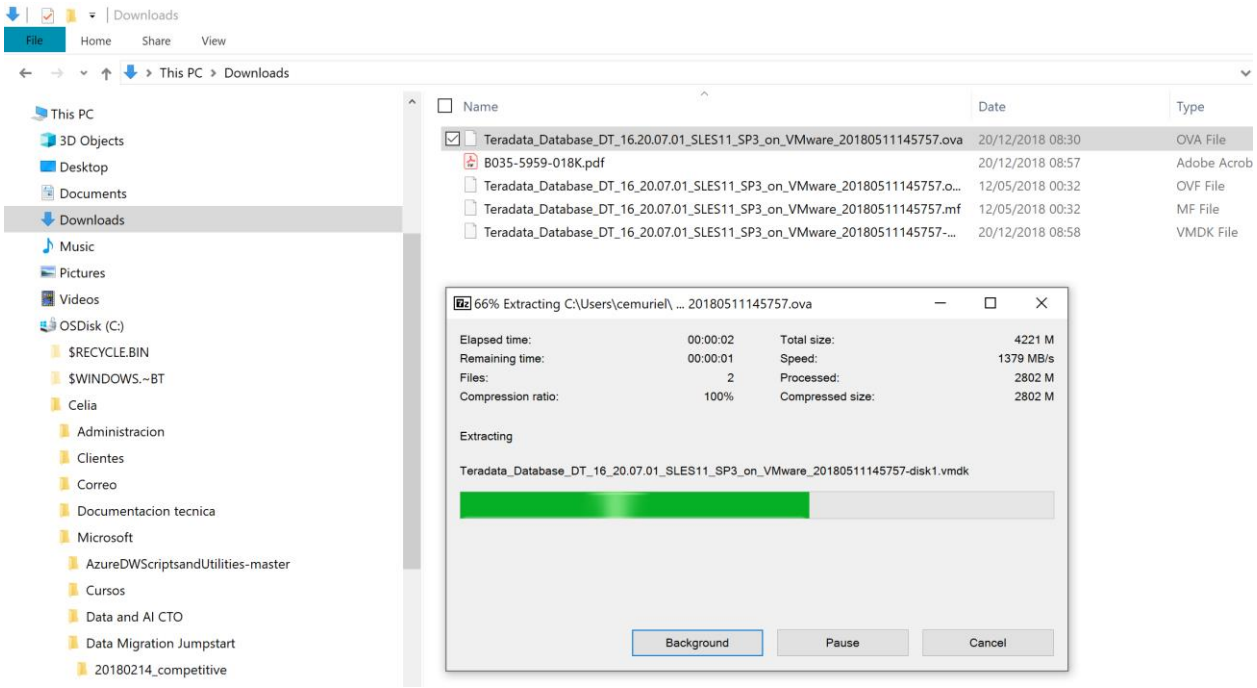


Save the file to a directory on your computer.

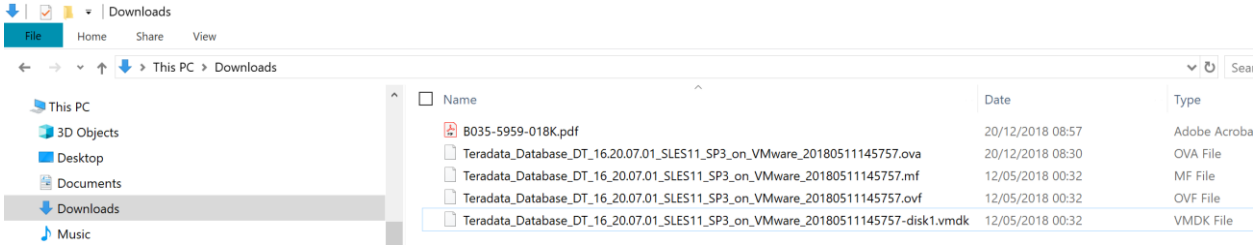


In our laptop, we go to the directory where we downloaded the OVA file, and we extract it. We use 7z to unzip.





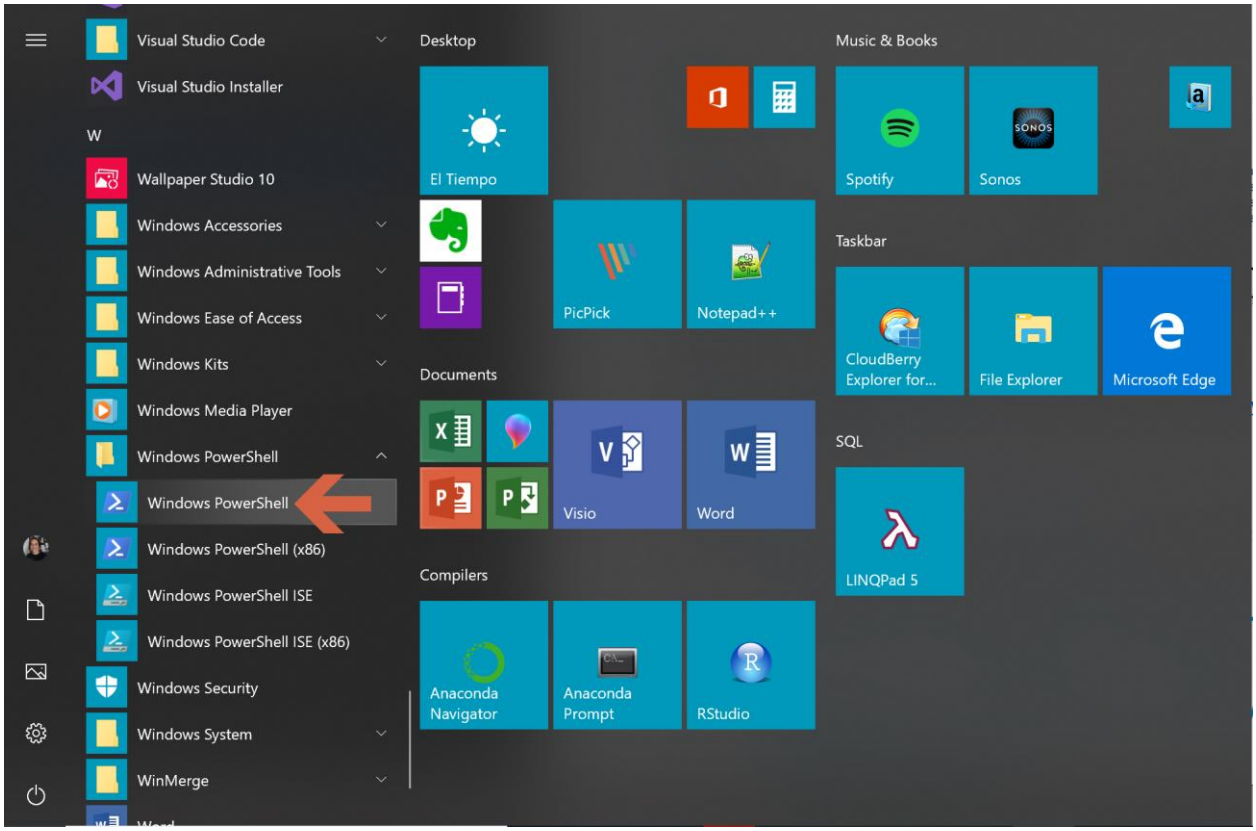
After uncompressing the OVA file, your directory should look like this.



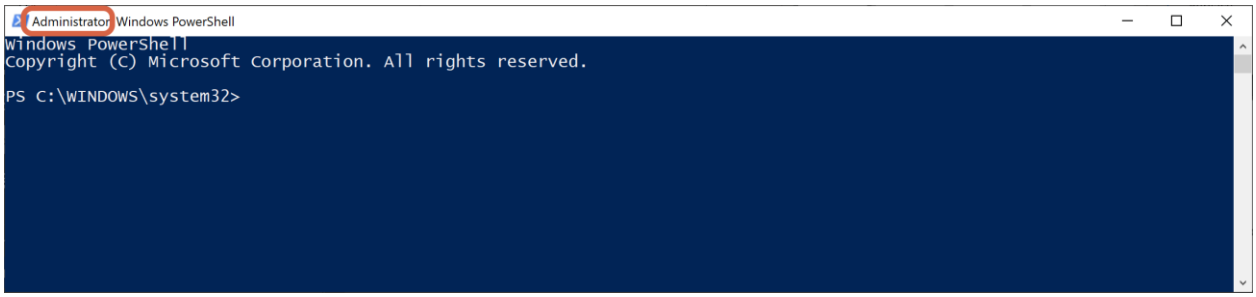
3. Converting the Teradata virtual machine to Hyper-V

To turn the image to Hyper-V format, we are going to use Microsoft Virtual Machine Converter. If you don't have it yet, download it and install it.

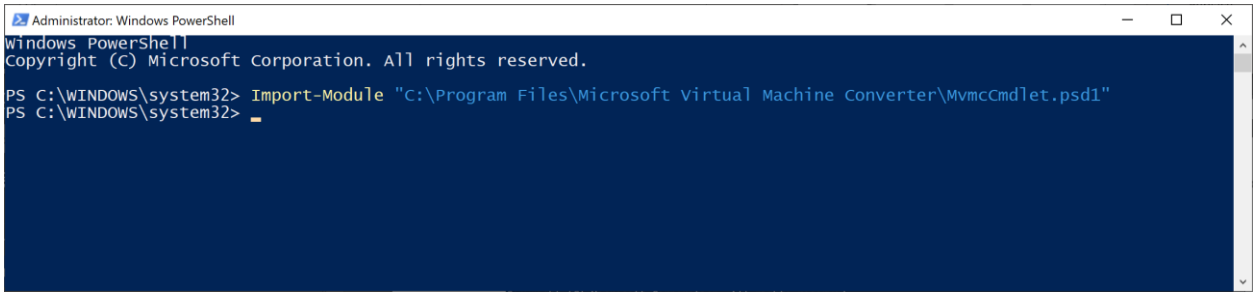
Once Microsoft Virtual Machine Converter is installed, we open PowerShell.



We are going to need Administrator privileges to convert the VMDK image, so we should open PowerShell as an administrator, or we run next command: *start-process powershell –verb runAs*

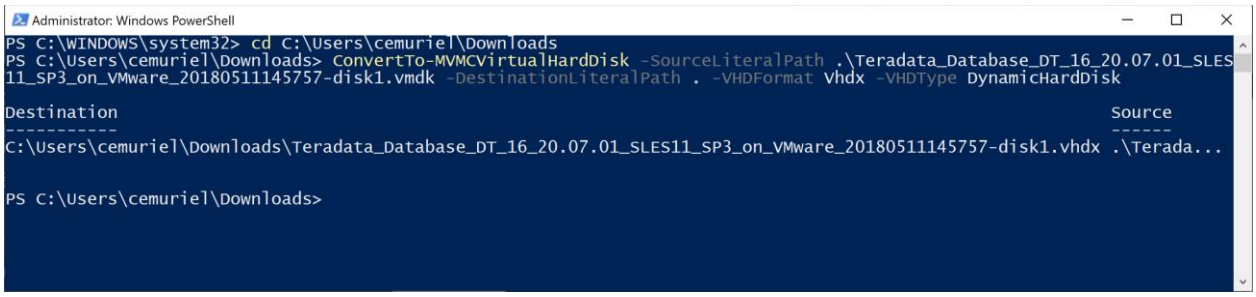


Now we import Microsoft Virtual Machine Converter into PowerShell: *Import-Module "C:\Program Files\Microsoft Virtual Machine Converter\MvmcCmdlet.psd1"*

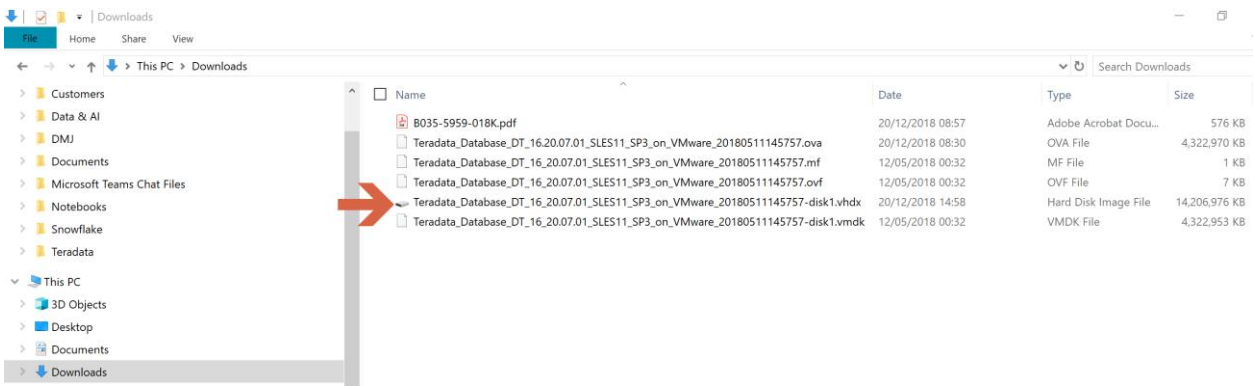


We can finally convert the VMDK image into the VHDX one. We are going to execute the command from the directory where the image is.

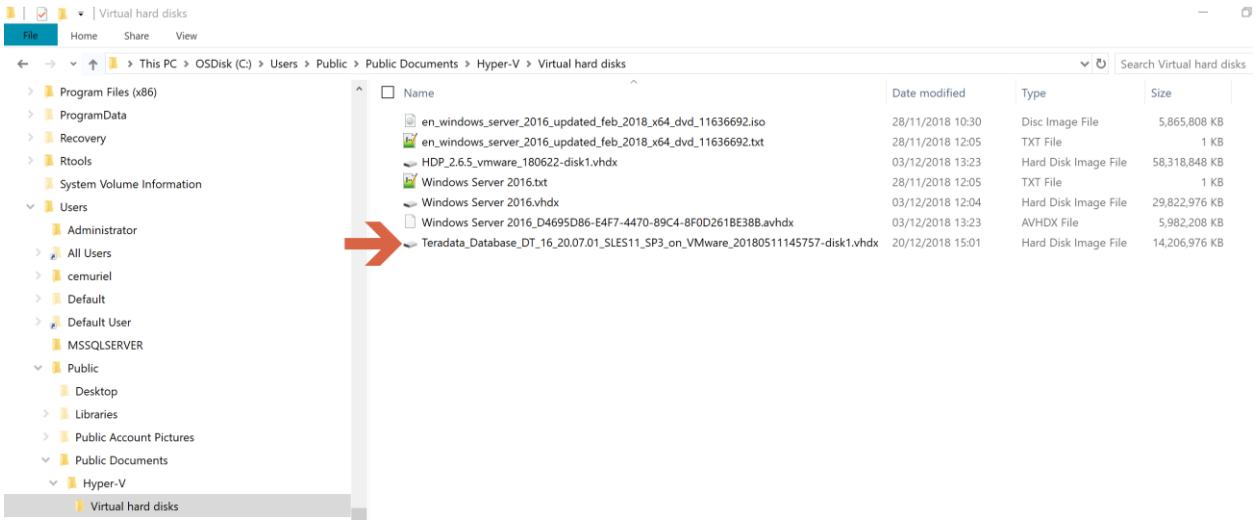
ConvertTo-MVMCVirtualHardDisk -SourceLiteralPath .\Teradata_Database_DT_16_20.07.01_SLES11_SP3_on_VMware_20180511145757-disk1.vmdk -DestinationLiteralPath . -VHDFormat Vhdx -VHDType DynamicHardDisk



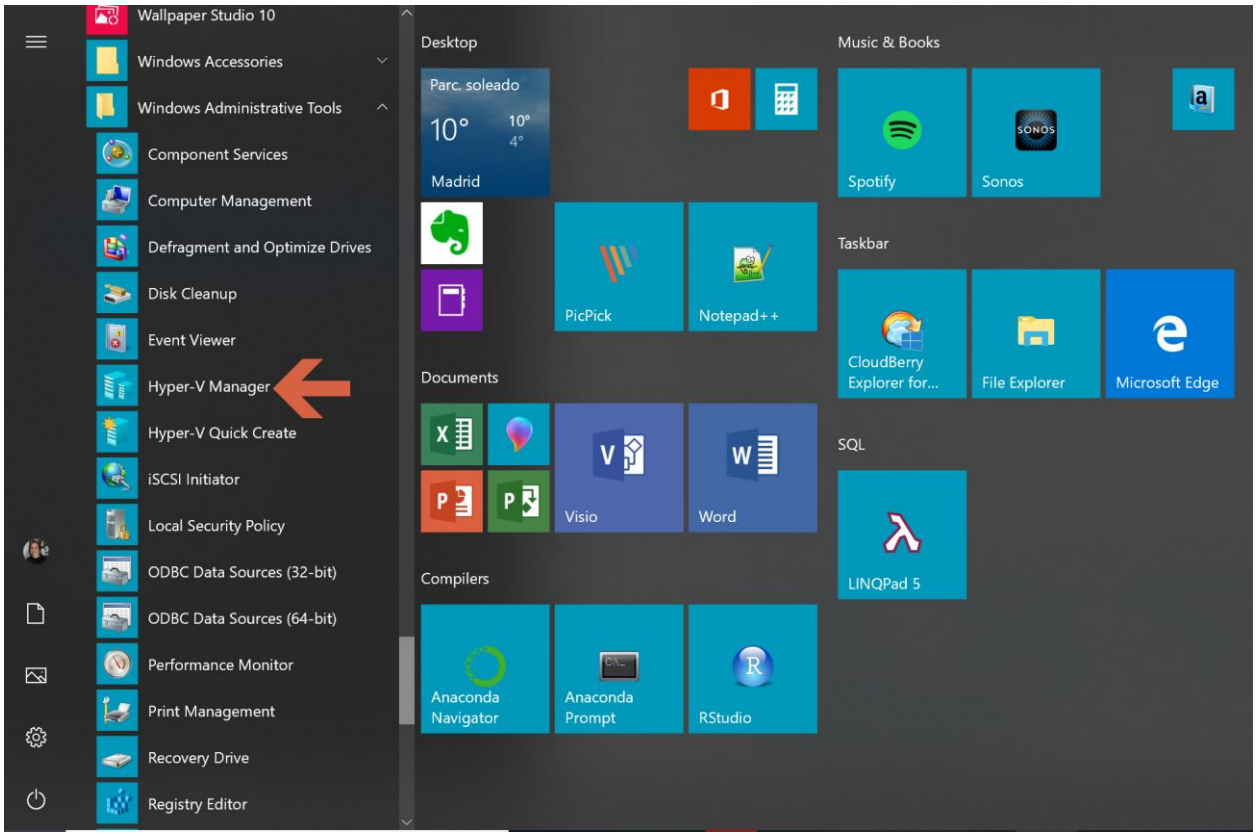
After the conversion, we have the VHDX image in the same directory with the VMDK image.



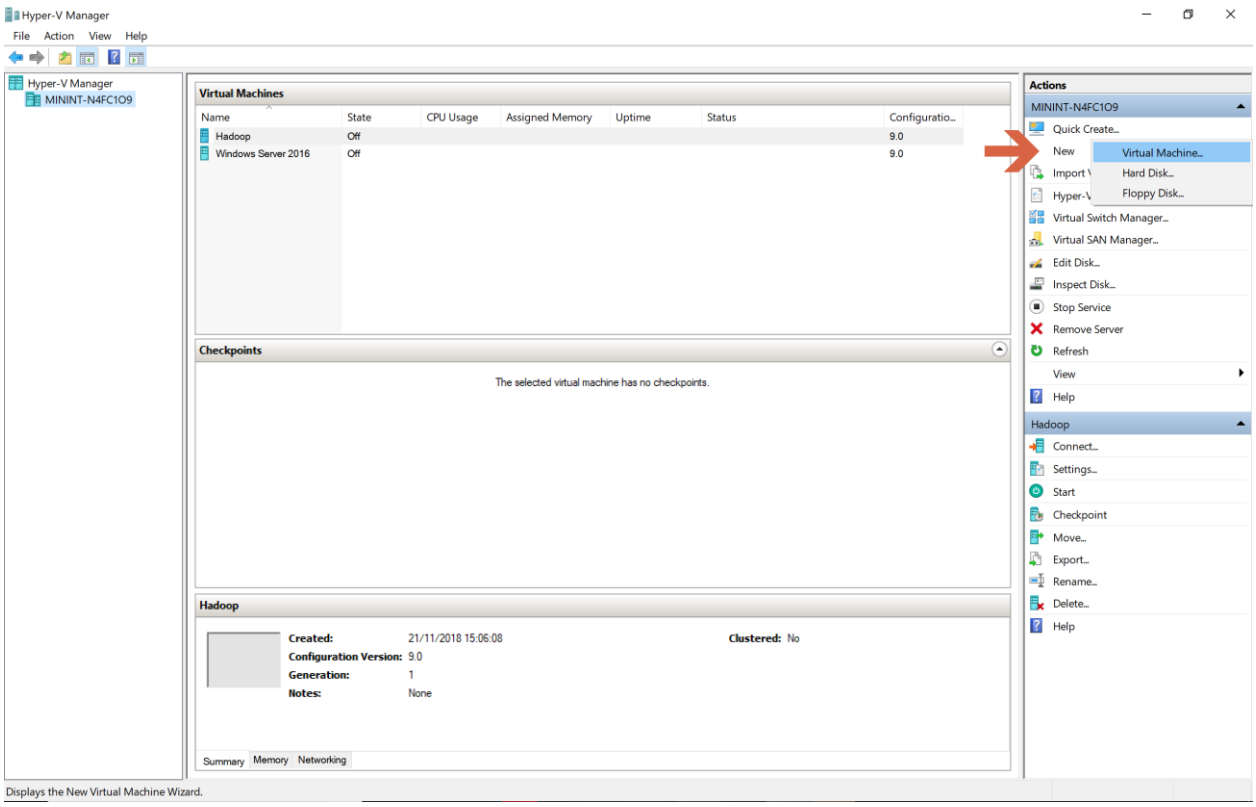
We are going to use the VHDX image to create a Hyper-V virtual machine. Hyper-V's default directory is `C:\Users\Public\Documents\Hyper-V\Virtual hard disks`. We are going to place our VHDX image in this directory.



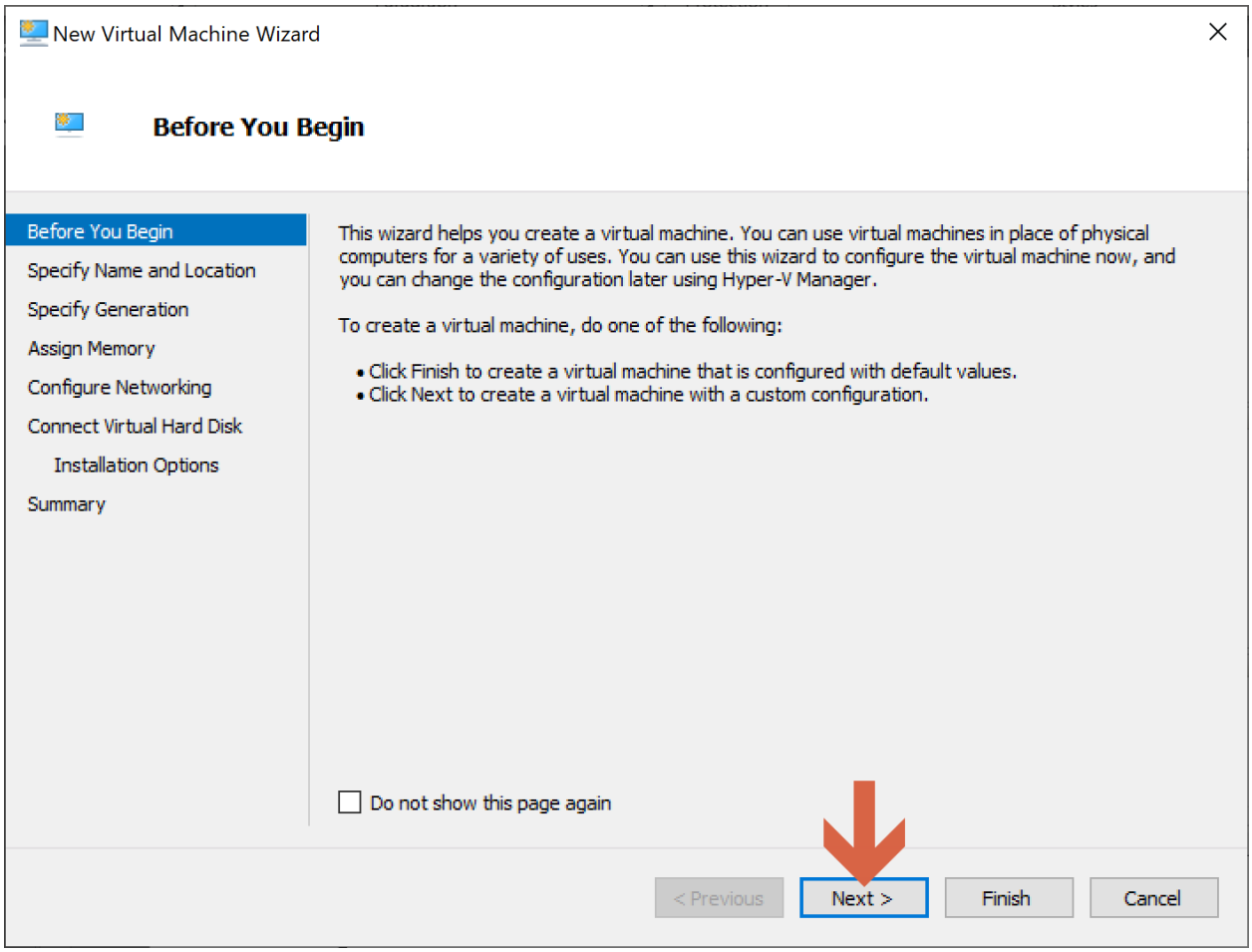
We open the Hyper-V Manager.



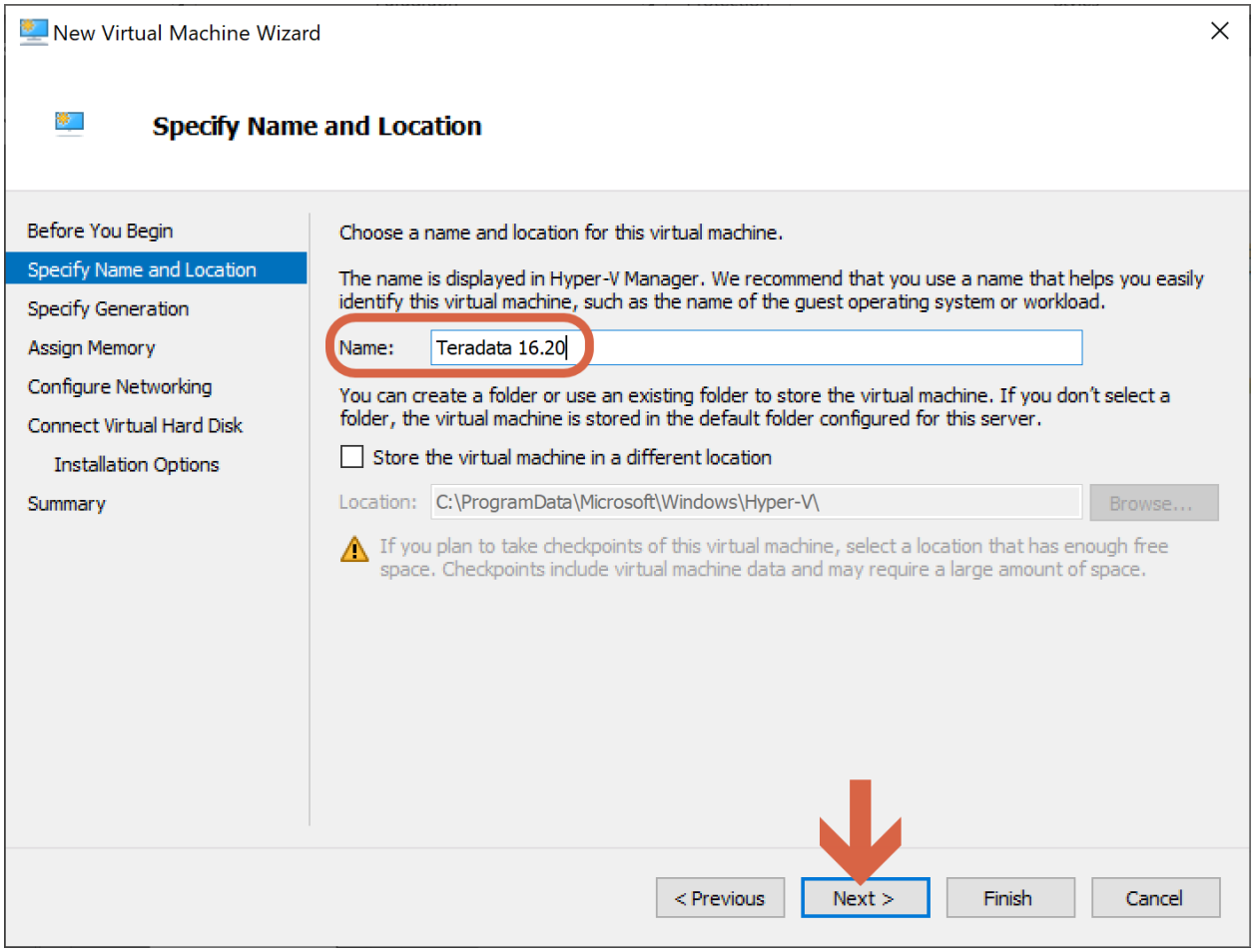
We create a new virtual machine.



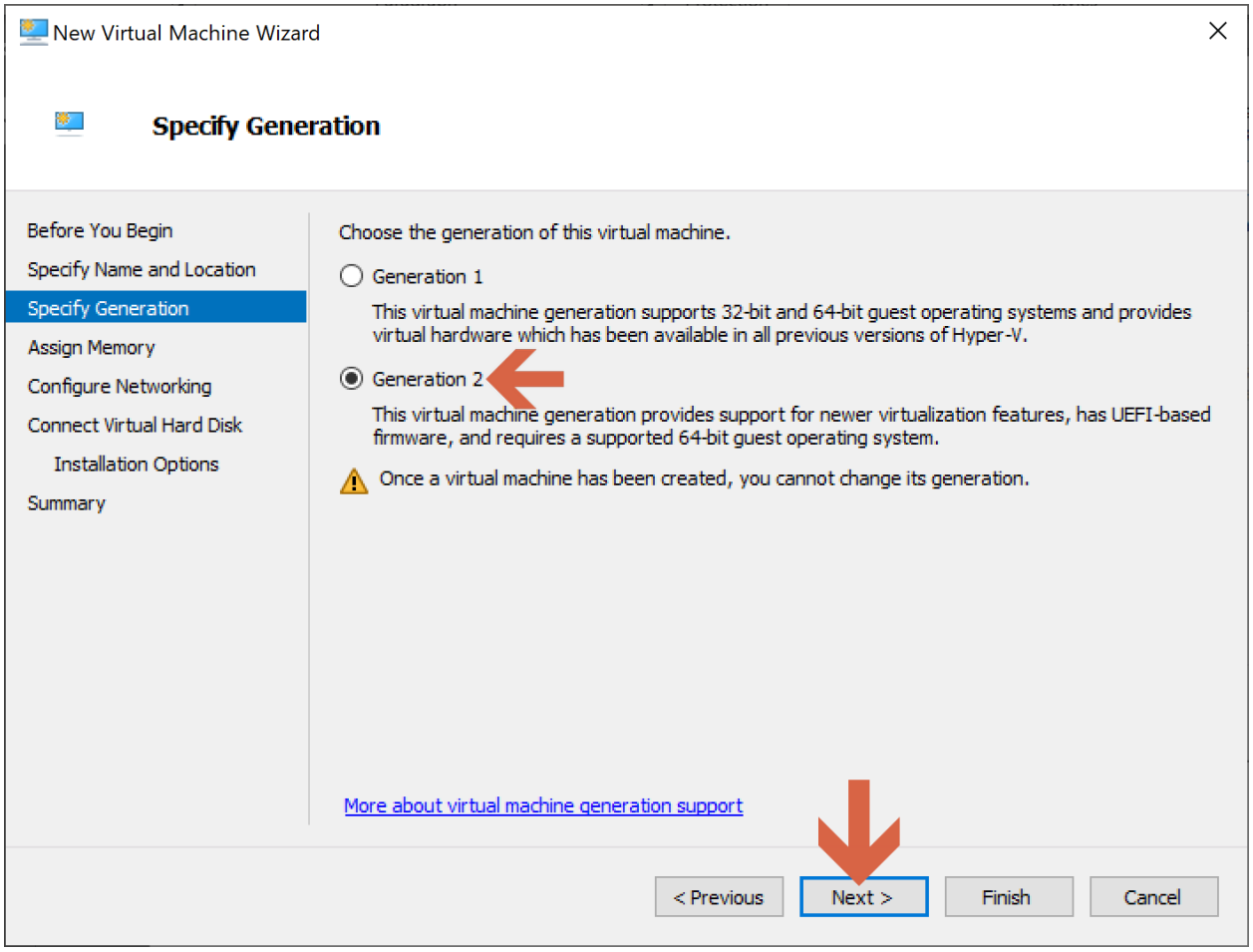
Click “Next” on the “Before you Begin” window.



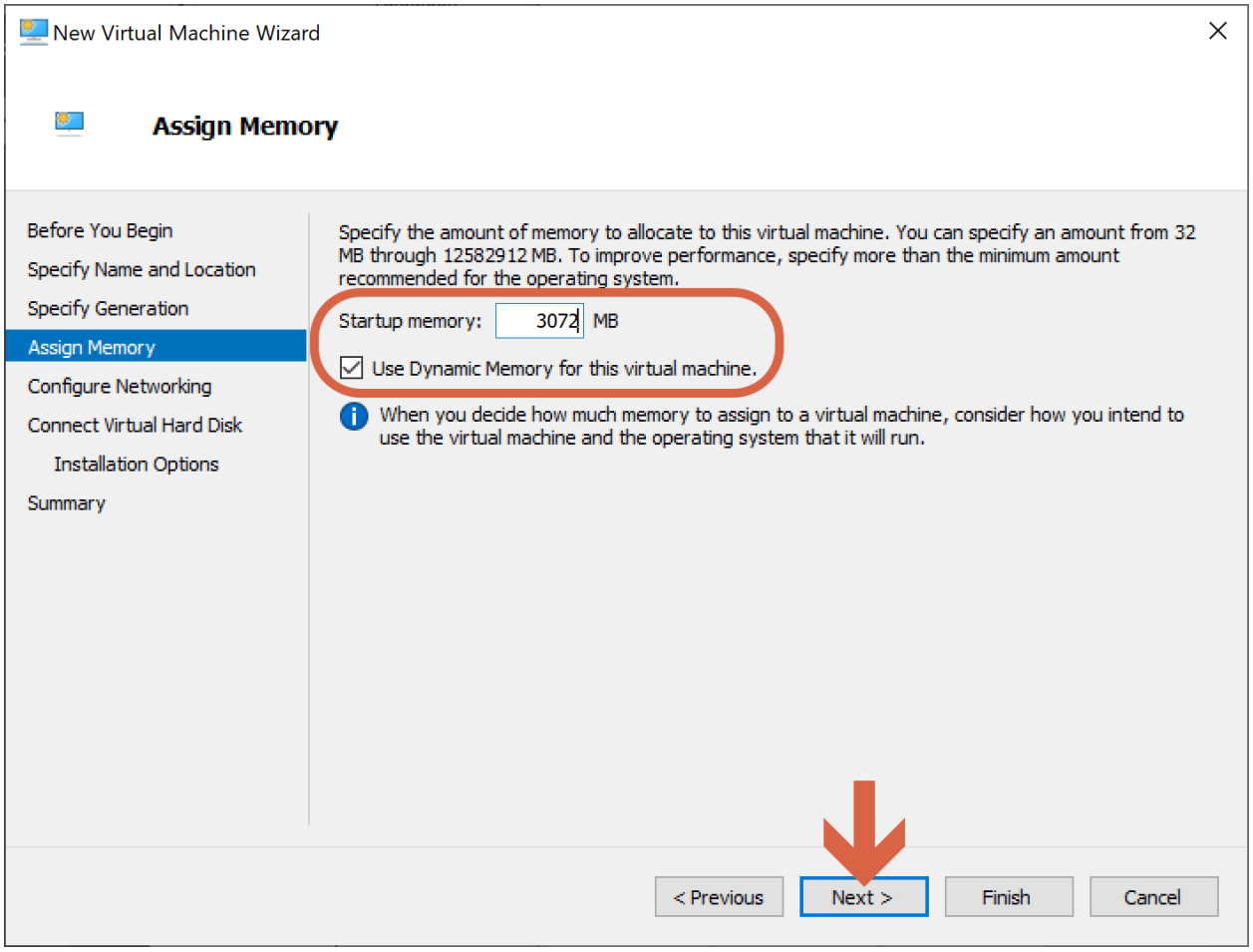
Name the new Hyper-V virtual machine.



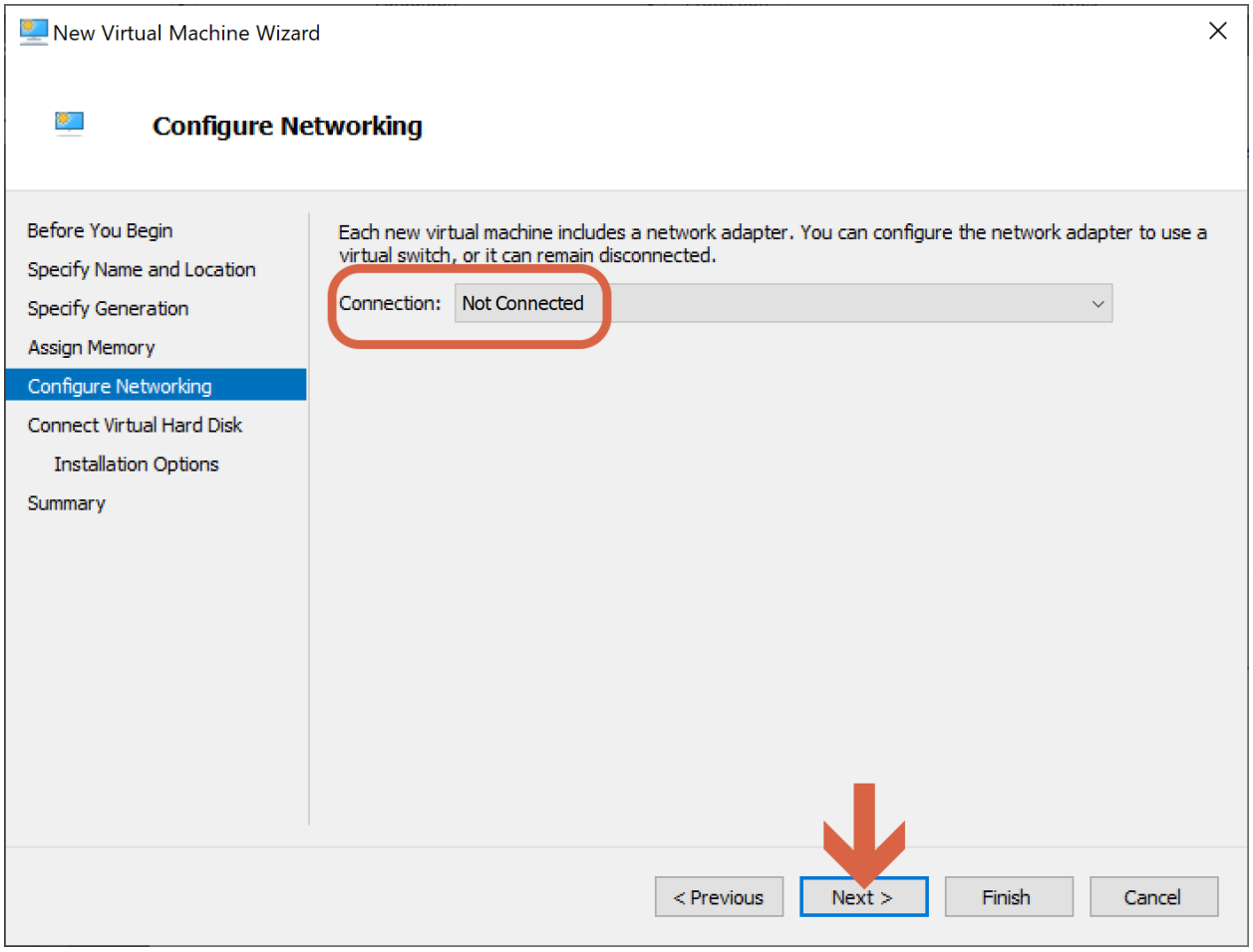
We choose the virtual machine generation.



We choose the memory for the virtual machine according to our PC's capabilities.

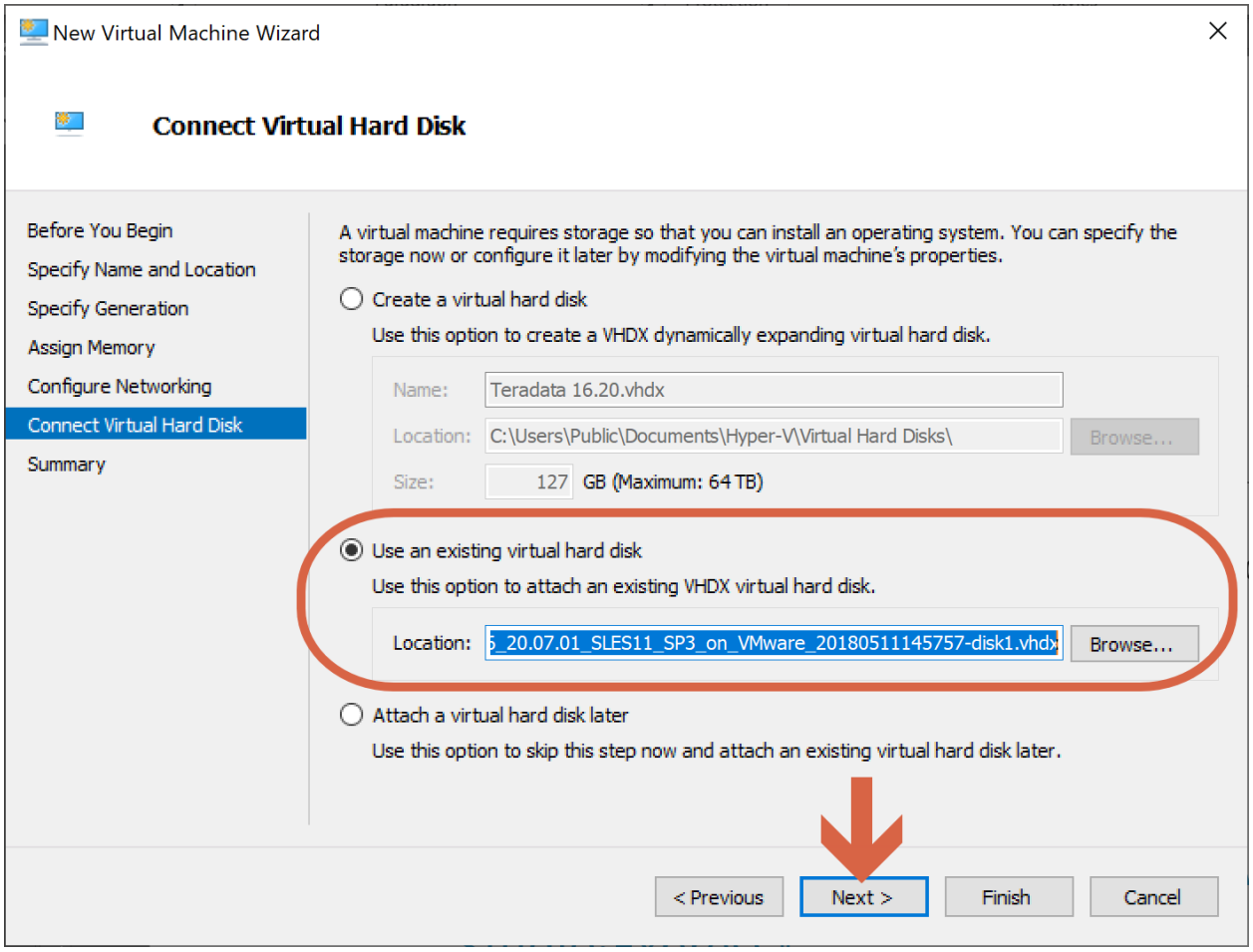


We will set up the network connection after the virtual machine has been created.

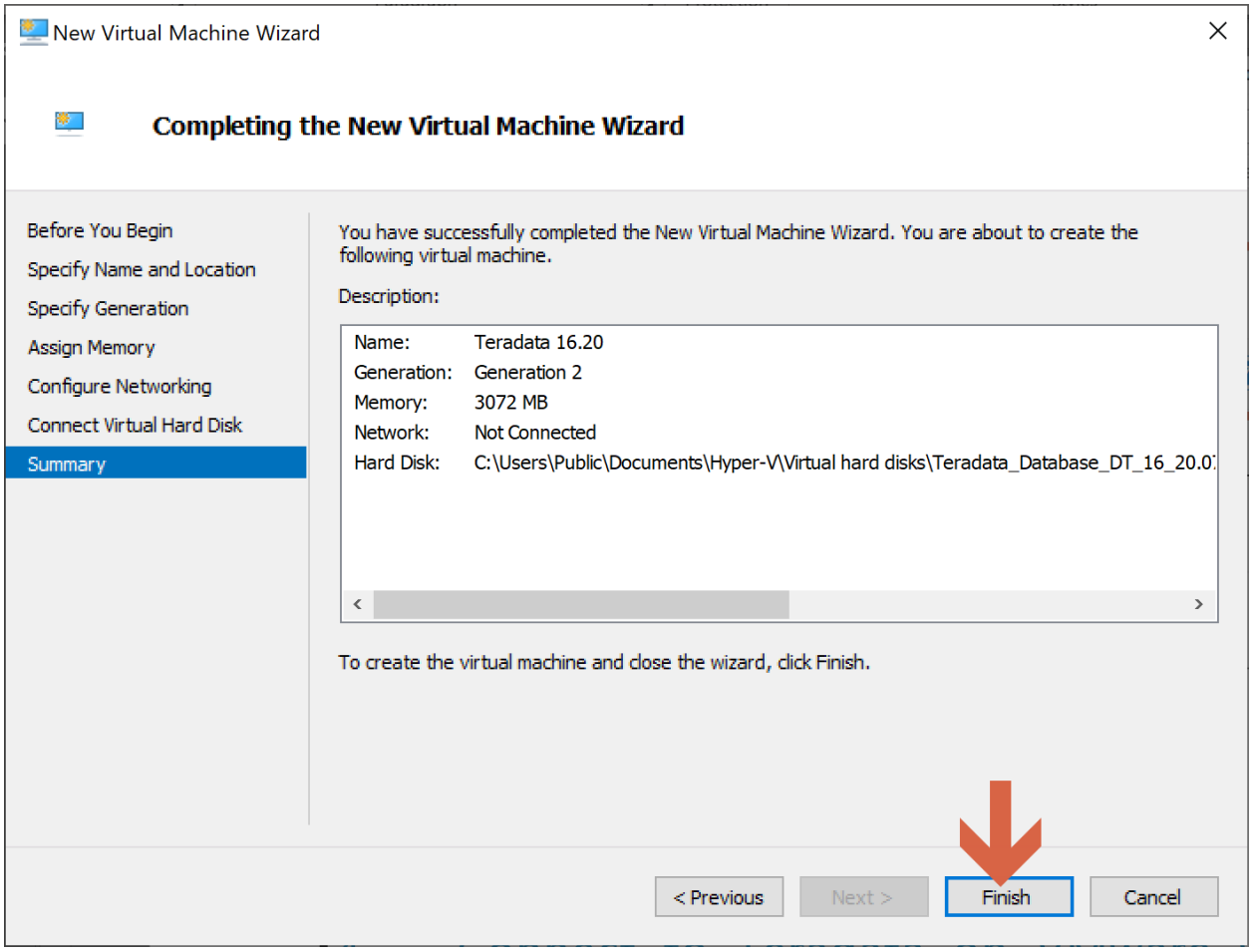


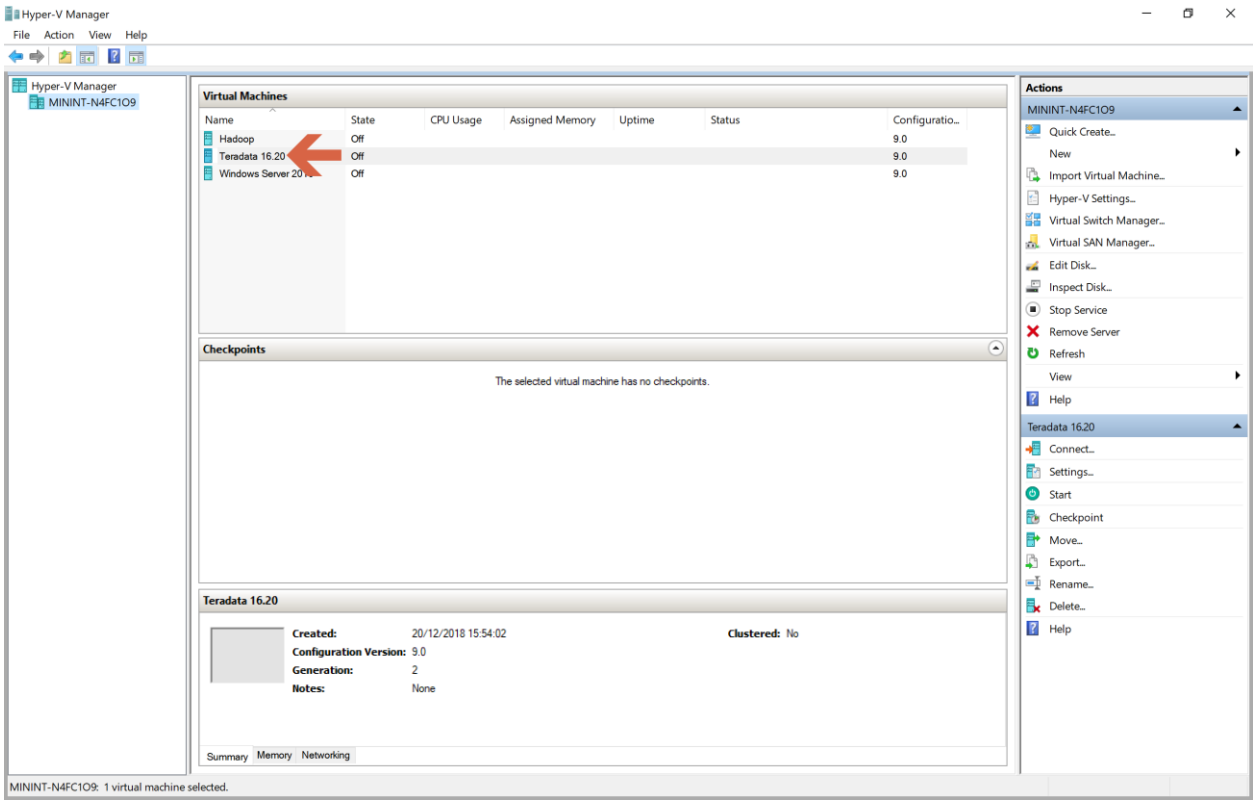
We connect this new virtual machine with the VHDX image we have just created and copied in the default Hyper-V directory.

```
C:\Users\Public\Documents\Hyper-V\Virtual hard
disks\Teradata_Database_DT_16_20.07.01_SLES11_SP3_on_VMware_20180511
145757-disk1.vhdx
```

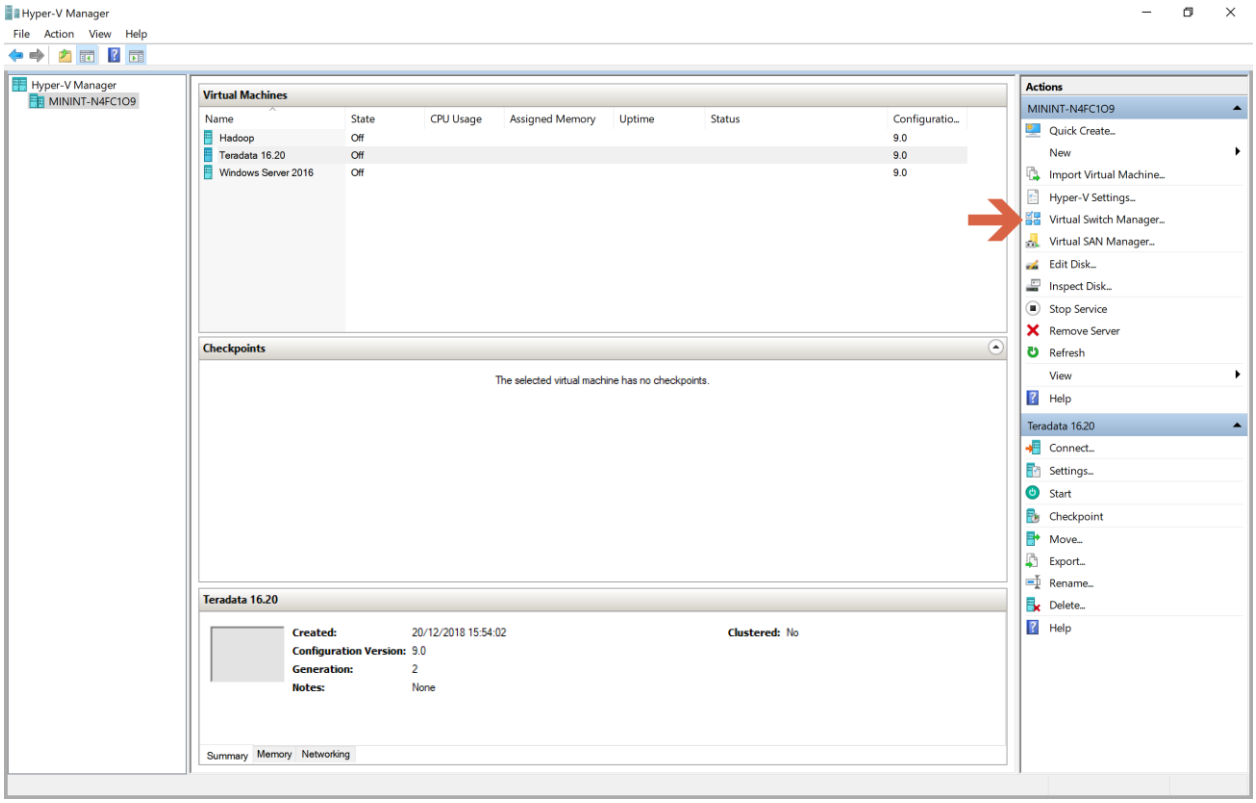


We click on “Finish” and wait to have our new Teradata 16.20 virtual machine ready.

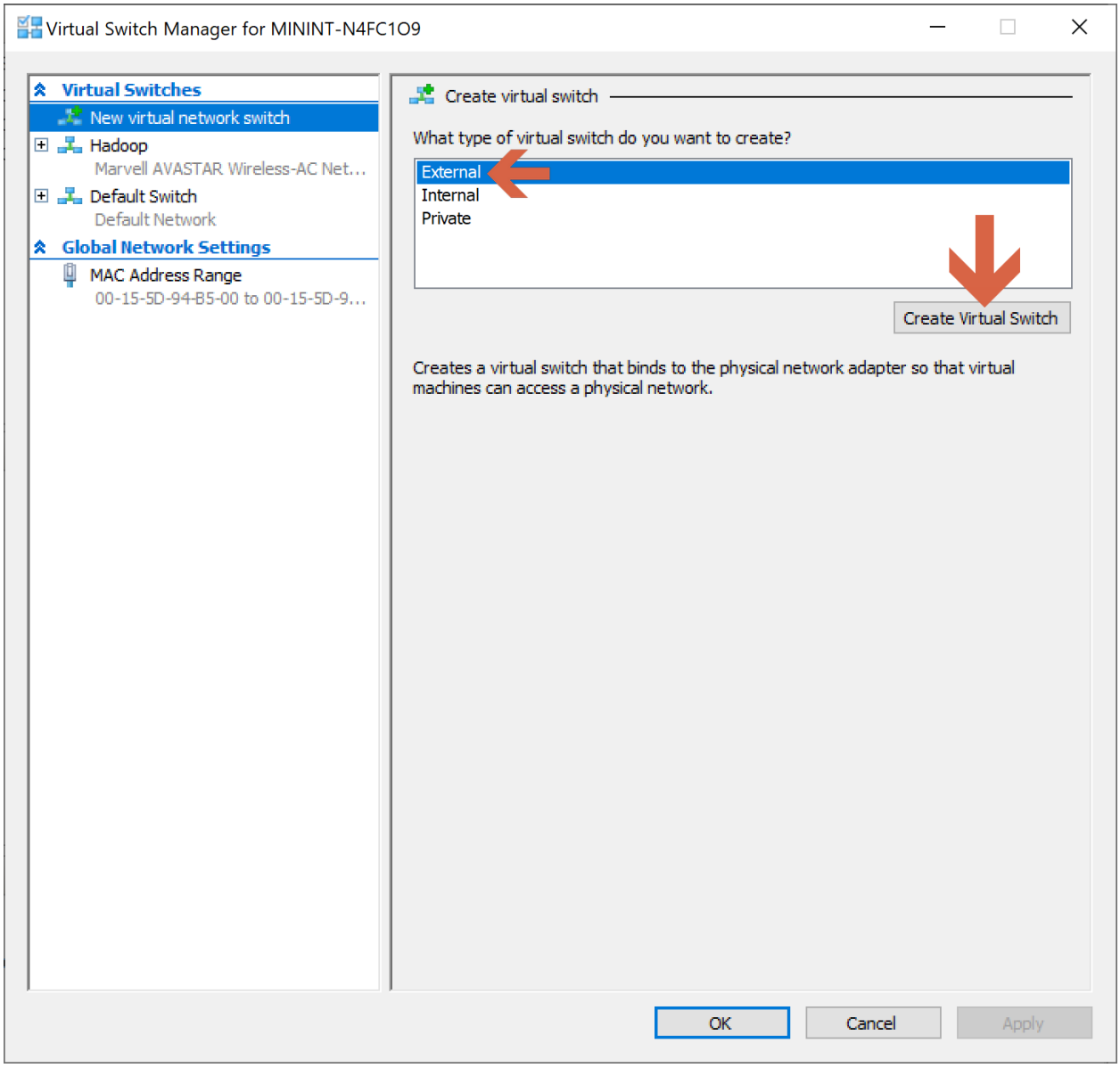




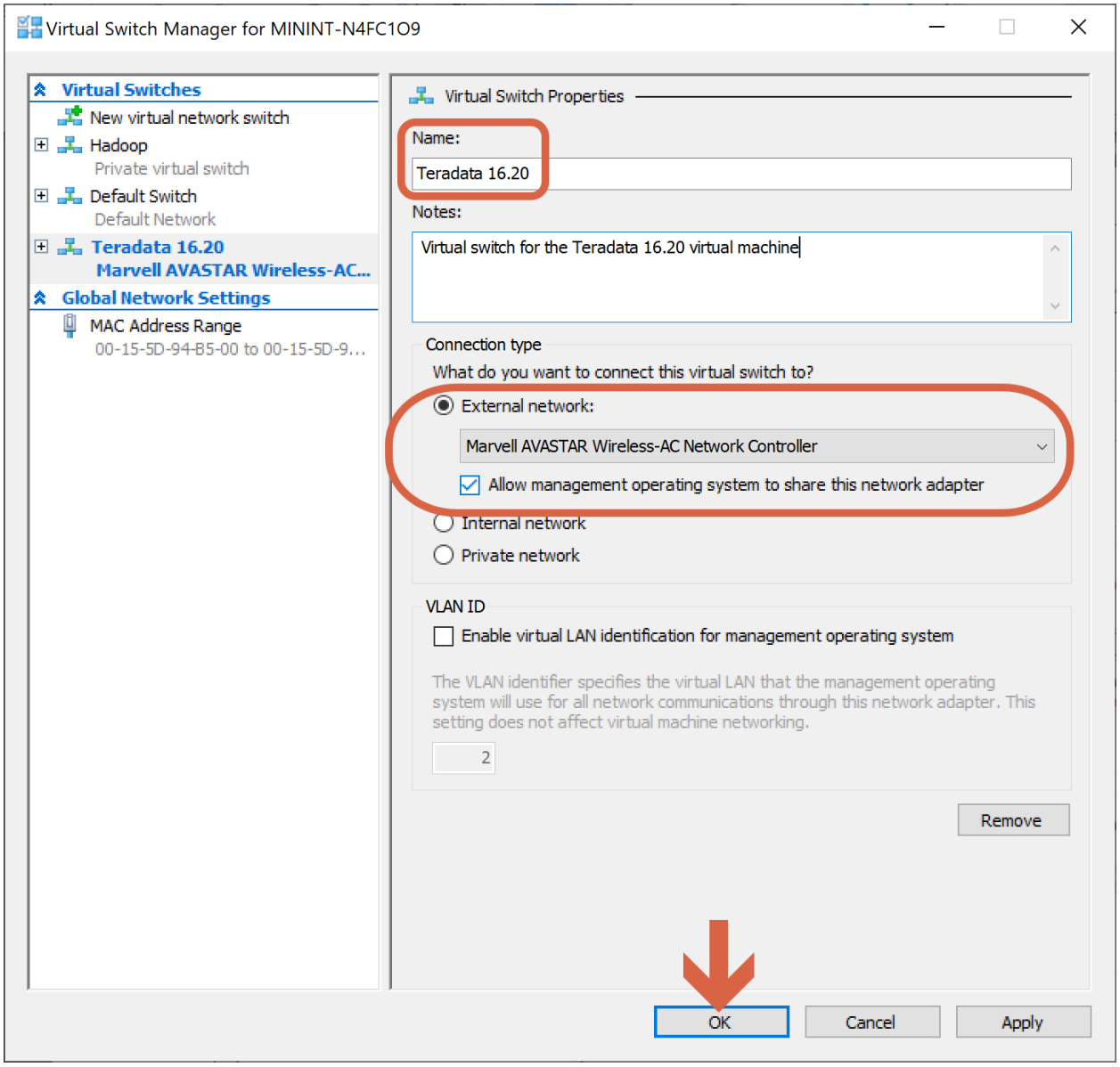
We open the Virtual Switch Manager.



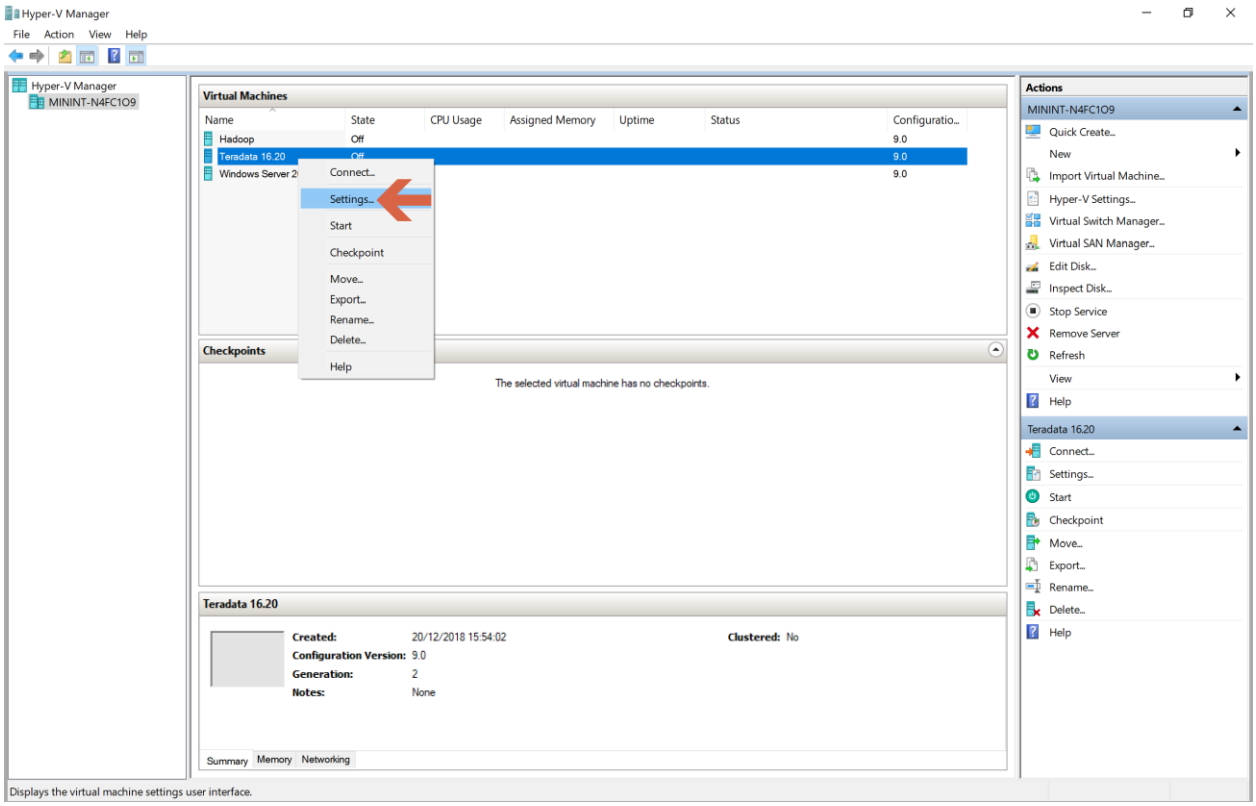
We choose “External” and click on “Create Virtual Switch”.



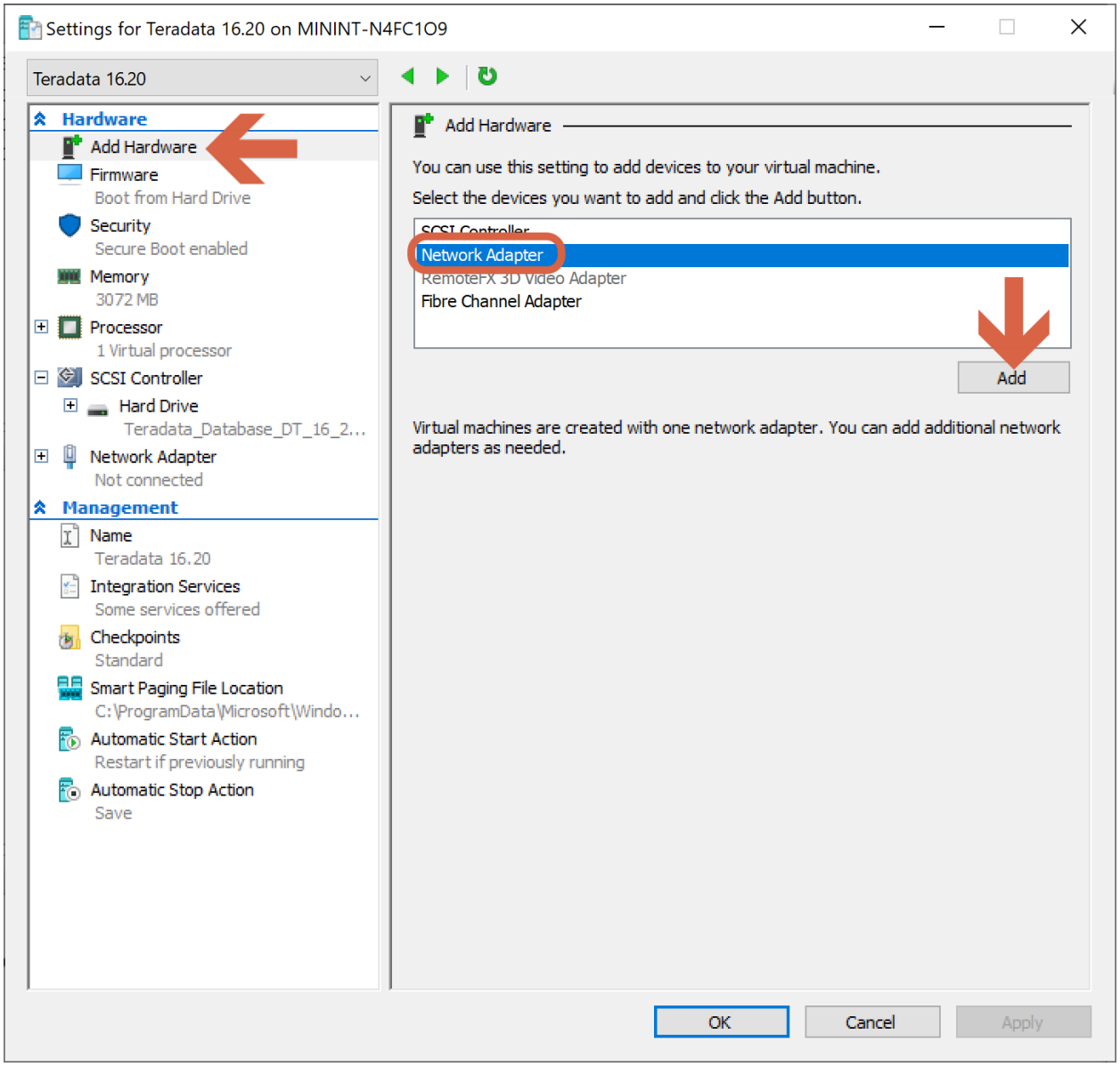
We name the virtual switch, confirm that it is an external network and click “OK”.



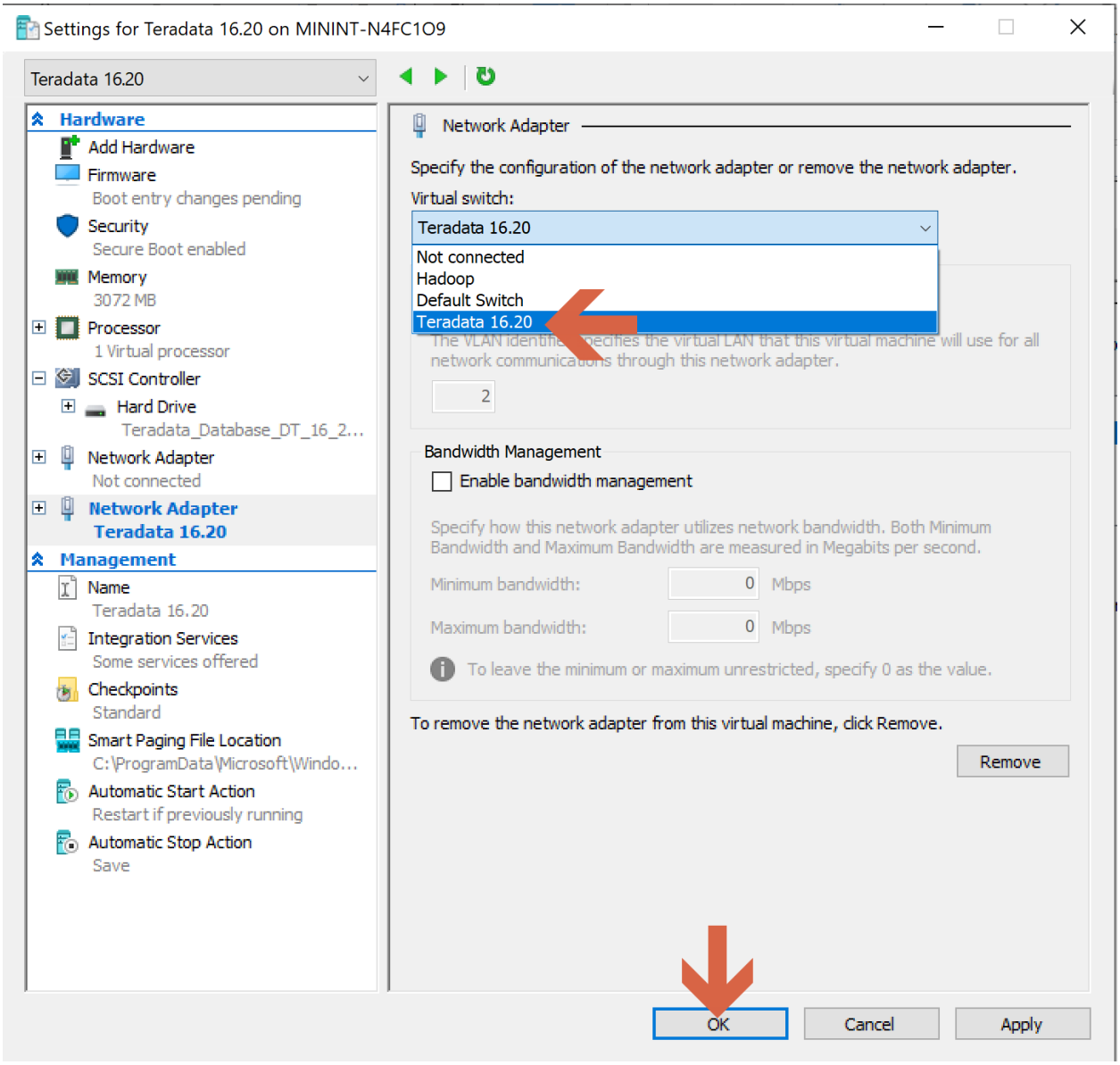
We right-click on the Teradata 16.20 virtual machine and select Settings.



We select “Add Hardware”, “Network Adapter” and click on “Add”.



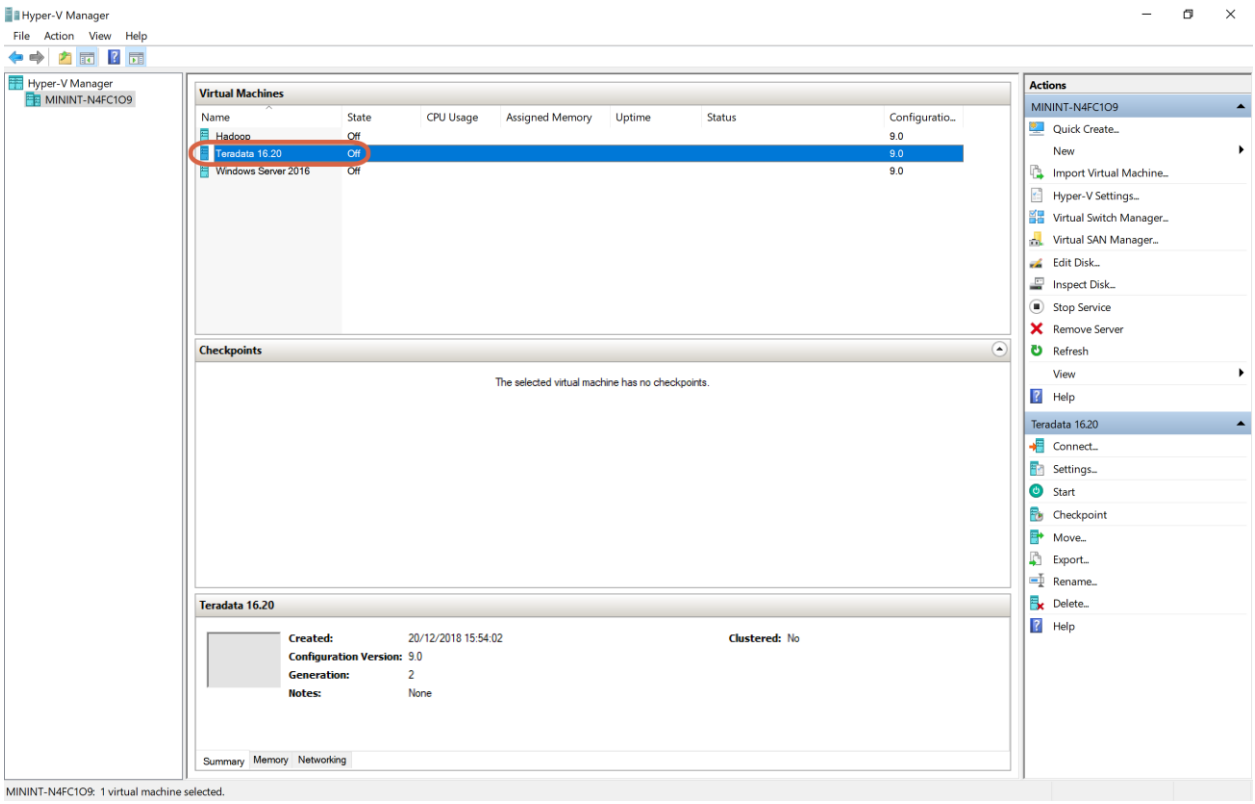
We select the virtual switch we have just created for the Teradata 16.20 virtual machine, and we click "OK".



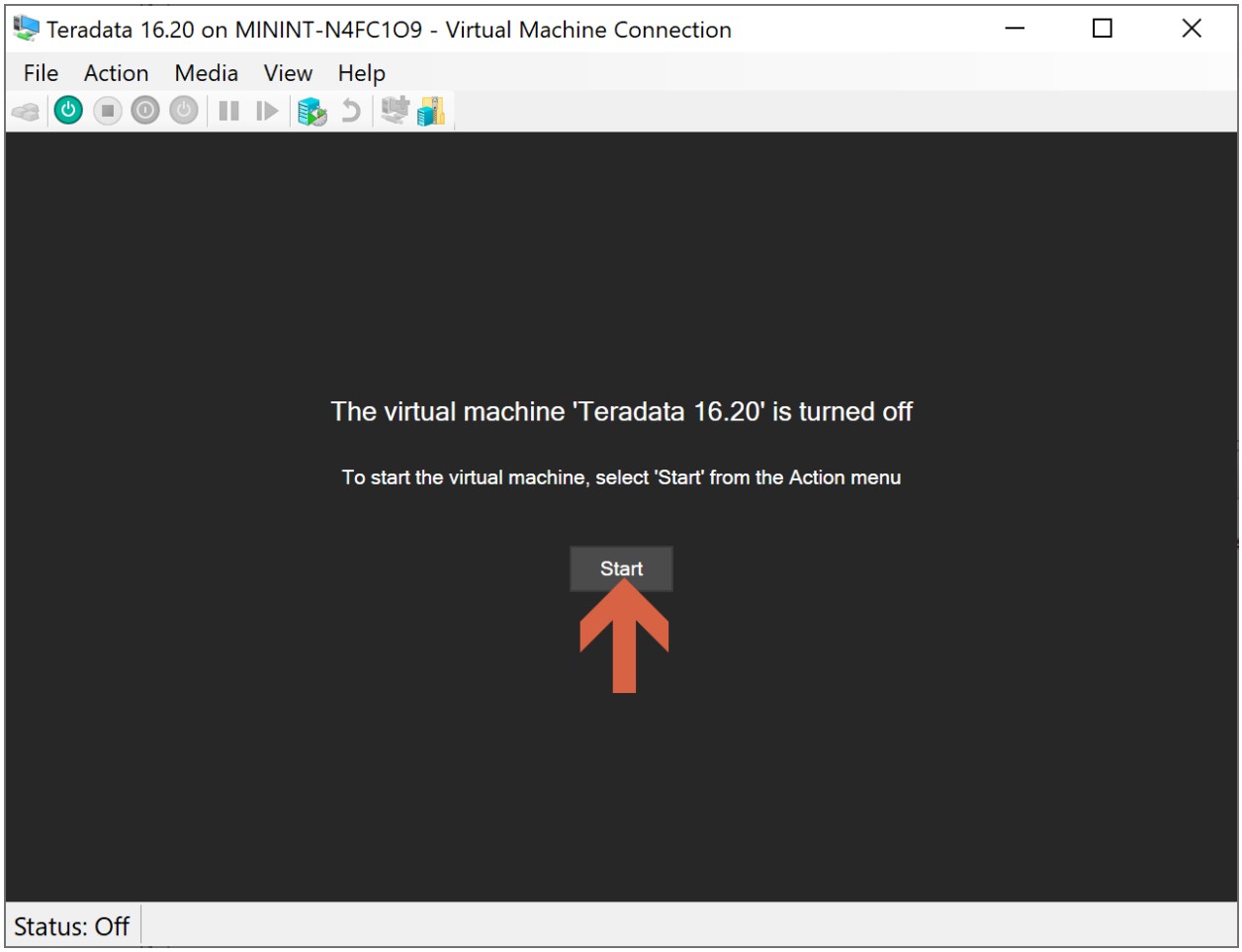
Now our Teradata virtual machine is ready to work with Hyper-V.

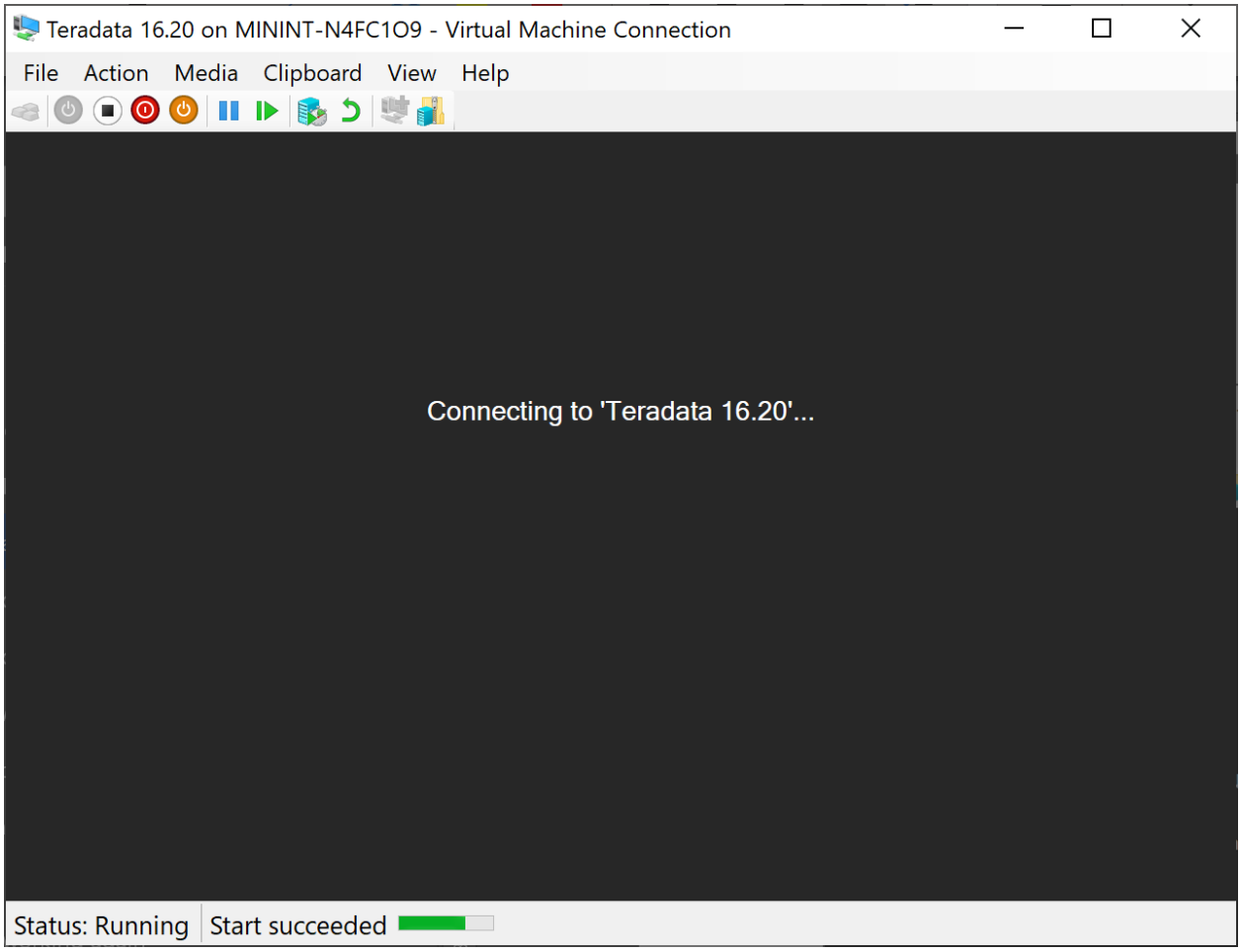
4. Setup the Teradata on Hyper-V (+ Viewpoint)

In the Hyper-V Manager, we double click on the Teradata 16.20 virtual machine.



We click on “Start”.





Network Configuration

When first installing Teradata on Hyper-V environment, there are some basic configuration steps that we need to make based on the IP addresses that are given to the instance when it is started. If at any point during your utilization, you have difficulty in communicating with the virtual machine, you need to check and validate that your network configuration is correct.

We typically use the NAT networking configuration. With this, the virtual machine manages the underlying virtual network for the instances and is the gateway for managing all inbound and outbound network traffic. This gateway provides the DHCP service to assign an IP address to the Teradata Express instance and will also provide the DNS services to resolve web domain names to their IP address. We'll need to first get the instance IP address as well as this gateway address. We'll do this using some basic Linux terminal command line utilities.

Double-click the “Gnome Terminal” icon to launch the terminal session.

The IP address assigned to your Teradata Express image is displayed with the ifconfig command:

```
# ifconfig eth0
```

This should result in output similar to this, in which I've bolded the IP address for this instance - 192.168.80.133. Yours will likely be a different address.

```
eth0 Link encap:Ethernet HWaddr 00:0C:29:FD:D9:5B
inet addr:192.168.80.133 Bcast:192.168.80.255 Mask:255.255.255.0
```

```
inet6 addr: fe80::20c:29ff:fefd:d95b/64 Scope:Link
UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
RX packets:381 errors:0 dropped:0 overruns:0 frame:0
TX packets:657 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:47151 (46.0 Kb) TX bytes:73442 (71.7 Kb)
Base address:0x2000 Memory:e8920000-e8940000
```

Now let's find the IP address being used to connect to the VMware gateway using the netstat command. Again, I've bolded the IP address for the gateway - 192.168.80.2. Again, yours will likely be a different address.

`route –n does the same as nstat -r`

```
# netstat -r
Kernel IP routing table
Destination       Gateway        Genmask        Flags MSS Window irtt
Iface
192.168.80.0      *              255.255.255.0  U 0 0 0 eth0
link-local        *              255.255.0.0    U 0 0 0 eth0
loopback          *              255.0.0.0      U 0 0 0 lo
default           192.168.80.2   0.0.0.0        UG 0 0 0 eth0
```

Now that we have our 2 address, let's check out Linux networking configurations.

First, let's look at the `/etc/resolv.conf` file. This file contains the address of our domain name server (DNS), which is used to translate network aliases to their IP address. For example, you can't browse the web without translating web domain names to their IP address. Use your favourite editing tool (such as `vi`) to update this file as I've done here:

```
# cat /etc/resolv.conf
nameserver 192.168.80.2
```

Once this is updated and saved, we can test using the ping utility with a web domain name:

```
# ping yahoo.com
PING yahoo.com (209.191.122.70) 56(84) bytes of data.
64 bytes from ir1.fp.vip.mud.yahoo.com (209.191.122.70): icmp_seq=1
ttl=128 time=106 ms
```

The second network configuration change we should make is to support Teradata connectivity. A common Teradata configuration used by client computers is to create what we call a 'cop' entry in the hosts file. This allows our connectivity tools to connect to Teradata using aliases instead of IP address. An example is the BTEQ query tool included in the Teradata Express image. I can create an alias name "localTD" in my `/etc/hosts` file with the pattern '{alias}cop1', or 'localTDcop1' as seen here mapped to my Teradata Express IP address:

```
# cat hosts
# hosts This file describes a number of hostname-to-address
# mappings for the TCP/IP subsystem. It is mostly
# used at boot time, when no name servers are running.
# On small systems, this file can be used instead of a
# "named" name server.
# Syntax:
```

```
#
# IP-Address Full-Qualified-Hostname Short-Hostname
#

127.0.0.1 localhost
192.168.80.133 localTDcop1
```

With this new 'cop' entry, my client tools can connect to Teradata using the alias instead of IP address. Here's an example using the BTEQ client on the Teradata Express instance. I've highlighted the alias entry. (Remember, though, if you want to use this alias from another client such as BTEQ from your Windows host environment, you'll need to create a similar 'cop' entry in its hosts file.)

```
# bteq
Teradata BTEQ 16.20.07.01 for LINUX.
Copyright 2018, Teradata Corporation. ALL RIGHTS RESERVED.
Enter your logon or BTEQ command:
.logon localTD/dbc

.logon localTD/dbc
Password:

*** Logon successfully completed.
*** Teradata Database Release is 13.00.00.19
*** Teradata Database Version is 13.00.00.19
*** Transaction Semantics are BTET.
*** Character Set Name is 'ASCII'.

*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
select * from dbcinfo;

select * from dbcinfo;

*** Query completed. 3 rows found. 2 columns returned.
*** Total elapsed time was 1 second.

InfoKey InfoData
------------------------------  ----------------------------------
-------
RELEASE 16.20.07.01
VERSION 16.20.07.01
LANGUAGE SUPPORT MODE Standard

BTEQ -- Enter your DBC/SQL request or BTEQ command:
.exit

.exit
*** You are now logged off from the DBC.
*** Exiting BTEQ...
*** RC (return code) = 0
```

With these simple configuration updates, your Teradata Express virtual instance should be ready to go.

Checking Teradata Status

Let's start with the basic status tool, pdestate:

```
1    # pdestate -a
```

If all is well and Teradata is running, you will see this output:

```
1    PDE state is RUN/STARTED.
2    DBS state is 5: Logons are enabled - The system is quiescent
```

Stopping the database

Should you need to stop the database in a controlled manner, the command is:

```
1    # tpareset -x bringing down the database
```

The syntax for this command is tpareset -x {comment}, where the comments can be used for logging/auditing purposes to capture the reason that the database was stopped.

Here is output from this command. Notice the prompt as a last check to make sure that this is what you really want to do.

```
1    You are about to shutdown the database
2    on the system
3    's10-1300'
4    Do you wish to continue (default: n) [y,n] y
```

Once the database has been stopped, you can check the status of the database like we did earlier:

```
1    # pdestate -a
2    PDE state is STOP/KILLTASKS.
```

But if there was a problem that halted Teradata instead of a controlled shutdown, you will see this message from pdestate:

```
1    #pdestate -a
2    "down/hardstop" is down
```

Restarting Teradata

To restart Teradata after manually stopping the database or to restart after a 'hardstop' event, run this command:

```
1    # /etc/init.d/tpa start
```

Which should give this output:

```
1    Teradata Database Initiator service is starting...
2    Teradata Database Initiator service started successfully.
```

To restart Teradata if it is any state other than "down/hardstop" use the command:

```
1    # tpareset -f restarting
```

This asks for a confirmation before beginning the restart process:

```
1    You are about to restart the database
2    on the system
3    's10-1300'
4    Do you wish to continue (default: n) [y,n] y
```

You can then issue the watch the various start up levels by issuing the pdestate command every few seconds..

```
1    # pdestate -a
2    PDE state is START/RECONCILE.
```

```
1    # pdestate -a
2    PDE state is START/STARTTPA.
```

```
1    # pdestate -a
2    PDE state is RUN/STARTED.
3    DBS state is 1/5: DBS Startup - Voting for Transaction Recovery
```

```
1    # pdestate -a
2    PDE state is RUN/STARTED.
3    DBS state is 1/4: DBS Startup - Starting PE Partitions
```

1 # pdestate -a

2 PDE state is RUN/STARTED.

3 DBS state is 5: Logons are enabled - The system is quiescent

Starting Viewpoint Services

Once the RAM is allocated, we can start the Viewpoint services. Click the Computer (Start button) on the lower left corner and choose:

YAST->System->System Services (Runlevel)

To set Viewpoint to automatically start during boot up, Enable the following services:

- postgresql
- dcs
- viewpoint

To manually start the Viewpoint servers:

A. Open up a Linux command prompt and enter the following 3 commands, waiting 1 minute for each to complete.

B. /etc/init.d/dcs start

C. /etc/init.d/postgresql start can check the status with /etc/init.d/postgresql status

D. /etc/init.d/viewpoint start You can monitor progress in /opt/teradata/viewpoint/logs/viewpoint.log

E. When all is up, just put the ip address in the Firefox Browser window. You can use ifconfig in a Linux terminal to see the eth0 IP as (example): 192.168.203.131.

F. the Viewpoint user is admin, and the Viewpoint password is teradata.

You can refer to a previous post I wrote on setting up Teradata in Azure, as these steps are the same

5. Connect to Teradata Express with Teradata Studio Express

Connect to Teradata Express with Teradata Studio Express

6. Connect to Teradata Express with other clients – Example BTEQ

Connect to Teradata Express with other clients – Example BTEQ