Relativity Dev VM
Pre-built VM
Documentation

[March 20, 2020]

Table of Contents

Introduction

The Developer Experience team has created virtual instances of Relativity environments that they are sharing with the Relativity Developer Community. These Dev VM's have been designed to help developers test the functionality of their Relativity applications. Please note that these VM's should not be used to performance test your applications as system resources are not equivalent to suggested production configurations. One key development feature that is included with the VM's is the ability to remotely debug your code.

Intended Use Cases

- Basic Relativity Development such as creating applications with Custom Pages, Agents, Event Handlers, etc.
- Remote Debugging Custom Applications
- The intended use of the DevVm is meant to be ephemeral. Loading of test data should be automated and there should be no upgrading the DevVm through the Relativity Installer files.

Not-supported

The following are **not** supported in Dev VM.

- Changing the VM name.
- Setting the VM in the cloud like Azure, AWS etc.
- Getting the VM image in a VM format other than Hyper-V.

License

The Dev VM comes with the following licenses.

- **Windows Server 2012 R2 Standard**
    - 6-month trial license
    - Instructions to update Windows license - https://github.com/relativitydev/relativity-dev-vm/wiki
- **Relativity**
    - Up to 7-day trial license. Depending on when you request a DevVM, it may come with an expired license.
    - Contact support@relativity.com to get a new DevVM Developer license
- **Processing**
    - Up to 7-day trial license. Depending on when you request a DevVM, it may come with an expired license.
    - Contact support@relativity.com to get a new DevVM Developer license
- **For Prior to 10.1 (Blazingstar), SQL Server 2016 Developer edition**
    - Free for non-production use
    - More info on SQL Licensing - https://download.microsoft.com/download/9/C/6/9C6EB70A-8D52-

48F4-9F04-08970411B7A3/SQL_Server_2016_Licensing_Guide_EN_US.pdf
- **For 10.1 (Blazingstar) and above, SQL Server 2017 Developer edition**
  - Free for non-production use
  - More info on SQL Licensing – http://download.microsoft.com/download/7/8/c/78cdf005-97c1-4129-926b-ce4a6fe92cf5/sql_server_2017_licensing_guide.pdf

Note: *You must update the licenses to ensure continuous functioning of the Dev VM.*

FAQ
- If you run into any issues with DevVM, please refer to the DevVM category on the Relativity DevHelp Community - https://devhelp.relativity.com/c/tools-testing-download-and-tutorials/devvm
- Instructions to sign up for Relativity DevHelp Community - https://platform.relativity.com/9.6/Content/Site_Resources/Get_started_with_DevHelp.htm
- Secret Store unseal key location: C:\Program Files\Relativity Secret Store\unseal.txt

System Requirements
- Windows 10/ Windows Server 2016 --> Hyper-V Version 6.2 or above
- Windows Server 2012 R2 --> Hyper-V Version 5.0
- Default on Import
  - 2 processing cores
  - 8GB RAM
  - 140GB free storage space.
- **Recommended Update** for a better experience:
  - 4 processing cores
  - 16GB RAM

Accounts
Windows Accounts
Username: **Administrator**

Password: **Test1234!**

Note: Use this account for configuring the VM, remoting into VM, remote debugging, accessing Event Viewer, SQL Server, etc.

Relativity Accounts

Username: **relativity.admin@relativity.com**

Password: **Test1234!**

SSMS (SQL) Accounts

Login: **EDDSDBO** (Recommended)

Password: **Test1234!**

*Note: Use the **EDDSDBO** account for all Relativity SQL related queries.*

Login: **sa**

Password: **Test1234!**

*Note: Use the **sa** account for any SQL Administrative tasks.*
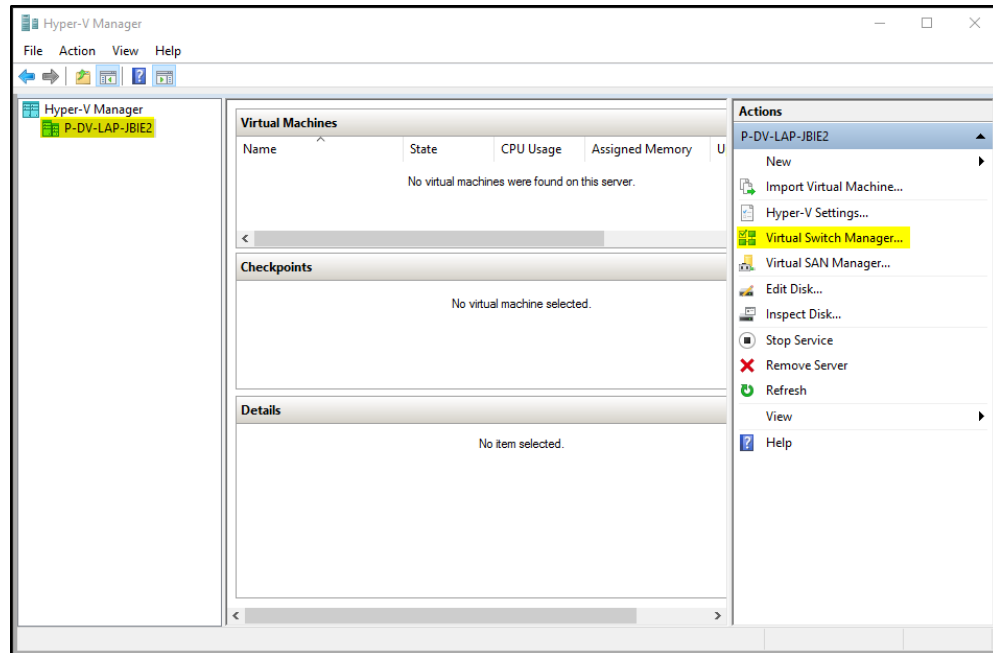
Features not available in Dev VM

- Truncating logs
- Database backups
- SMTP configuration

Instructions for setting up a Dev VM

1. Each Dev VM uses Microsoft **Hyper-V** to spin up a Relativity instance.
2. Make sure **Hyper-V** is installed on your local machine.
   a. To verify that you have **Hyper-V** installed, go to **Programs and Features** and click the **Turn Windows features on or off** link and look for **Hyper-V**. Confirm that the checkbox next to **Hyper-V** is selected (checked), see screenshots below.
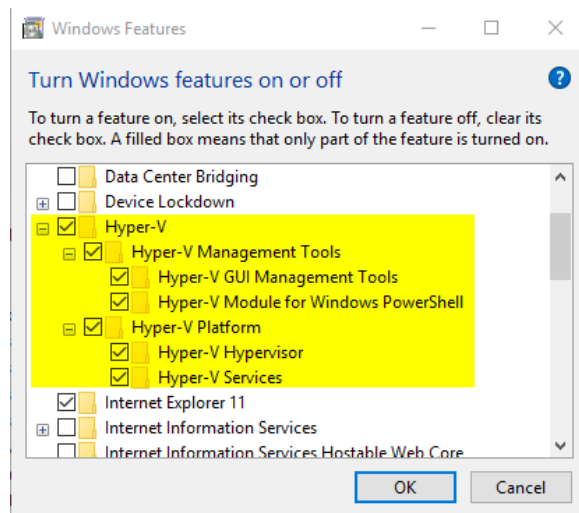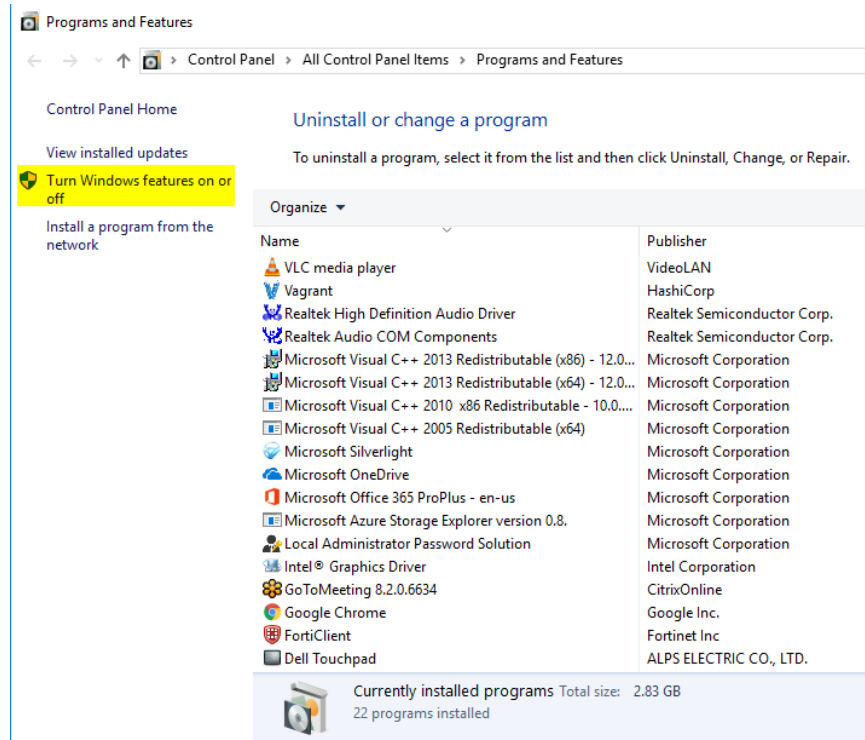
    b. If **Hyper-V** is not checked, please click the check box next to **Hyper-V** and click **OK** to install.

3. Extract the provided zip file and copy the **RelativityDevVm** folder onto your computer's C drive.

4. Open the **Hyper-V Manager** application.

5. Make sure you have an **External Virtual Switch** which lets you access your VM from your local machine.
   a. Select your local machine from the left pane.
   b. In the Actions pane located in **Hyper-V Manager**, click on the **Virtual Switch Manager** link, see screenshot below.

c. If you don't already have an external network switch please click the **New virtual network switch** link under **Virtual Switches**. Select **External** for the type of switch and click on the **Create Virtual Switch** button.
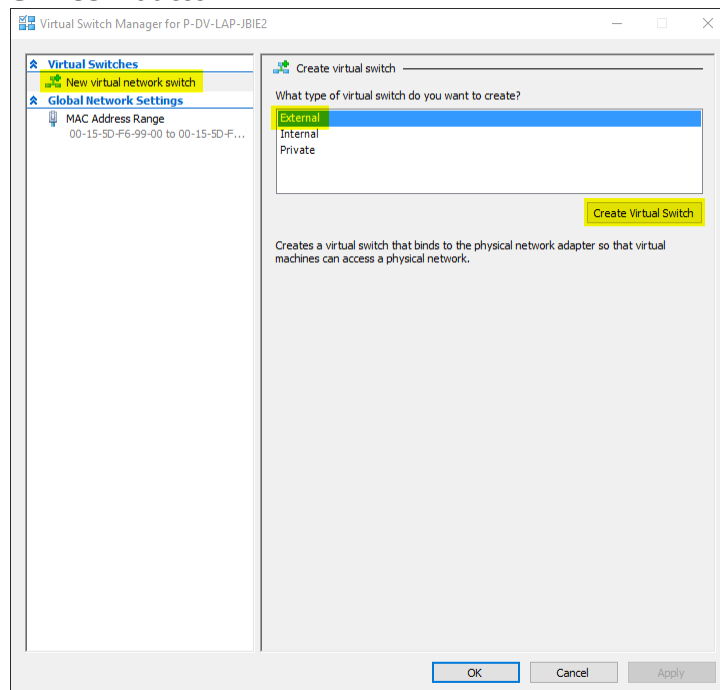
d. Provide a friendly name for the new switch (For example: **RelativityDevVmSwitch**) and click **Apply** button.
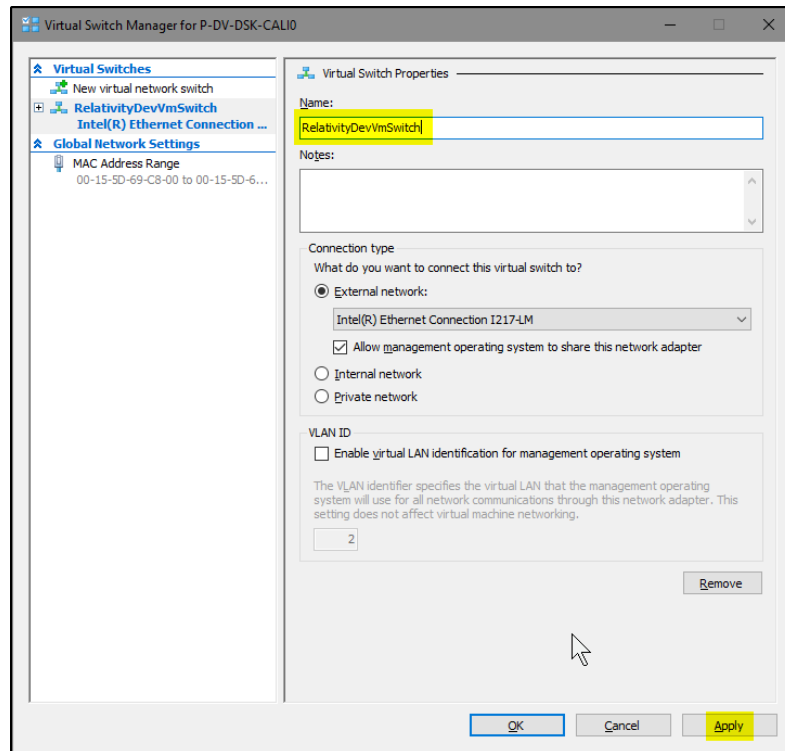
e. If you get a warning message (see screenshot below), please click the **Yes** button.
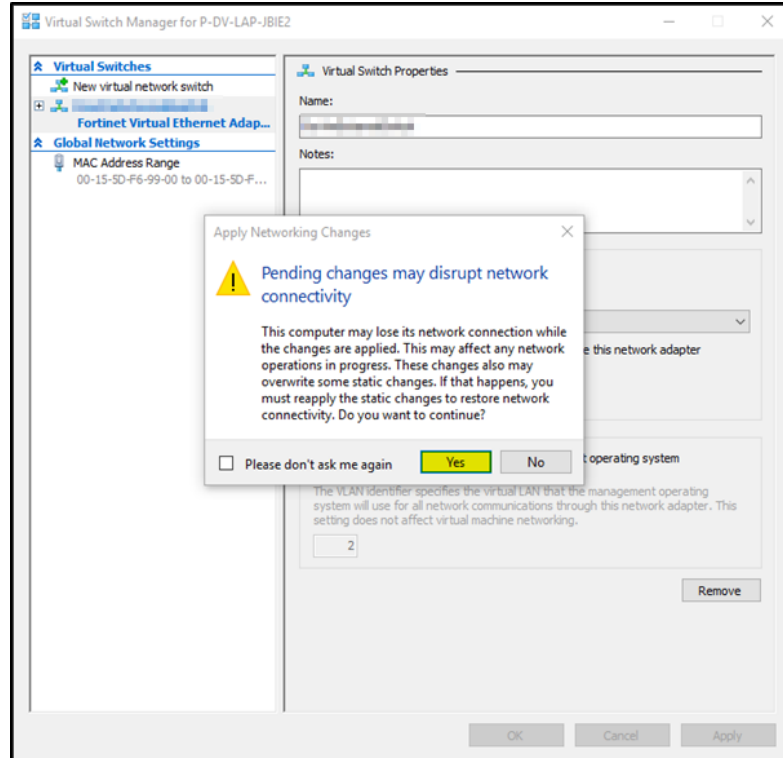
f. Next click the **OK** button to create your new virtual switch.
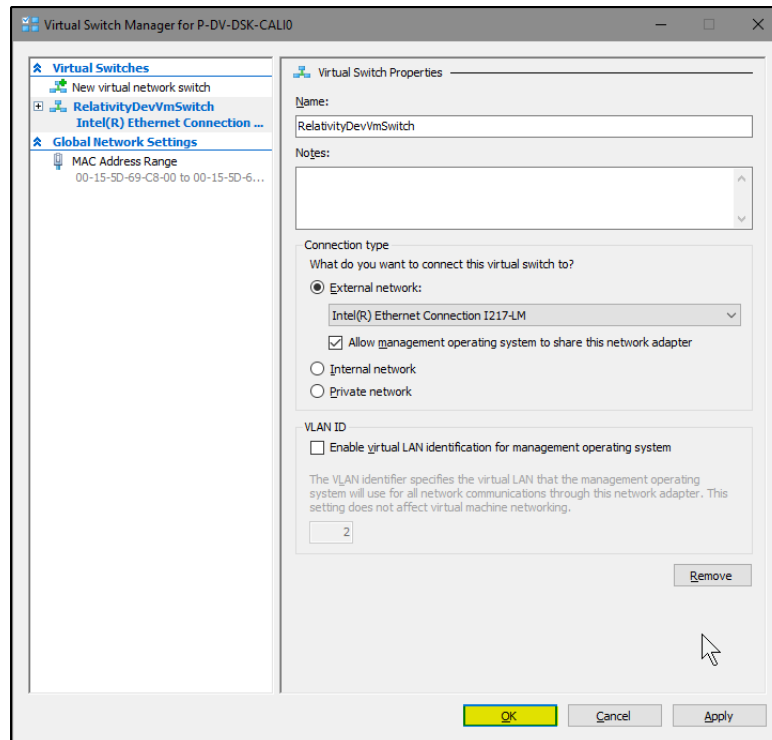
6. To import the Hyper-V VM, please click on the **Import Virtual Machine** link in the right-side pane located within **Hyper-V Manager**, see screenshot below.
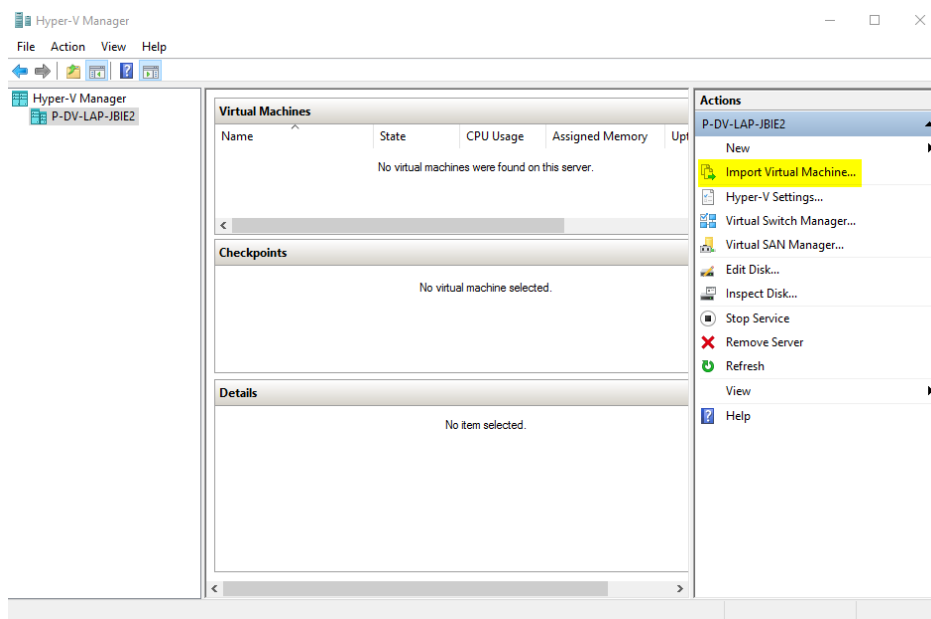
7. Click the **Next** button.

Import Virtual Machine   ✕

↗ **Before You Begin**

Before You Begin
Locate Folder
Select Virtual Machine
Choose Import Type
Summary

This wizard helps you import a virtual machine from a set of configuration files. It guides you through resolving configuration problems to prepare the virtual machine for use on this computer.

☐ Do not show this page again

< Previous   **Next >**   Finish   Cancel

8. In the **Locate Folder** section, click the Browse to select the location where you copied the **RelativityDevVm** folder in step 3 located above.
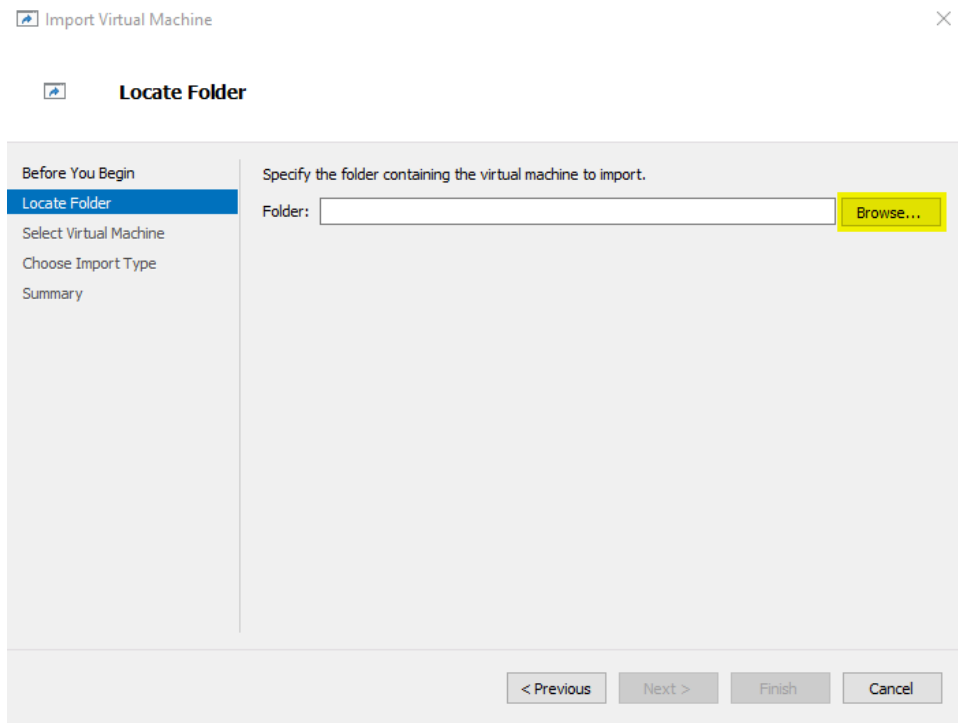
9. Select the **RelativityDevVm** folder and click the **Select Folder** button.
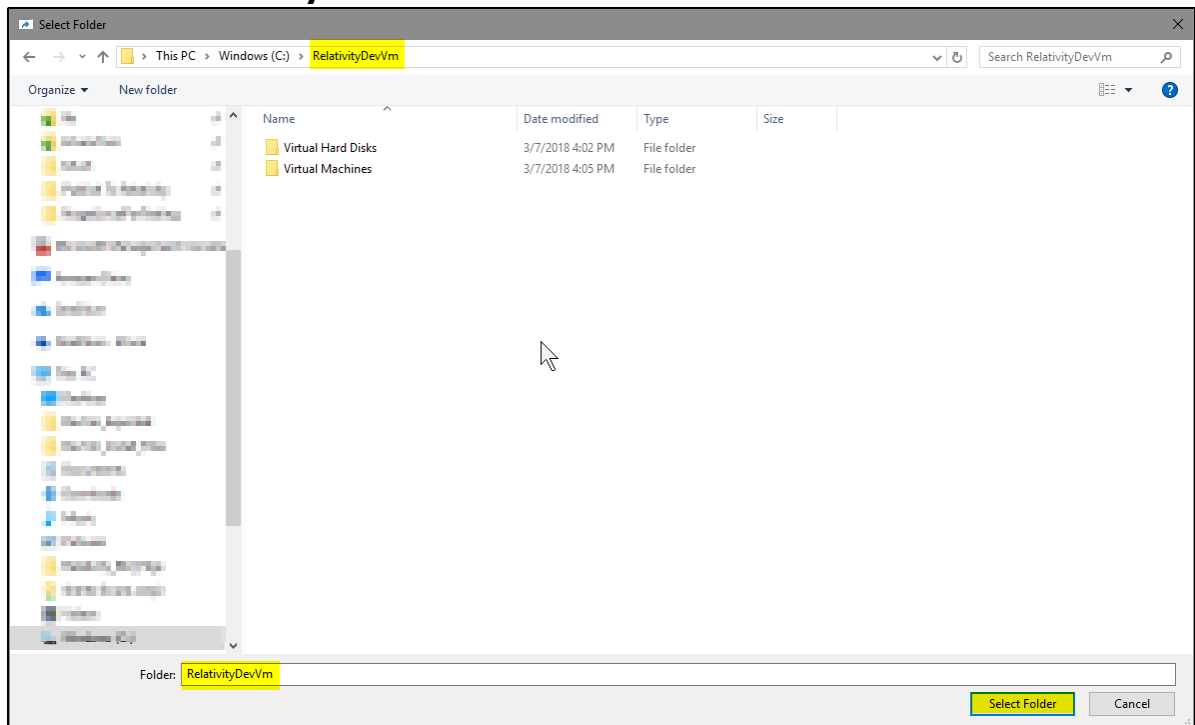
10. In the **Locate Folder** section, click the **Next** button.
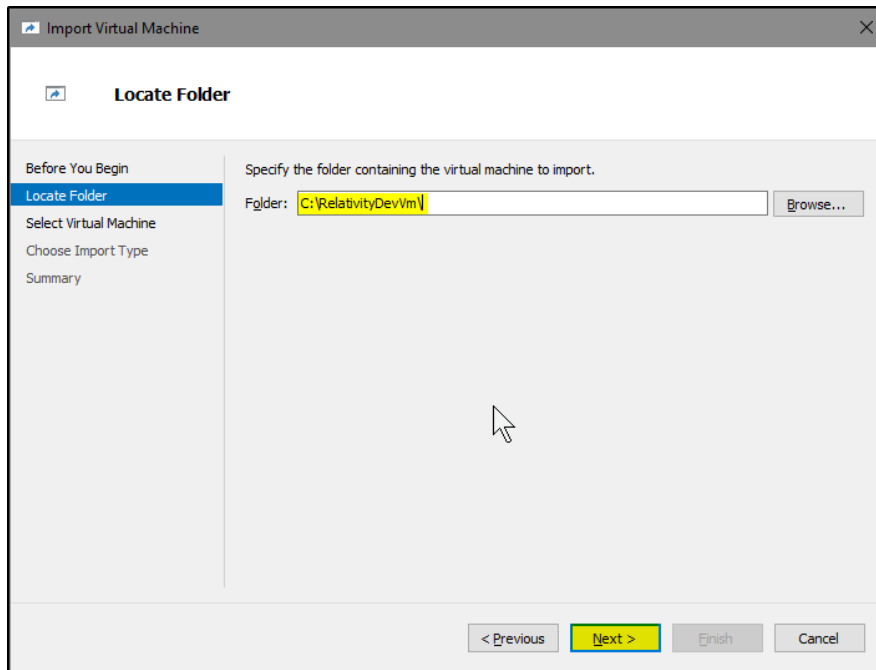
11. In the **Select Virtual Machine** section, you should see the Hyper-V VM as shown in the below screenshot. Verify and click the **Next** button.
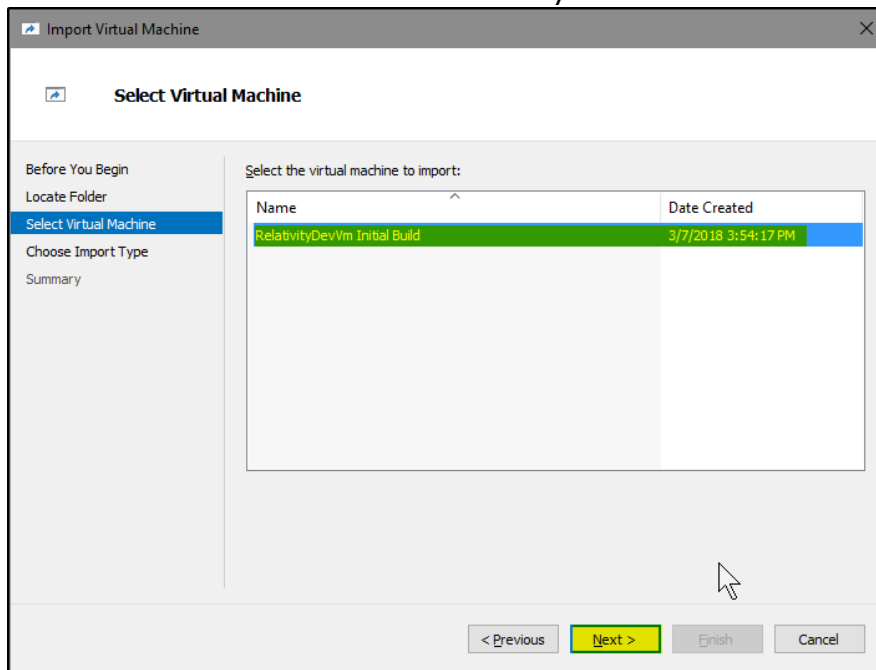

12. In the **Choose Import Type** section, choose the **Register the virtual machine in-place (use the existing unique ID)** option and click the **Next** button.
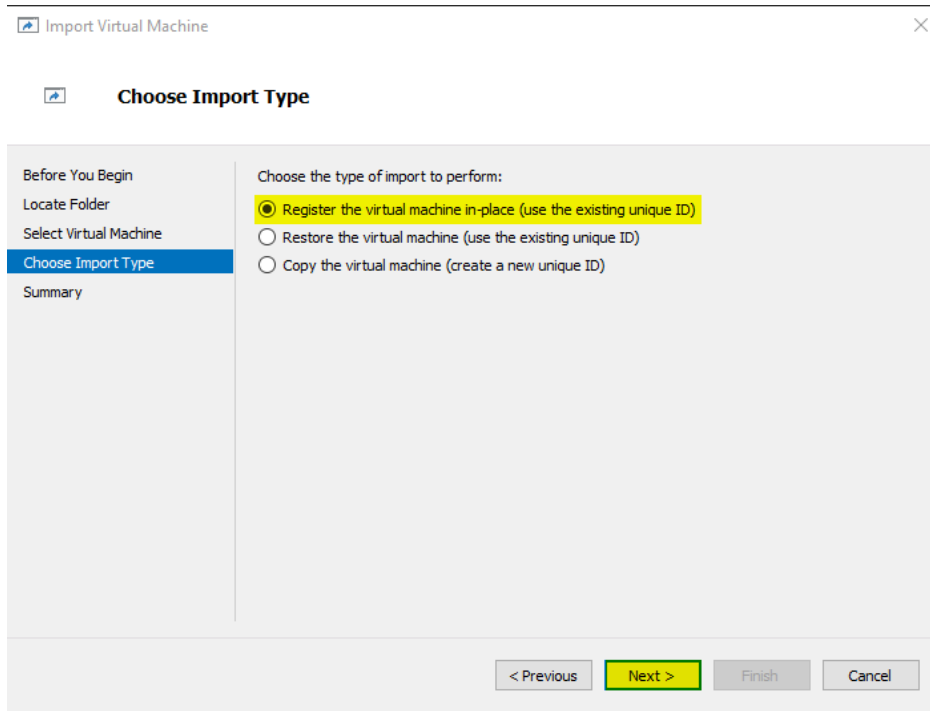
Note: *In **Connect Network** section, if the DevVM is looking for a different virtual switch then you might get a warning. Please choose a valid virtual switch from the **Connection** dropdown list.*
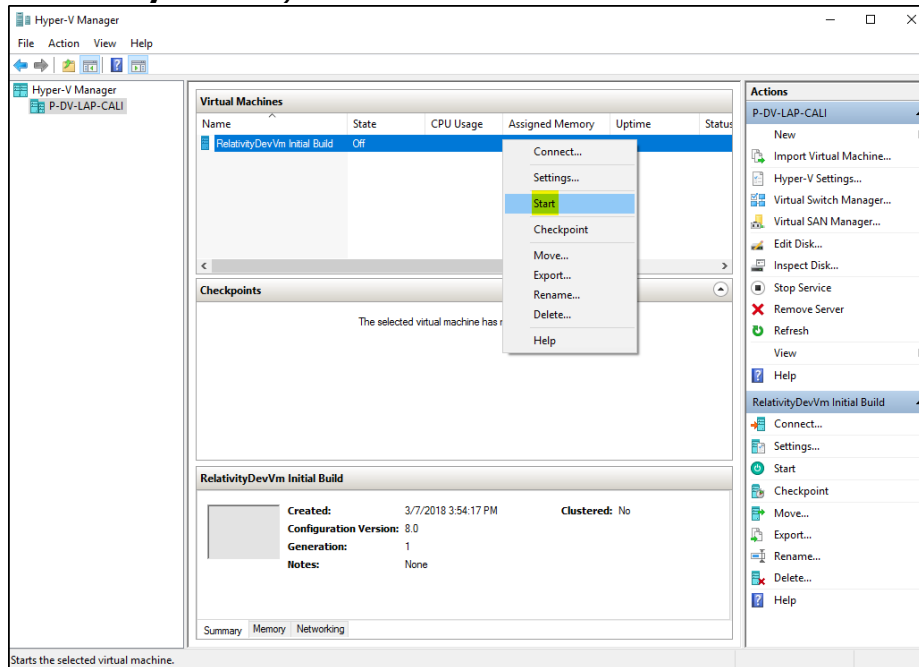
13. Next click the **Finish** button to complete the import of your new Hyper-V VM.
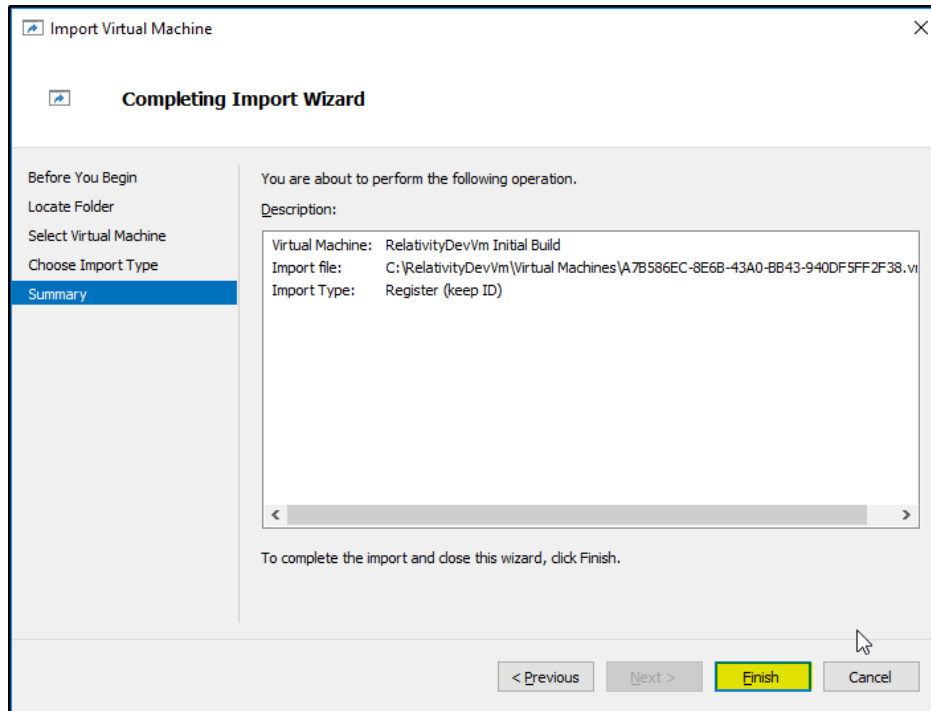
14. Now you should see the imported VM in the **Virtual Machines** section within **Hyper-V Manager**, see screenshot below.
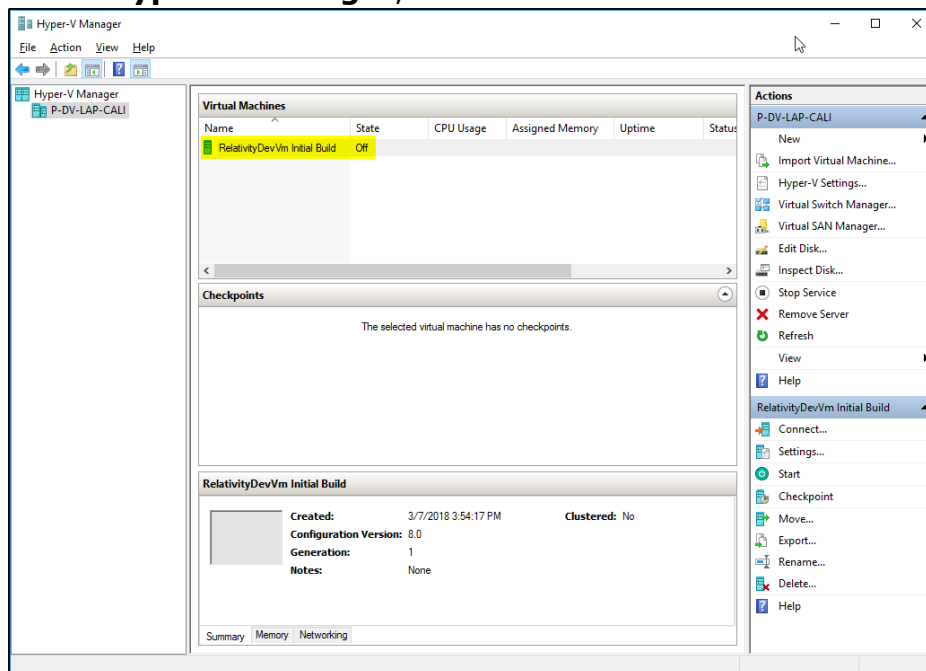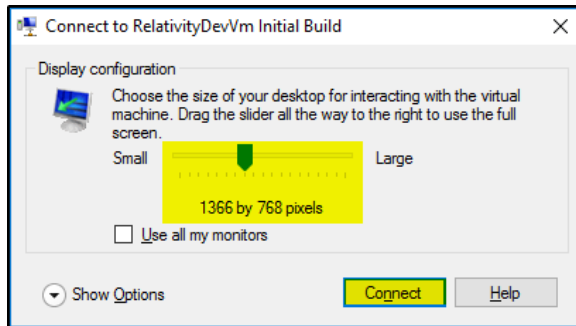
15. To start the Dev VM, right click on the VM instance (in the example below, **RelativityDevVm**) and select **Start**.
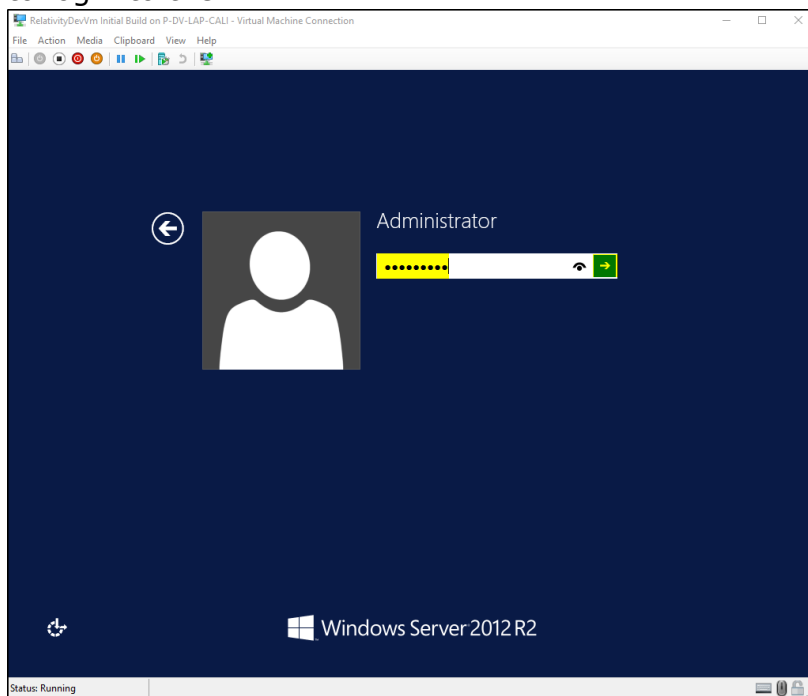

16. To connect to the Dev VM, right click on the VM instance and select **Connect**.
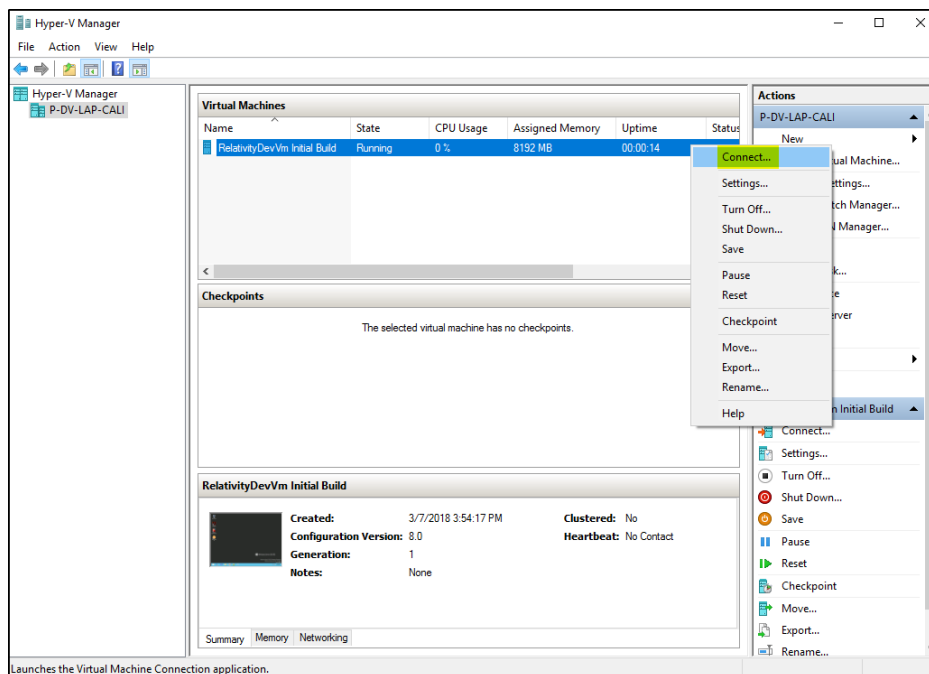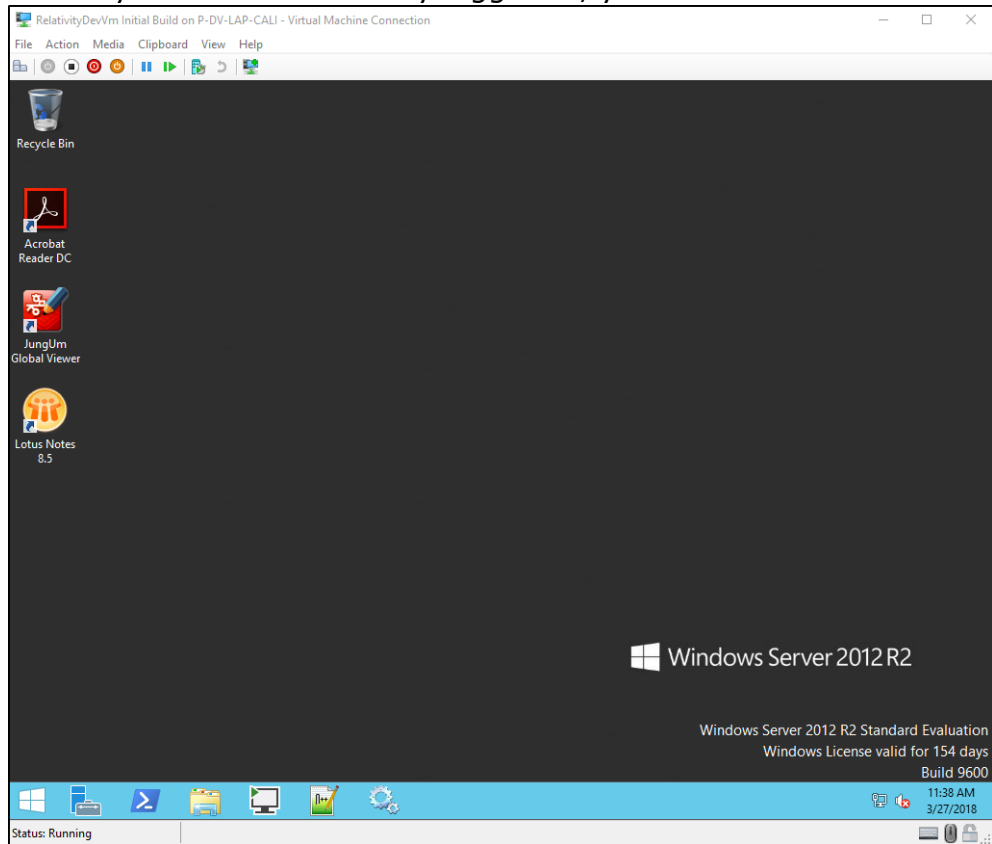
17. Select the appropriate resolution for your Dev VM and click the **Connect** button.

18. Use the **Administrator** account credentials located in the Accounts section to login to the VM.

19. Once you have successfully logged in, you should see the Desktop screen.

Updating Resources for VM

Note: Please be aware that even though you have the ability to adjust the system resources, it still is not recommended to use these environments for performance testing, gathering metrics, etc.

1. First make sure that the VM is shut down. Then right click the VM and select **Settings** as shown in below screenshot.
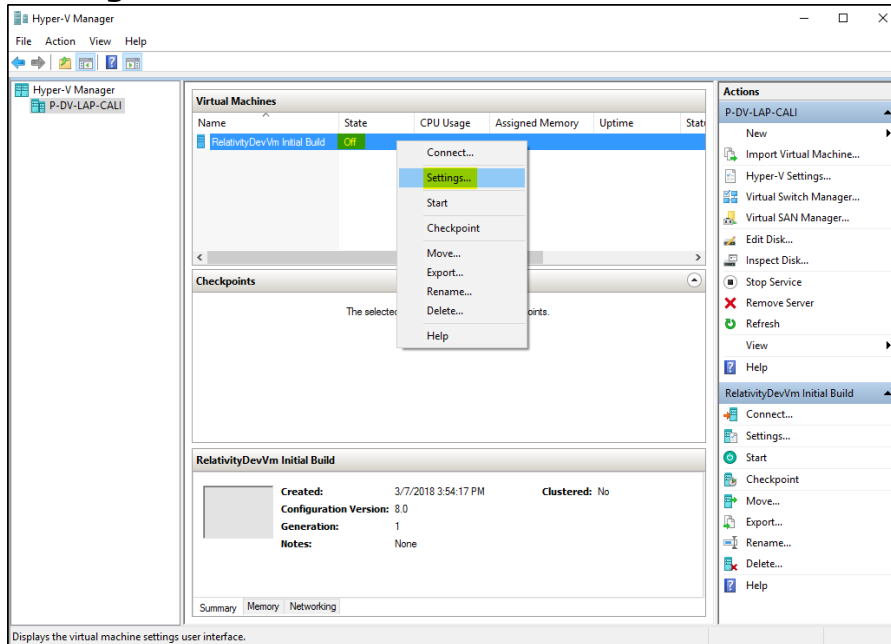
2. If you want to increase the number of processor cores for the VM, click on the **Processor** link in the **Hardware** section and input the desired **Number of virtual cores** for the VM as shown in below screenshot.
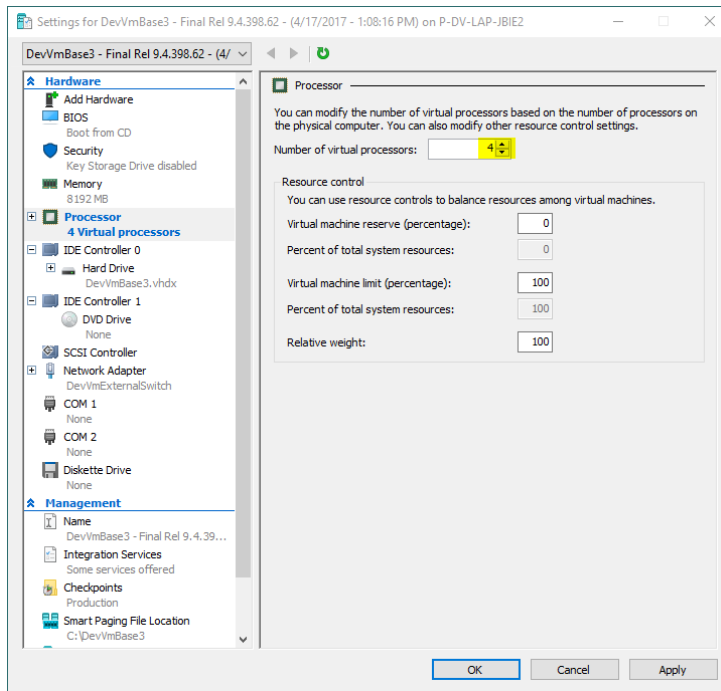
3. If you want to increase the RAM for the VM, click on the **Memory** link in the **Hardware** section and input the desired **RAM** for the VM as shown in below screenshot.
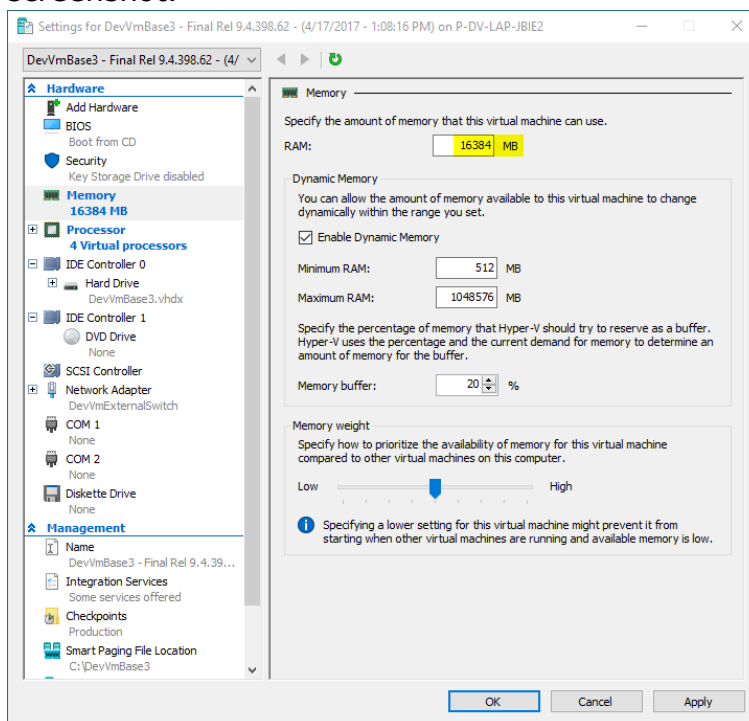

4. After you are done configuring the VM hardware, click the **Apply** button and then click the **OK** button.

Instructions for Expanding the C Drive on the DevVm

This feature is only available on the 10.2.227.16 and 10.3.142.22 DevVms and above

1. Shut down the VM and delete all checkpoints created.
2. Right Click on the VM and select Settings
3. Next on the left side select Hard Drive and on this page select Edit
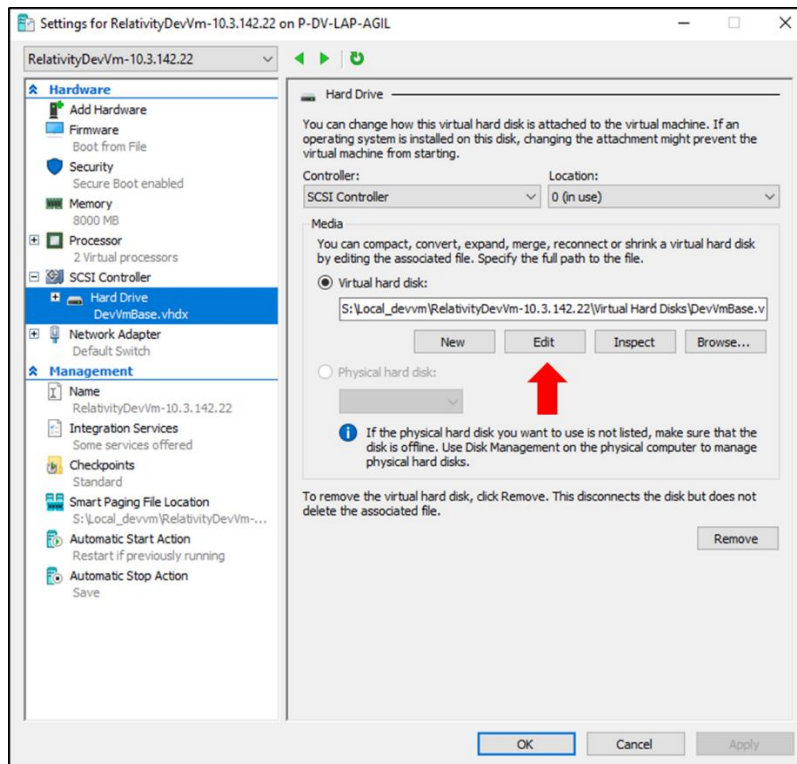
4. Click Next → Check Expand → Next → Enter the desired new disk size and click Finish. (Note: Do not give the DevVm more space than is available in the drive where the DevVm files exist on your local machine).
5. You can now turn on the DevVm
6. Right click on the windows icon in the bottom left corner of the DevVm and select Computer Management.
7. Next select Disk Management and right click on the c drive at the bottom and choose Extend Volume.
8. Then click Next until you can choose Finish and select Finish. You will now have an expanded disk.

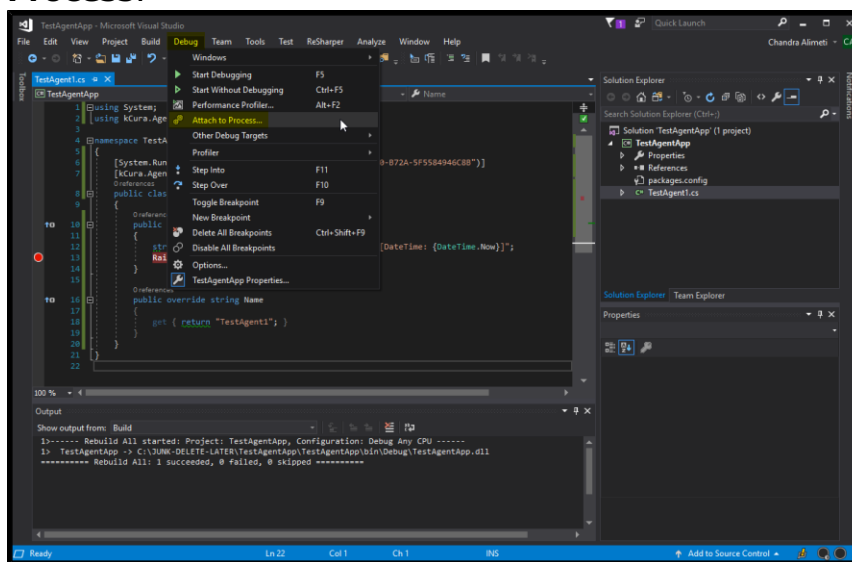# Remote Debugging with Visual Studio 2017

1. Remote into the Dev VM and launch **Visual Studio 2017 Remote Debugger** application which is pinned to **Taskbar**.
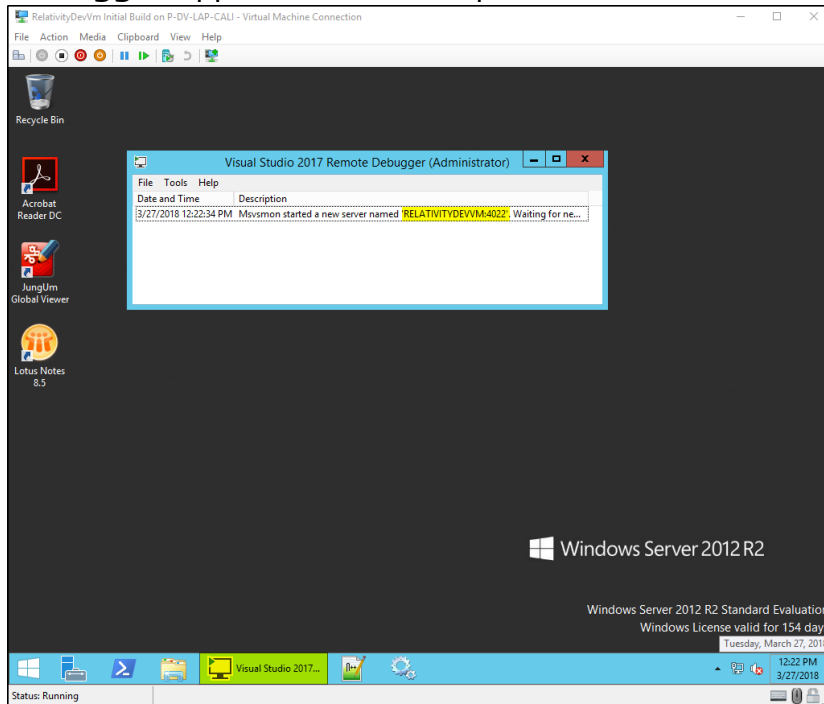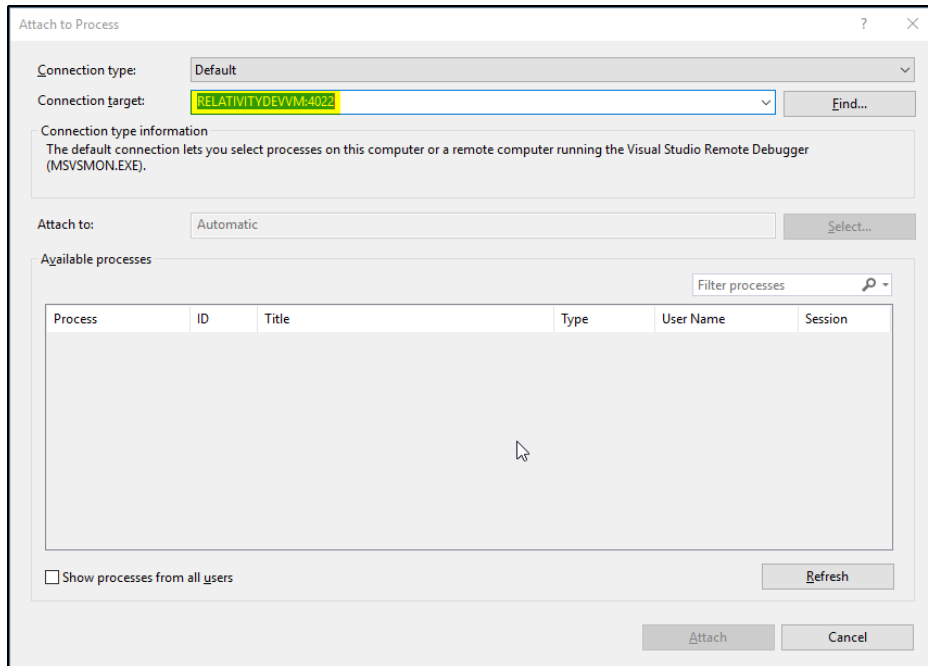

2. Open your visual studio project in **Visual Studio 2017** and add a breakpoint on the source code line you want to monitor.
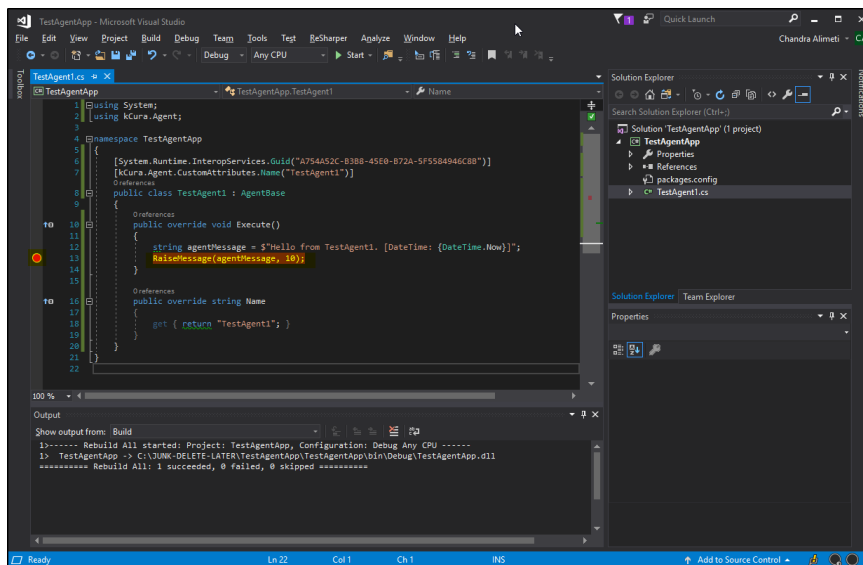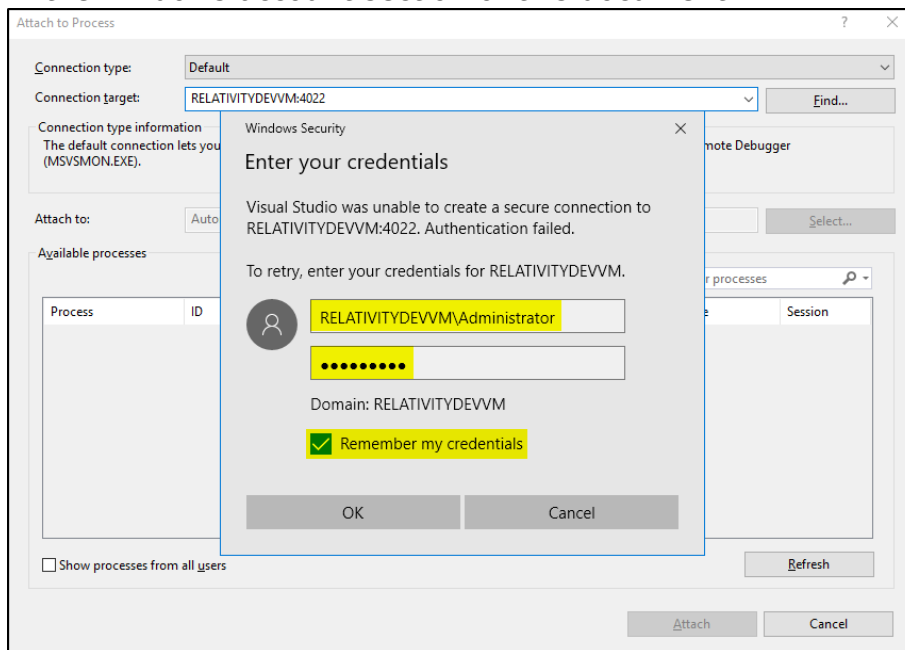
3. In **Visual Studio 2017**, go to **Debug** menu option and select **Attach to Process**.
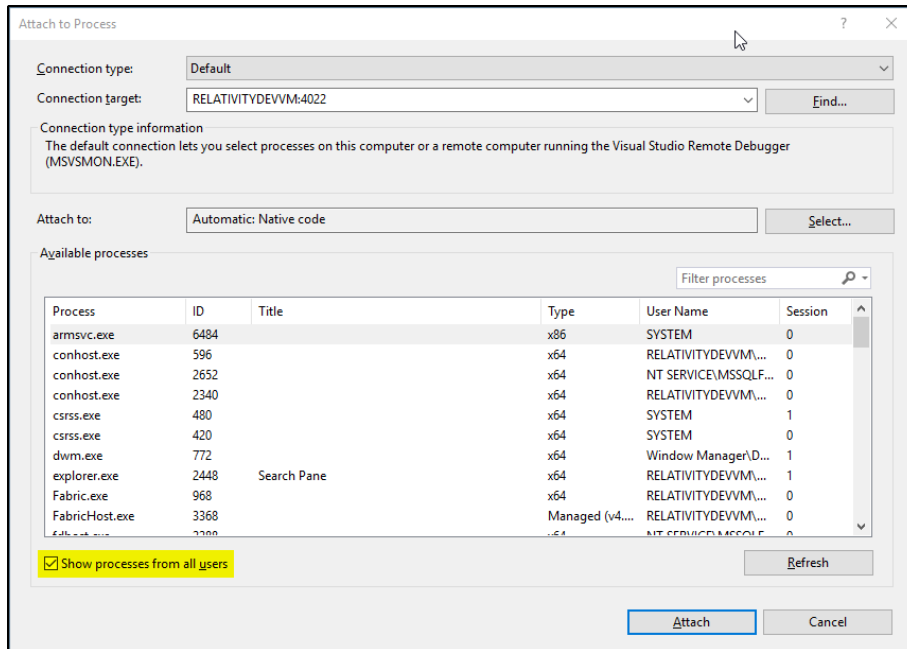

4. In the **Attach to Process** windows, enter your VM name along with port 4022 as shown in below screenshot and press **Enter**. (Example: **RELATIVITYDEVVM:4022**)

5. If prompted for credentials, enter the **RelativityService** credentials provided in the windows account section of this document.

6. Next check the box for option **Show processes from all users**.

7. If you want to troubleshoot agent code, select both the **kCuraAgentManager.exe** processes and click on the **Attach** button as shown in the below screenshot.
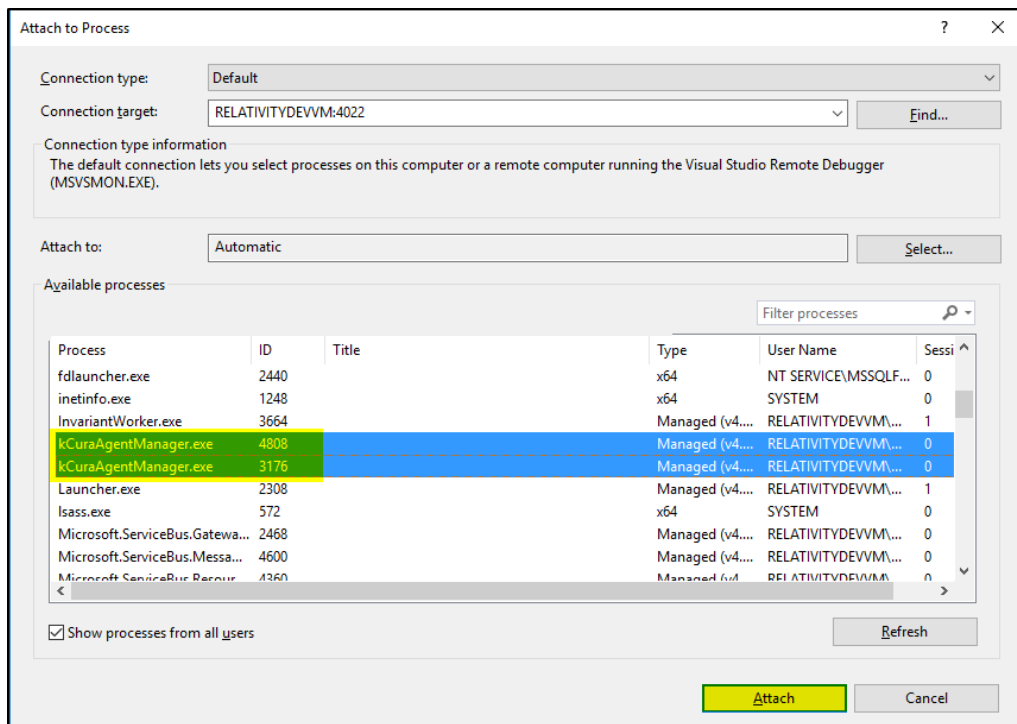
8. When your agent code gets executed on the Agent server, the breakpoint you set in your code will be hit as shown in the below screenshot.

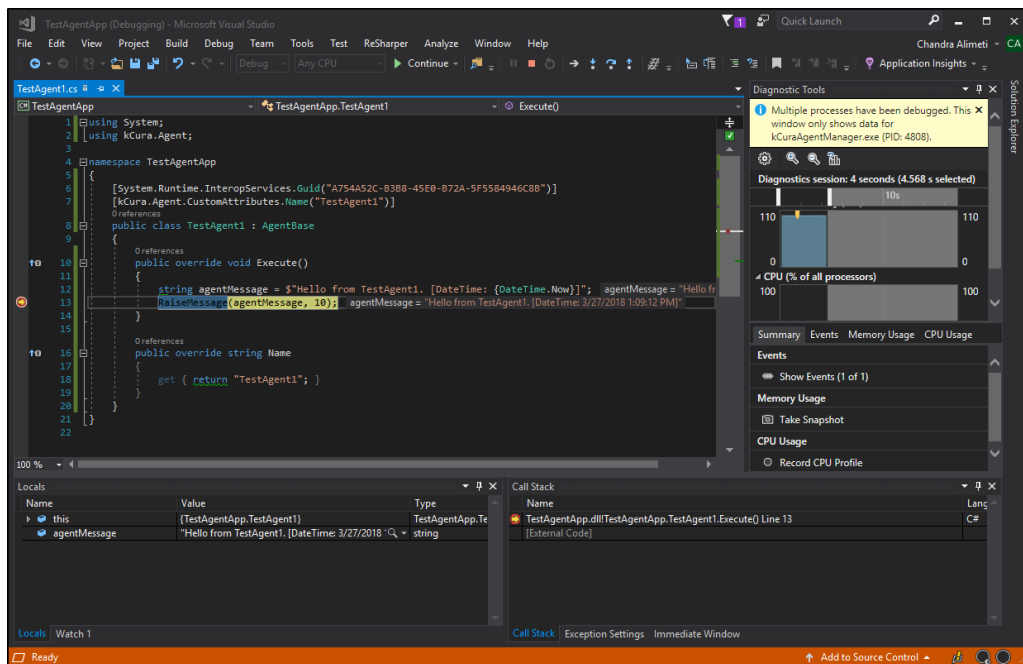*Note*: You can find more information on how to use remote debugging to troubleshoot agents, event handlers and custom pages at this link - https://platform.relativity.com/9.4/Content/Search.htm#search-remote%20debugging
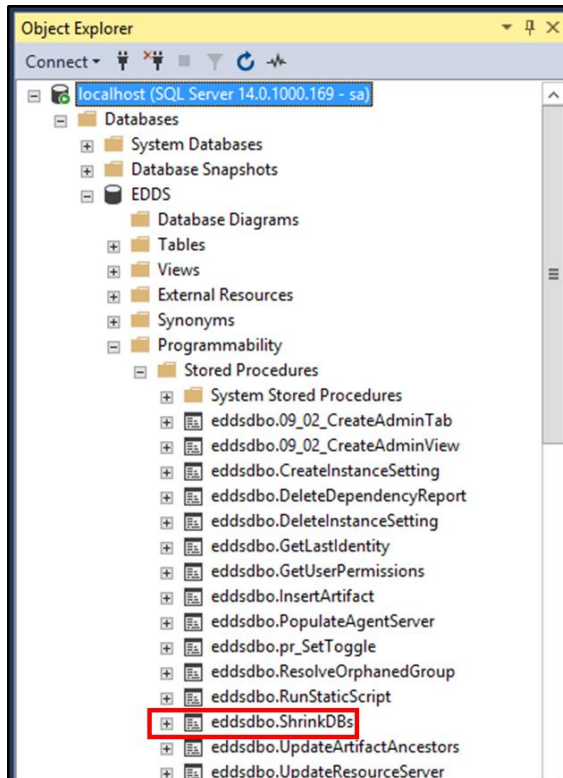
Running SQL Procedure to Shrink Databases and Set Recovery Model to Simple

1. Open SQL Server Management Studio
2. Log in using the following credentials:

    Login: **sa**

    Password: **Test1234!**

3. Check if EDDS → Programmability → Stored Procedures contains the ShrinkDBs stored procedure.

If it does run the following command in a new query:

COMMAND: `EXEC EDDS.eddsdbo.ShrinkDBs`

If it does not contain ShrinkDBs, run the following command in a new query:

COMMAND:

```
USE EDDS;

GO

IF (OBJECT_ID('tempdb..#databases') IS NOT NULL) DROP TABLE #databases;
SELECT      d.name as db_name
      INTO   #databases
FROM       sys.databases d
      WHERE   d.name LIKE 'EDDS%' OR d.name LIKE 'Invariant%'

DECLARE @db_name nvarchar(100) = N''

DECLARE cursor_mini CURSOR FAST_FORWARD FOR
      SELECT      d.db_name
      FROM         #databases d
OPEN cursor_mini
FETCH NEXT FROM cursor_mini INTO @db_name
```

```
    WHILE @@FETCH_STATUS = 0
    BEGIN
        DECLARE @script_01 nvarchar(max) = N'
            USE @@DB_NAME
            -- Truncate the log by changing the database recovery model to
SIMPLE.
            ALTER DATABASE @@DB_NAME
            SET RECOVERY SIMPLE;
        '
        SET @script_01 = REPLACE(@script_01, '@@DB_NAME', @db_name)
        --PRINT @script_01
        EXEC (@script_01)

        DECLARE @script_02 nvarchar(max) = N'
            USE @@DB_NAME
            -- Shrink the truncated log file to 1 GB.
            DBCC SHRINKFILE (@@DB_NAME, 1);
        '
        SET @script_02 = REPLACE(@script_02, '@@DB_NAME', @db_name)
        --PRINT @script_02
        EXEC (@script_02)

    FETCH NEXT FROM cursor_mini
    INTO @db_name
    END
    CLOSE cursor_mini
    DEALLOCATE cursor_mini
```

How to setup Processing on the DevVm

1. You first need to request a processing license from support@relativity.com and apply that license.
2. The Processing application already exists in the Application library on the DevVm. Install the application into a workspace and you should now be able to use processing in that workspace.

DevVM Tips

1. Instead of shutting down your DevVm, try saving it. This puts the VM in a "hibernated" state. When you start the DevVm after saving it, you will not have to start up the services again. It will save the state of your DevVm and will reduce time that it takes to set up the VM when you start it up. To save your DevVm:
    a. Right click on the VM in Hyper-V and select Save
2. Add checkpoints to your DevVm. In case something goes wrong on the VM, having checkpoints will save the state of your VM so you can go back to that checkpoint later. To add a checkpoint:
    a. Shut down (or save) your VM
    b. Right click on the VM and select Checkpoint