

Relativity Dev VM
Pre-built VM
Documentation
[April 3, 2018]

Table of Contents

Introduction	3
License:	3
System Requirements	3
Accounts	3
Windows Accounts	3
Relativity Accounts	3
SSMS (SQL) Accounts	4
Instructions for setting up a Dev VM	4
Updating Resources for VM	17
Remote Debugging with Visual Studio 2017	19

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.

License:

The Dev VM comes with the following licenses.

- Windows Server 2012 R2 Standard
 - 6-month trial license
- Relativity License
 - 3-month license
- Processing License
 - 3-month license
- SQL
 - SQL Server 2016 Developer edition

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

System Requirements

- Windows 10 Pro with Hyper V functionality
- 2 processing cores
- 8GB RAM
- 140GB free storage space.

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.

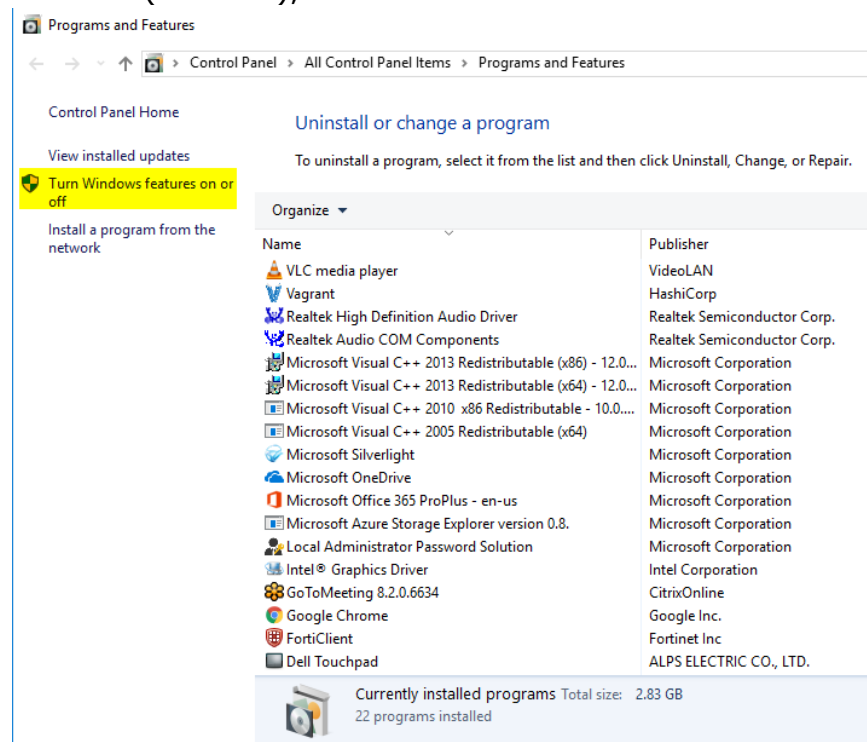
Login: **sa**

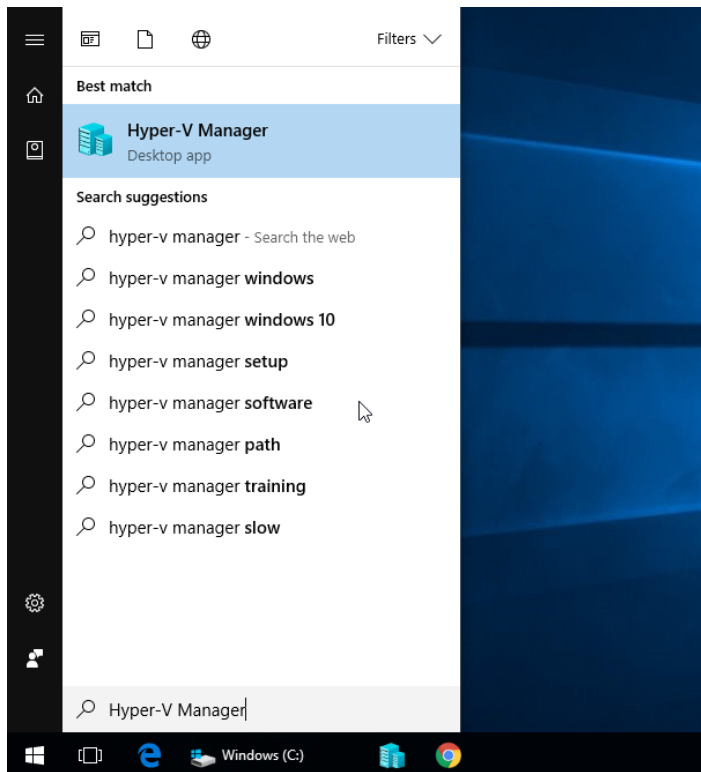
Password: **Test1234!**

Note: Use the **sa** account for any SQL Administrative tasks.

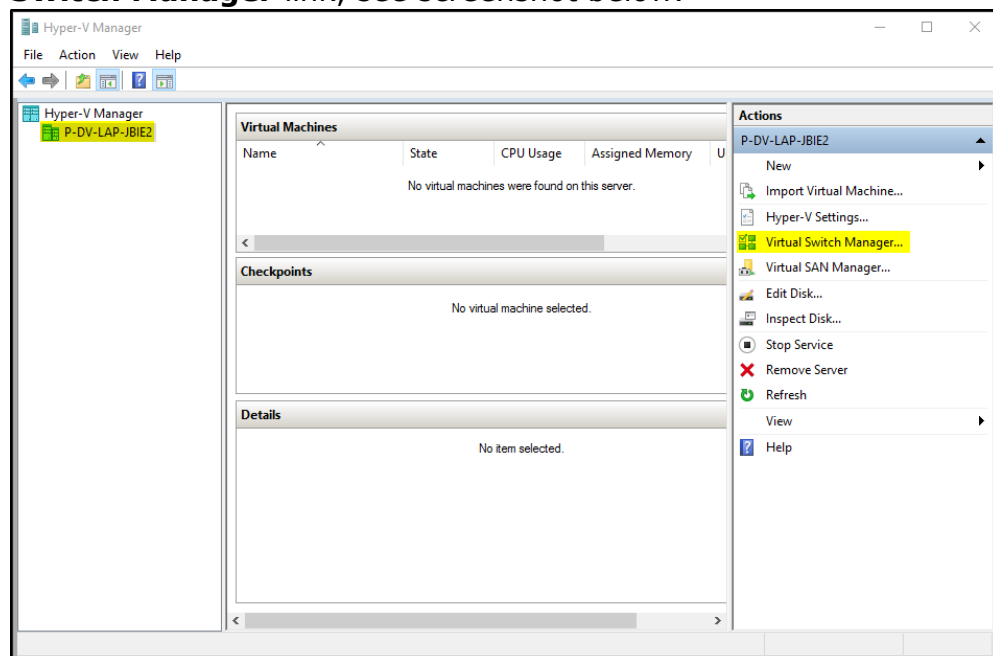
Instructions for setting up a Dev VM

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.

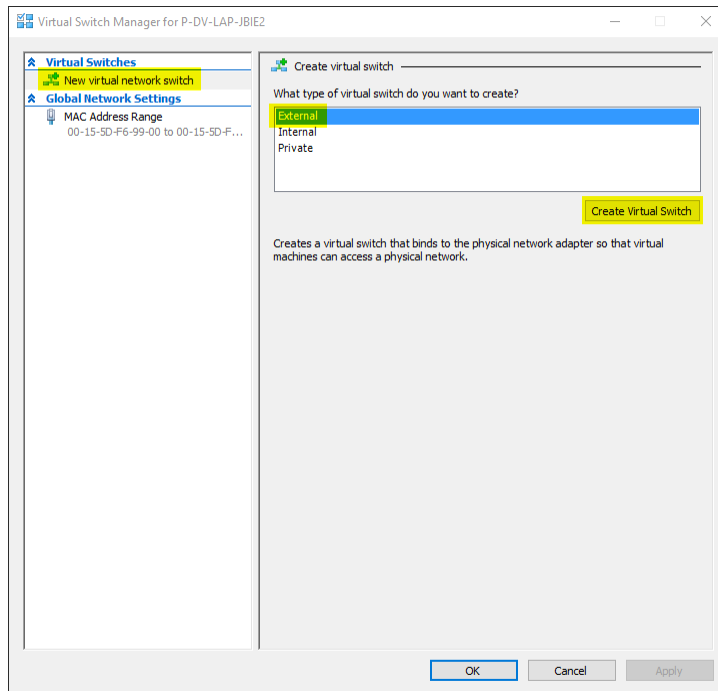




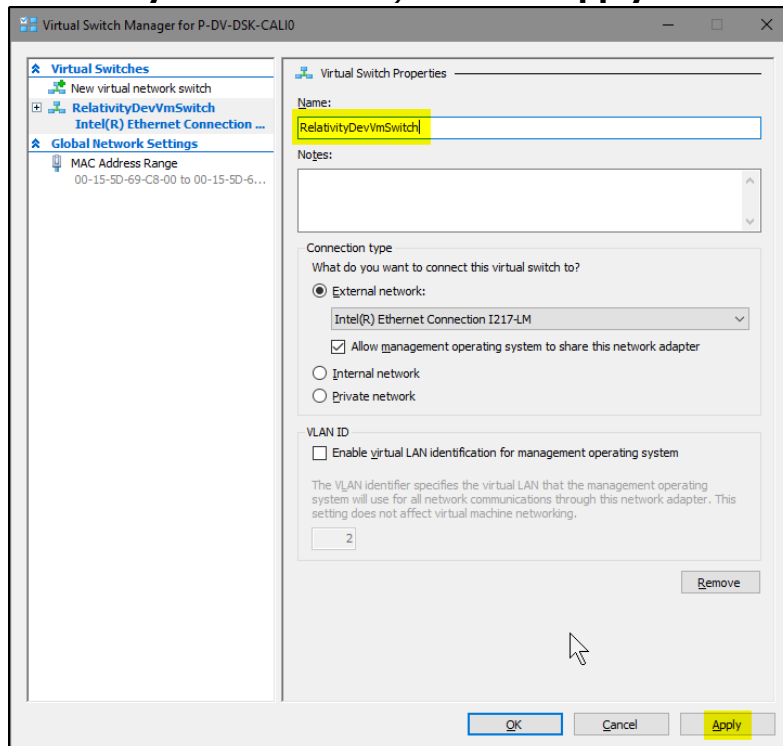
5. Make sure you have an **External Virtual Switch** which lets you access your VM from your local machine.
 - a. Select your local machine from the left pane.
 - b. In the Actions pane located in **Hyper-V Manager**, click on the **Virtual Switch Manager** link, see screenshot below.



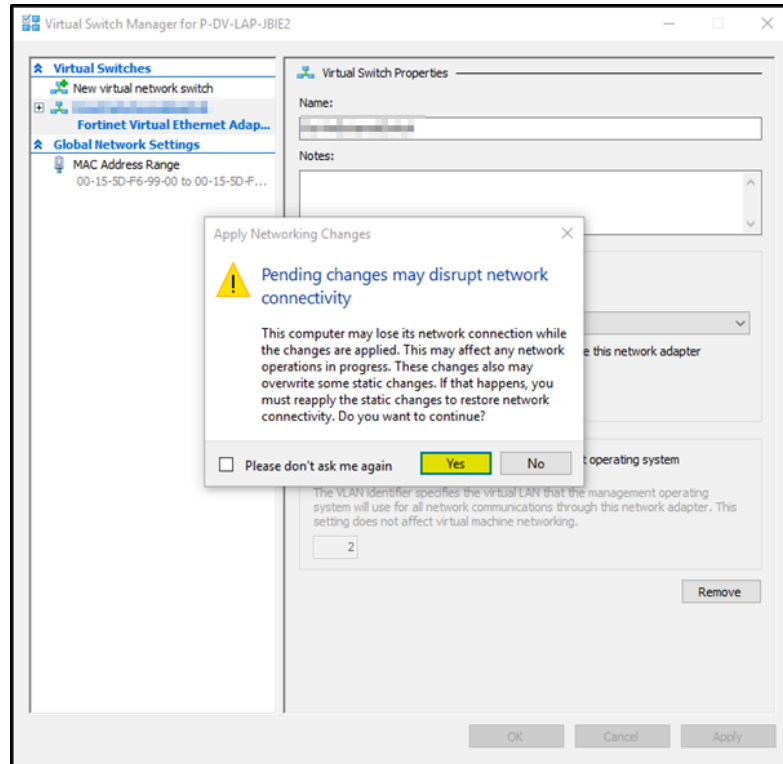
- c. If you don't already have an external network switch please click the **New virtual network switch** link under **Virtual Switches**. Select **External** for the type of switch and click on the **Create Virtual Switch** button.



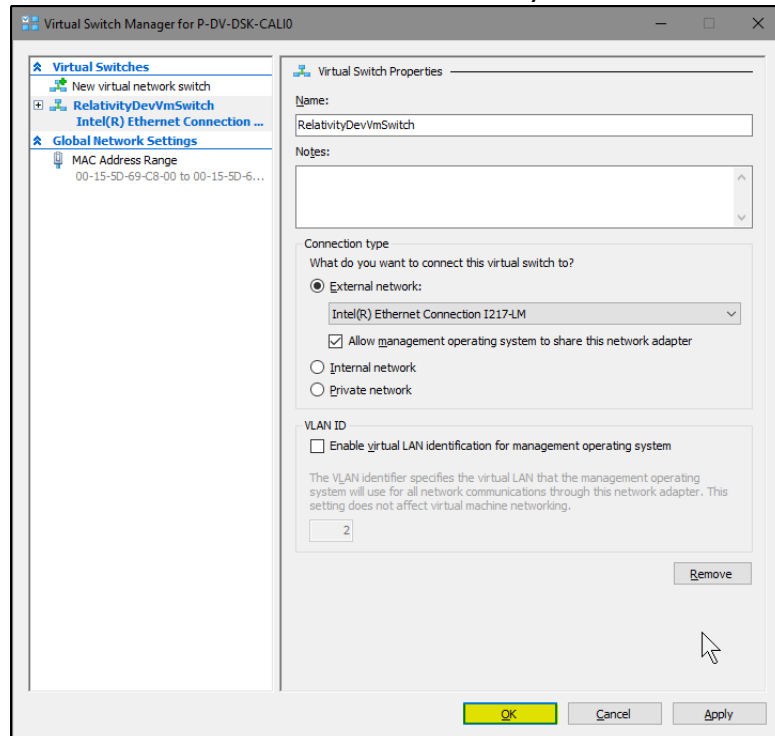
- d. Provide a friendly name for the new switch (For example: **RelativityDevVmSwitch**) and click **Apply** button.



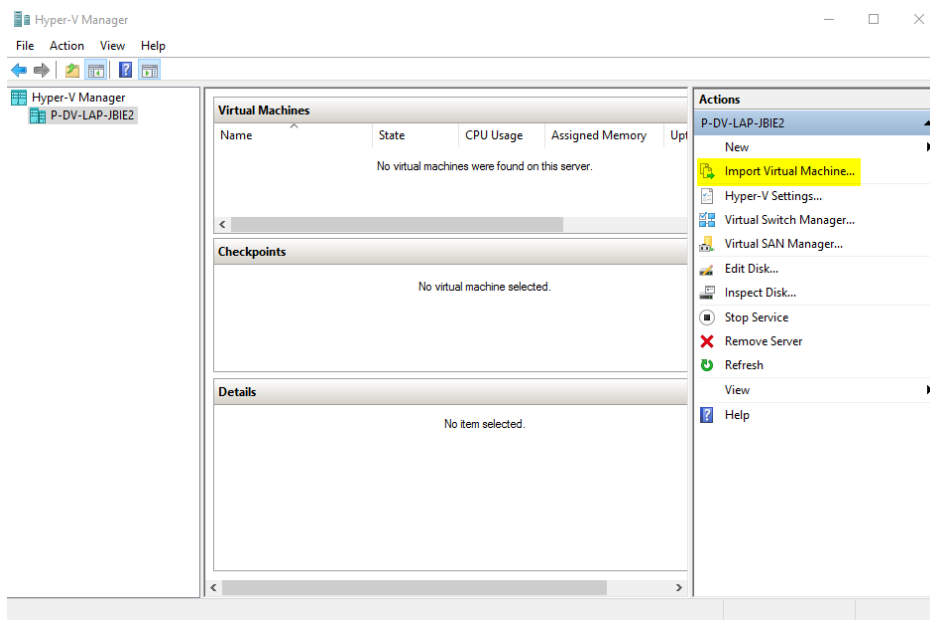
- e. If you get a warning message (see screenshot below), please click the **Yes** button.



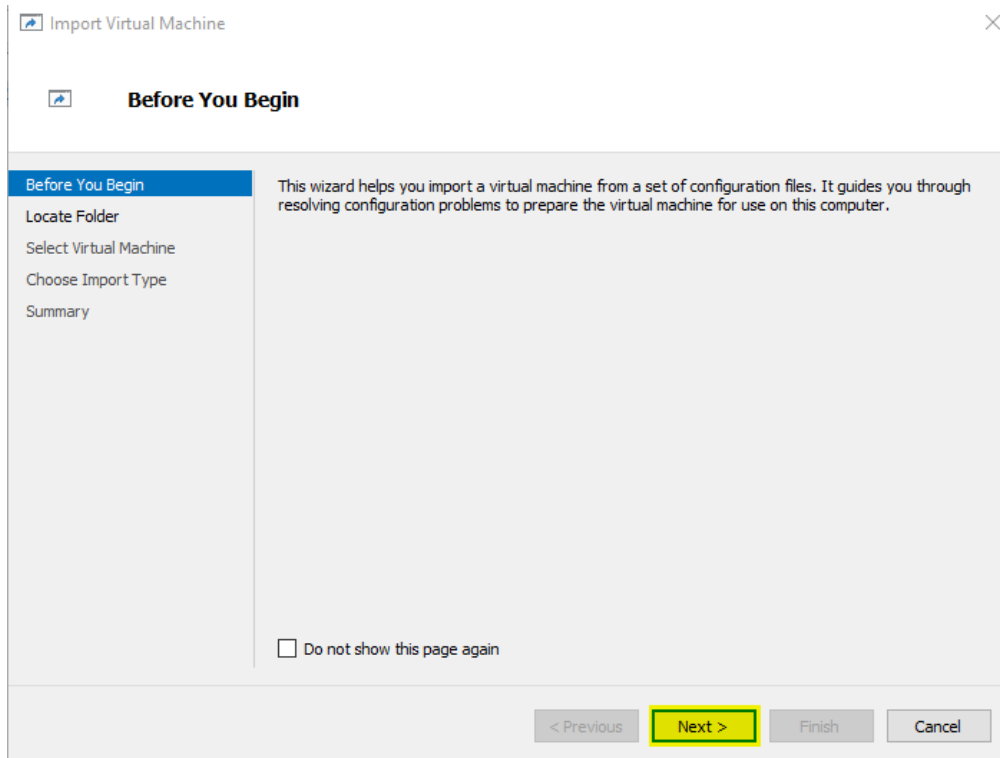
- f. Next click the **OK** button to create your new virtual switch.



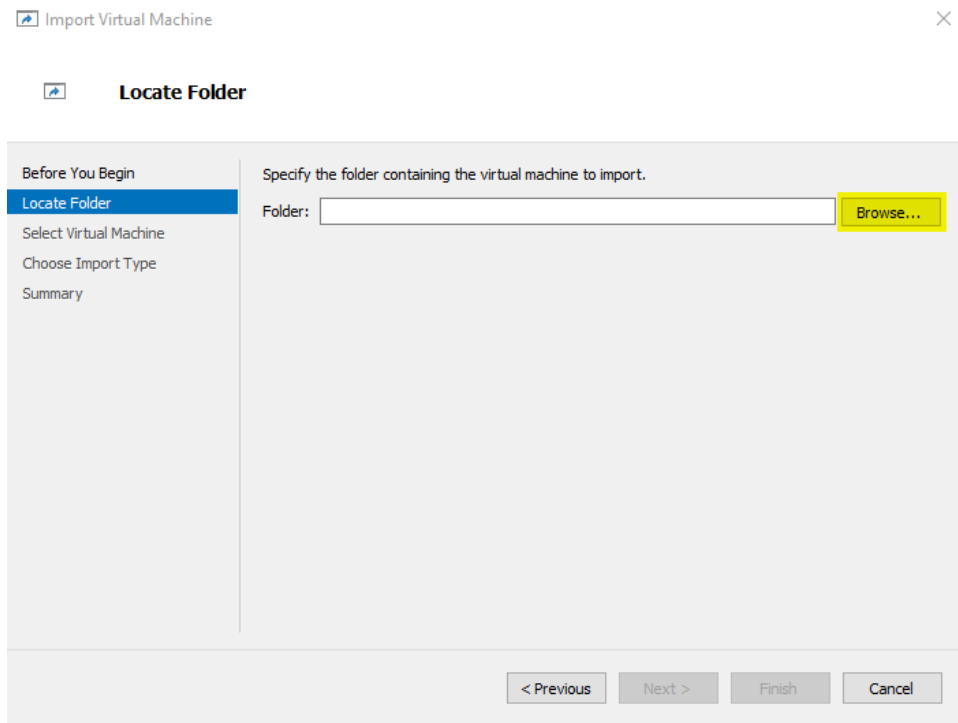
6. To import the Hyper-V VM, please click on the **Import Virtual Machine** link in the right-side pane located within **Hyper-V Manager**, see screenshot below.



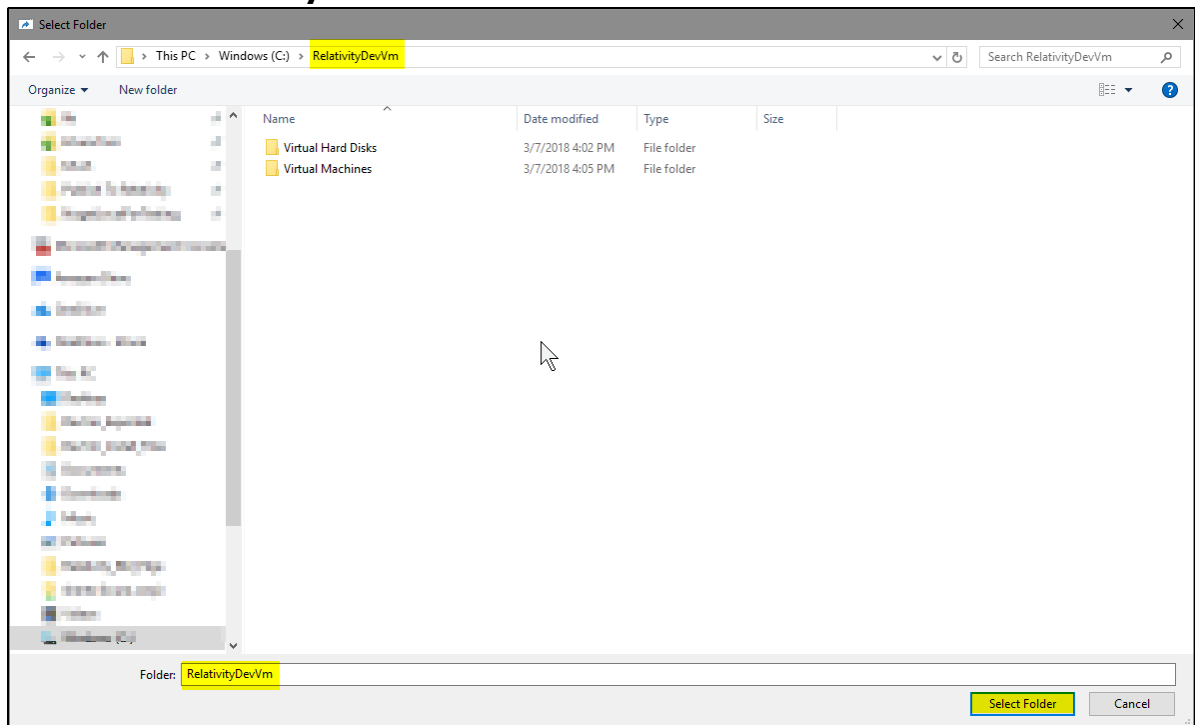
7. Click the **Next** button.



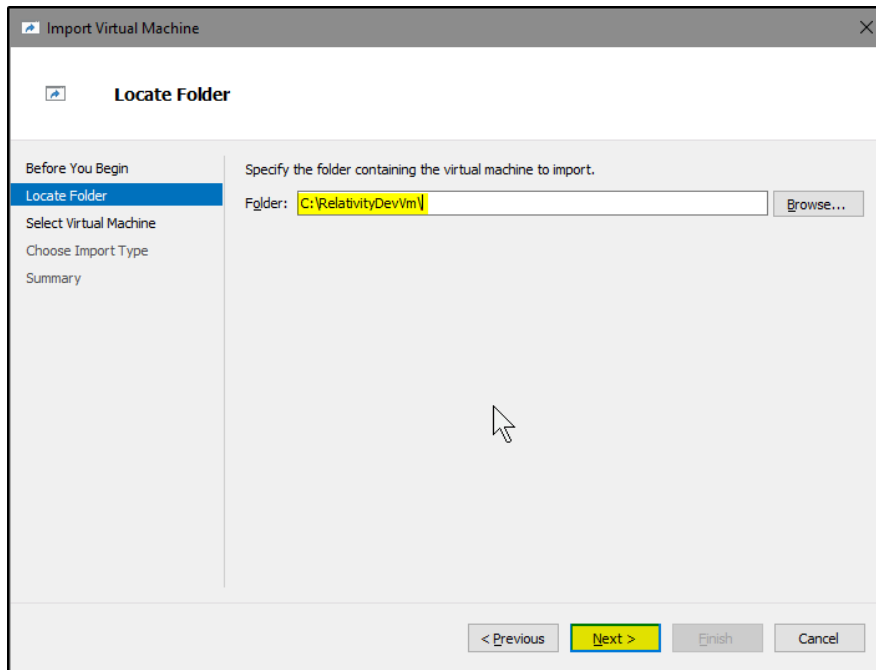
8. In the **Locate Folder** section, click the **Browse** to select the location where you copied the **RelativityDevVm** folder in step 3 located above.



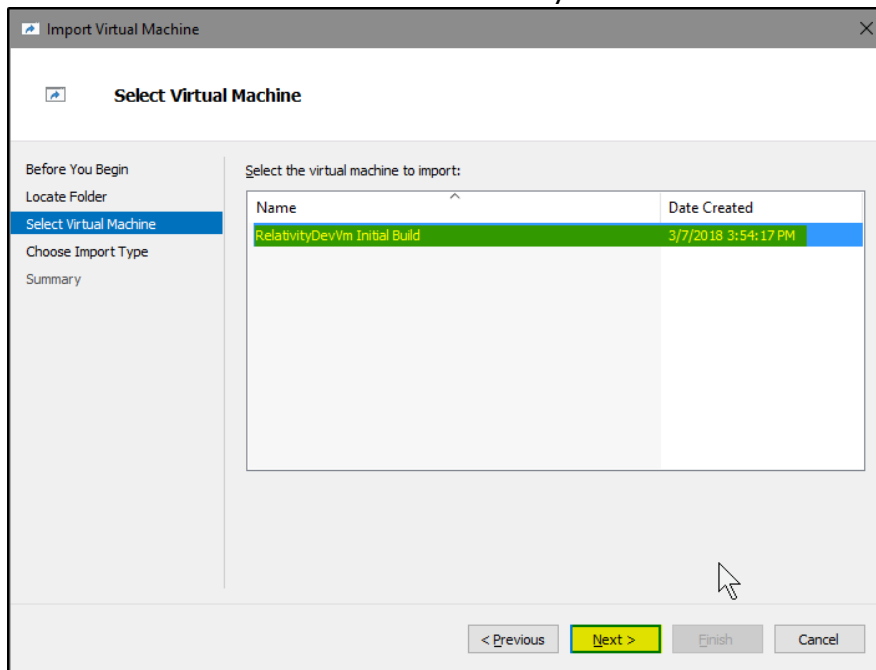
9. Select the **RelativityDevVm** folder and click the **Select Folder** button.



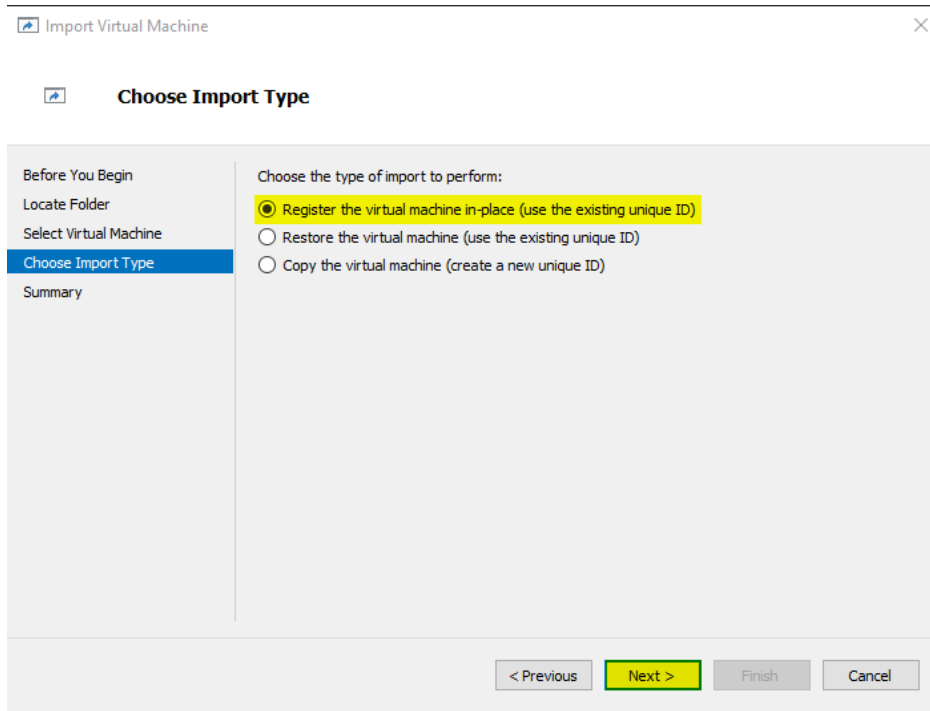
10. In the **Locate Folder** section, click the **Next** button.



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.

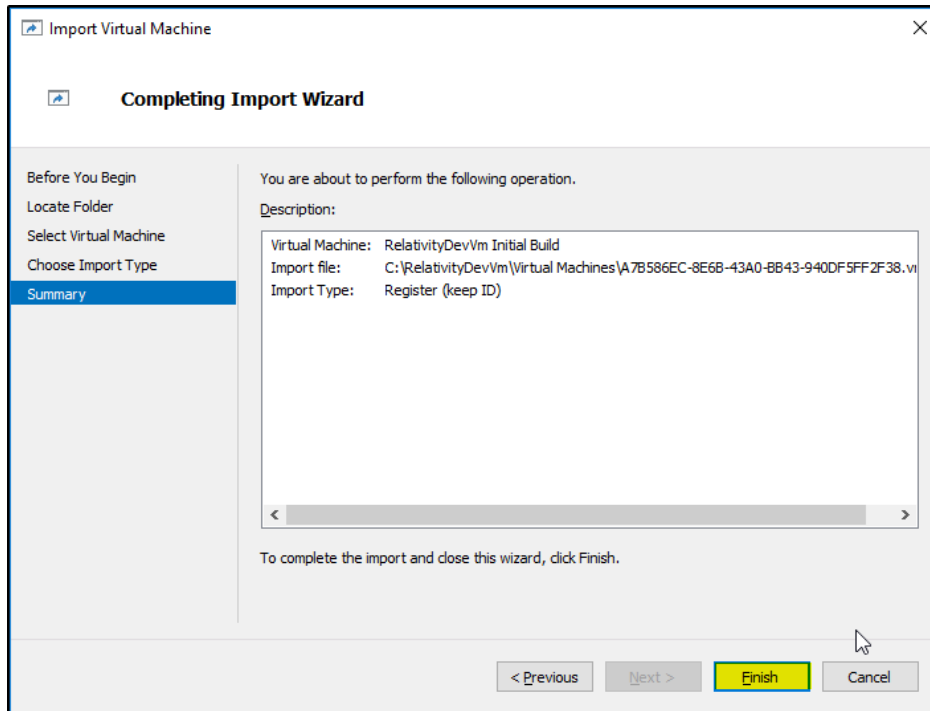


12. In the **Choose Import Type** section, choose the **Register the virtual machine in-place (use the existing unique ID)** option and click the **Next** button.

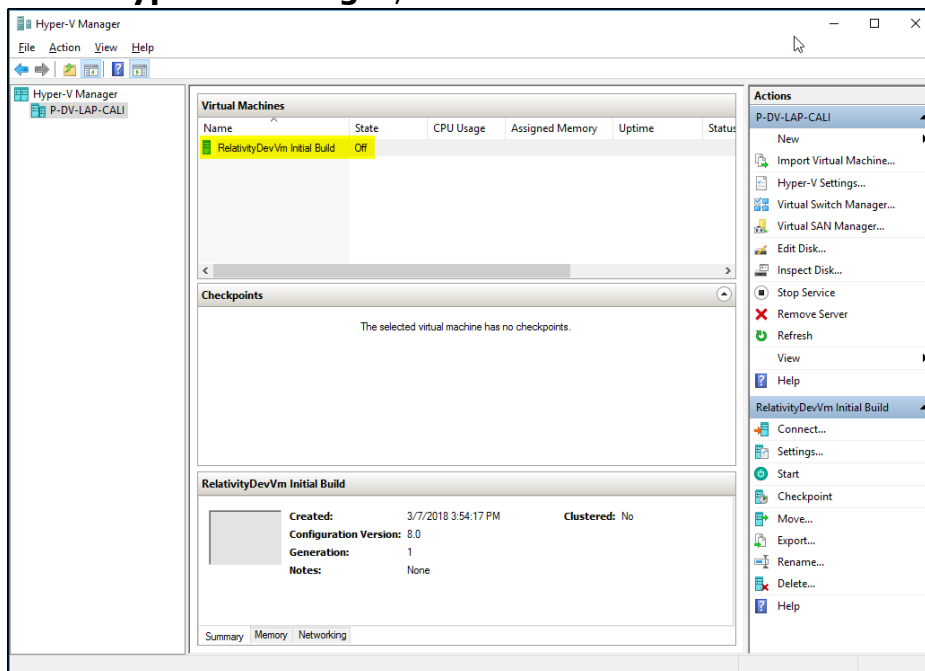


Note: In **Connect Network** section, if the DevVM is looking for a different virtual switch then you might get a warning. Please choose a valid virtual switch from the **Connection** dropdown list.

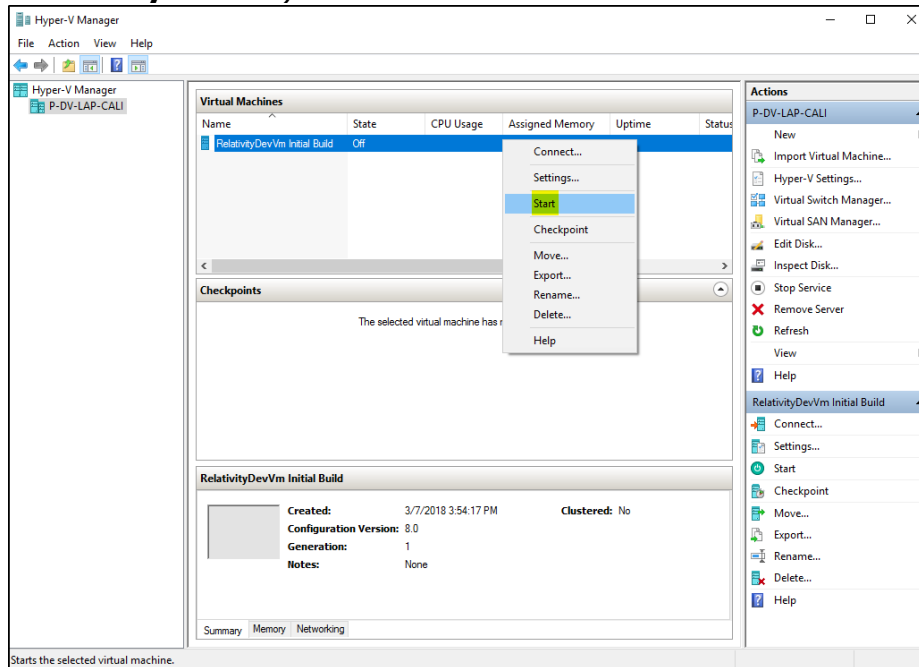
13. Next click the **Finish** button to complete the import of your new Hyper-V VM.



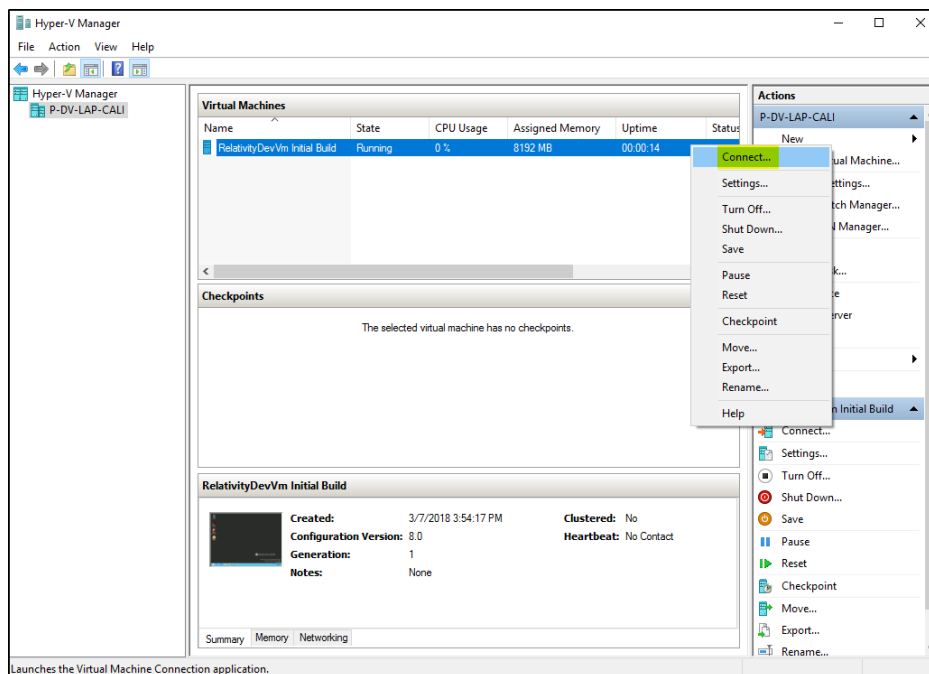
14. Now you should see the imported VM in the **Virtual Machines** section within **Hyper-V Manager**, see screenshot below.



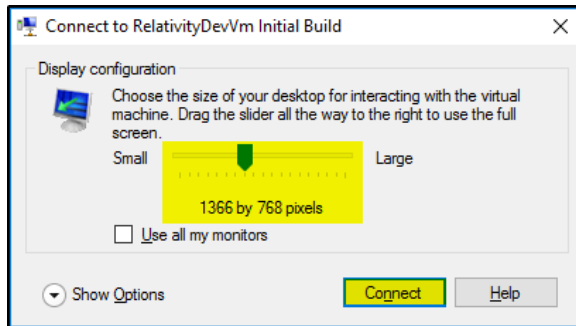
15. To start the Dev VM, right click on the VM instance (in the example below, **RelativityDevVm**) and select **Start**.



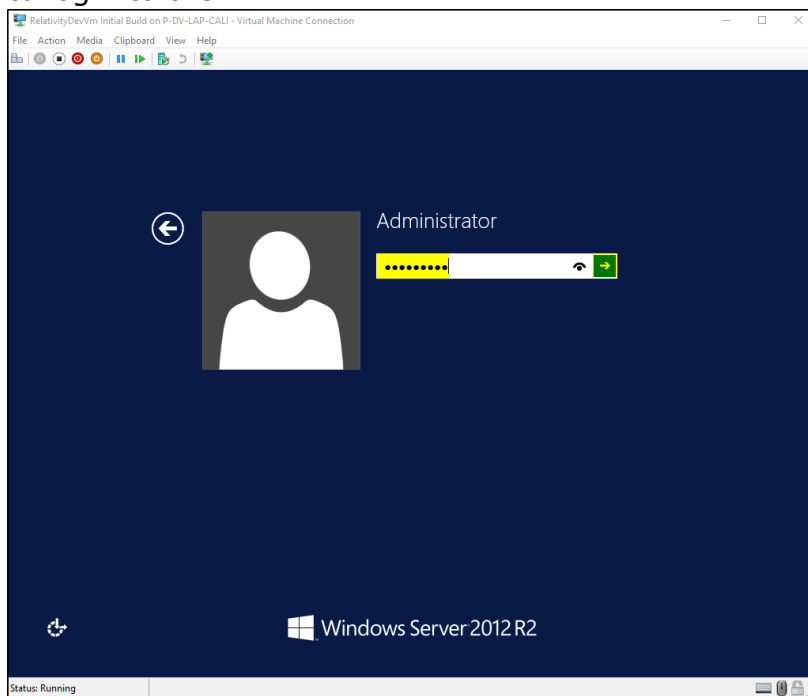
16. To connect to the Dev VM, right click on the VM instance and select **Connect**.



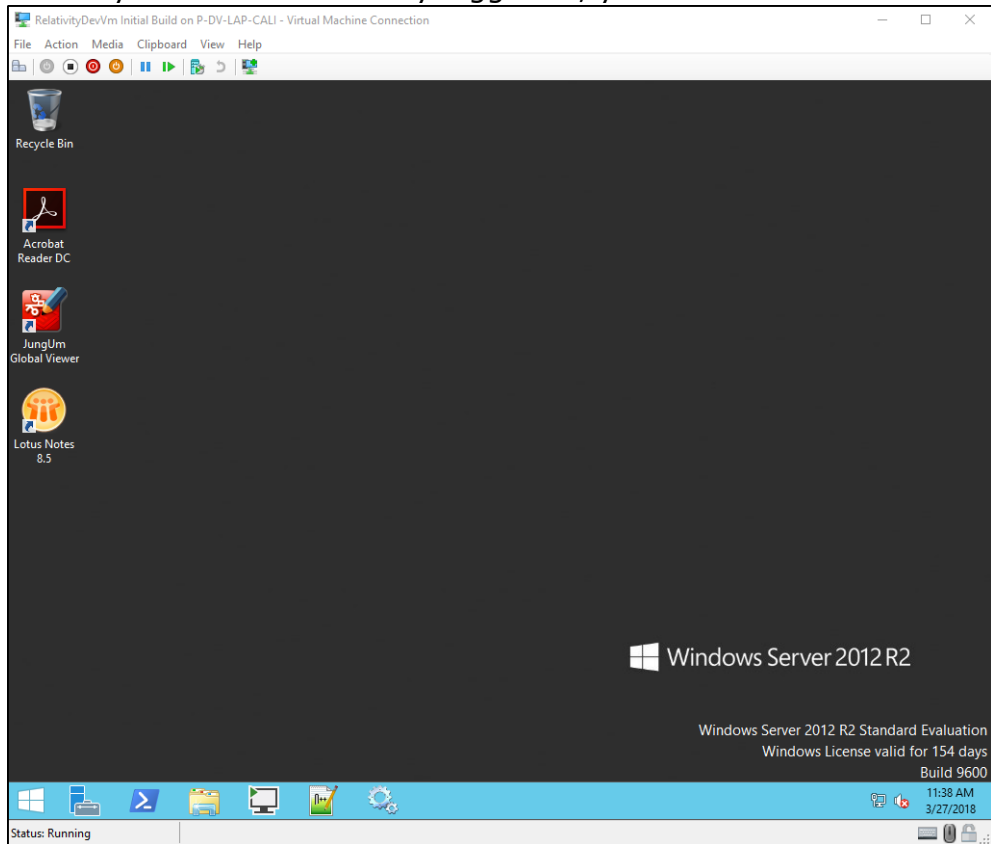
17. Select the appropriate resolution for your Dev VM and click the **Connect** button.



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



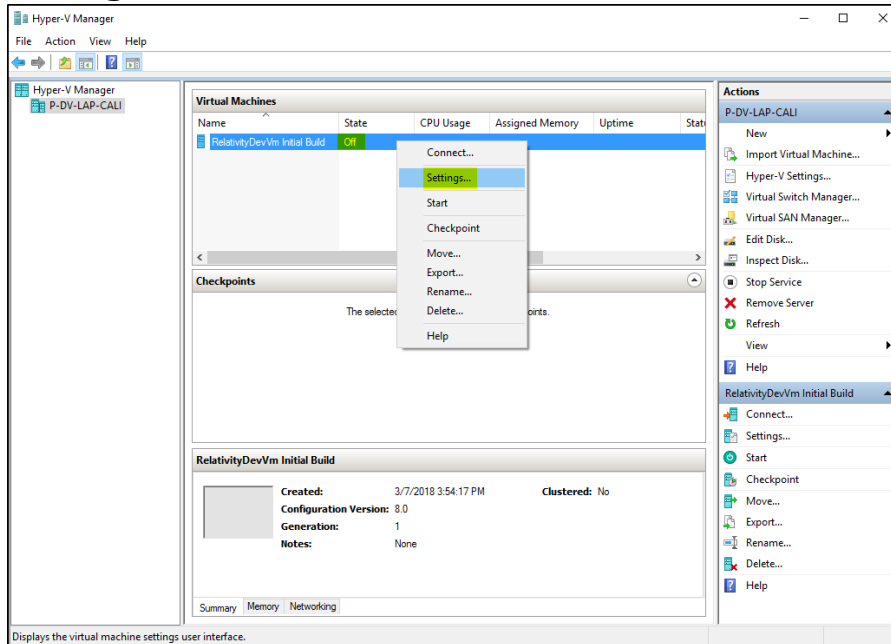
19. Once you have successfully logged in, you should see the Desktop screen.



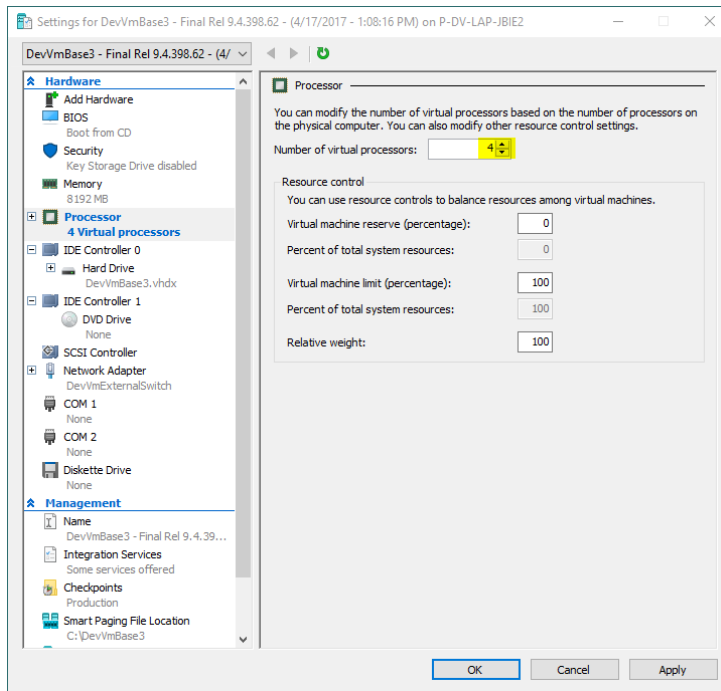
Updating Resources for VM

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.

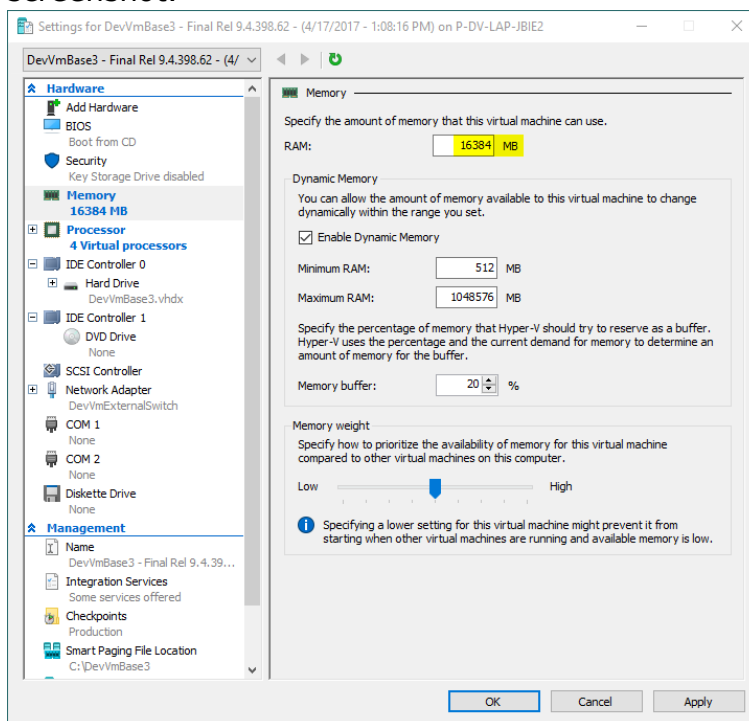
1. First make sure that the VM is shut down. Then right click the VM and select **Settings** as shown in below screenshot.



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.



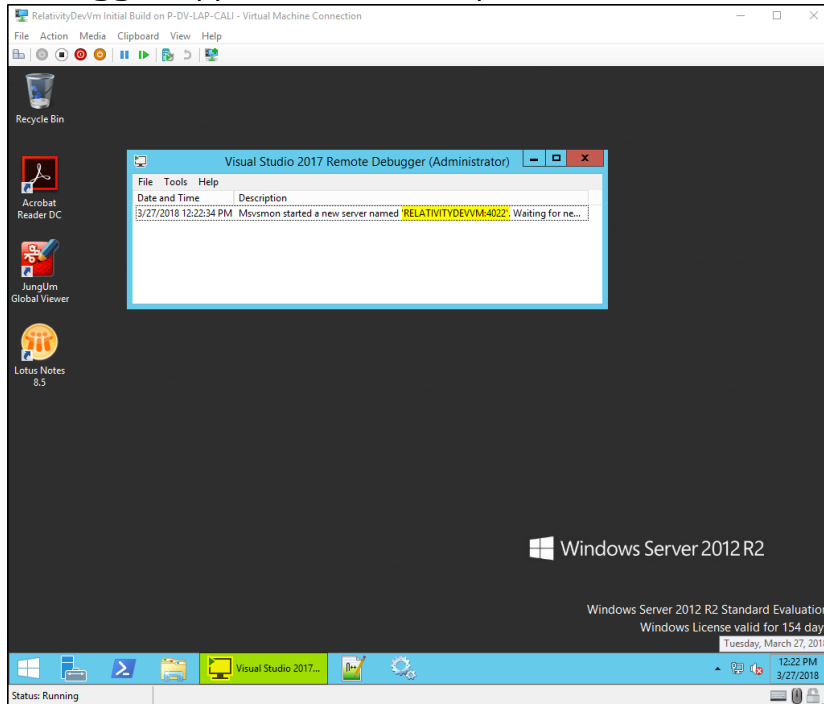
3. If you want to increase the RAM for the VM, click on the **Memory** link in the **Hardware** section and input the desired **RAM** for the VM as shown in below screenshot.



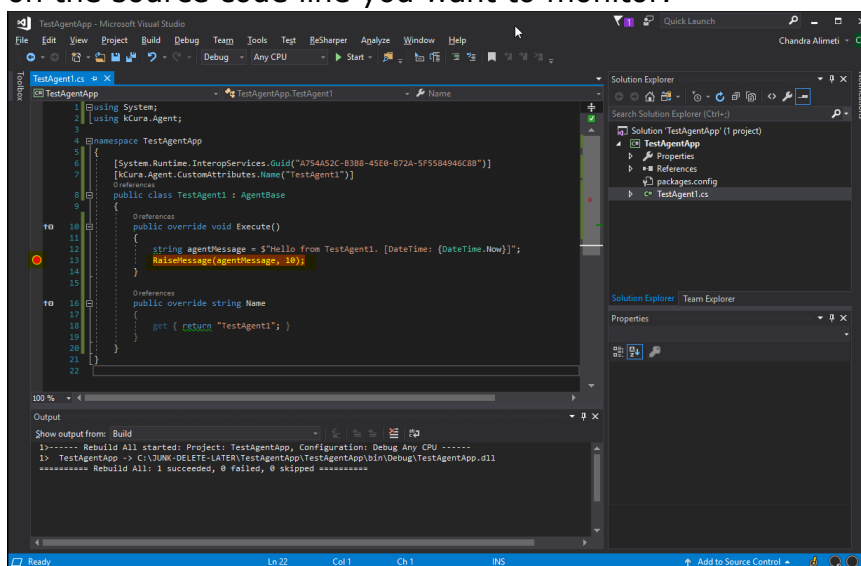
4. After you are done configuring the VM hardware, click the **Apply** button and then click the **OK** button.

Remote Debugging with Visual Studio 2017

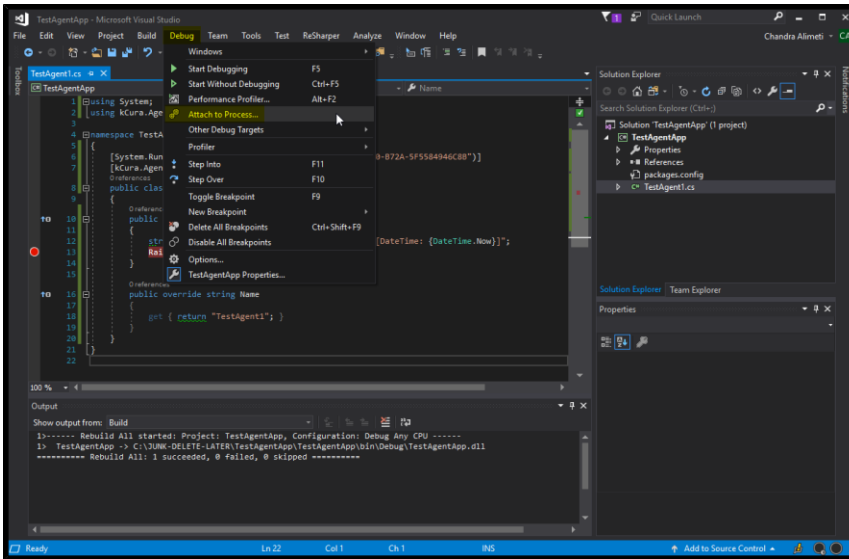
1. Remote into the Dev VM and launch **Visual Studio 2017 Remote Debugger** application which is pinned to **Taskbar**.



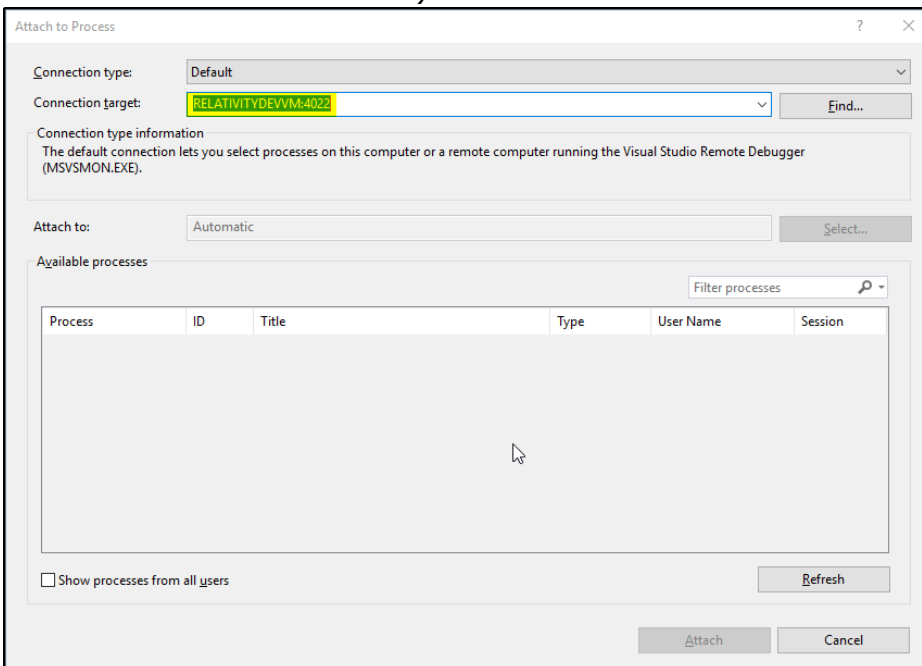
2. Open your visual studio project in **Visual Studio 2017** and add a breakpoint on the source code line you want to monitor.



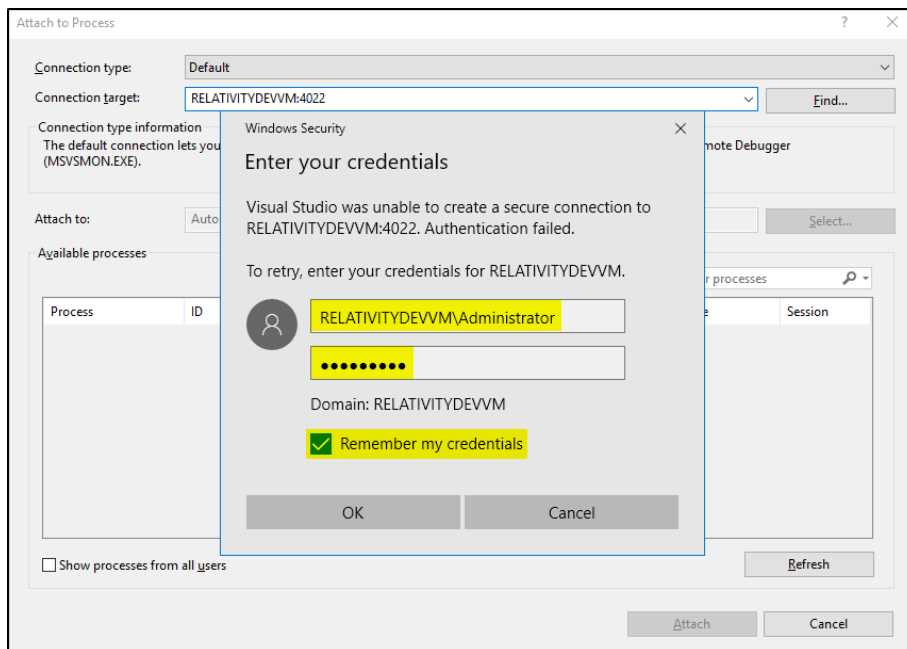
3. In **Visual Studio 2017**, go to **Debug** menu option and select **Attach to Process**.



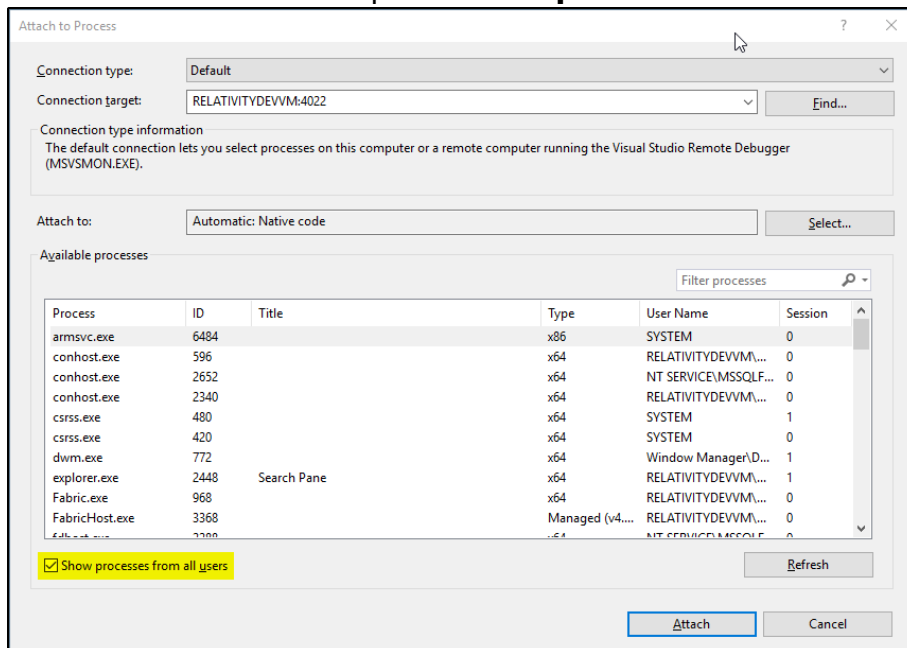
4. In the **Attach to Process** windows, enter your VM name along with port 4022 as shown in below screenshot and press **Enter**. (Example: **RELATIVITYDEVVM:4022**)



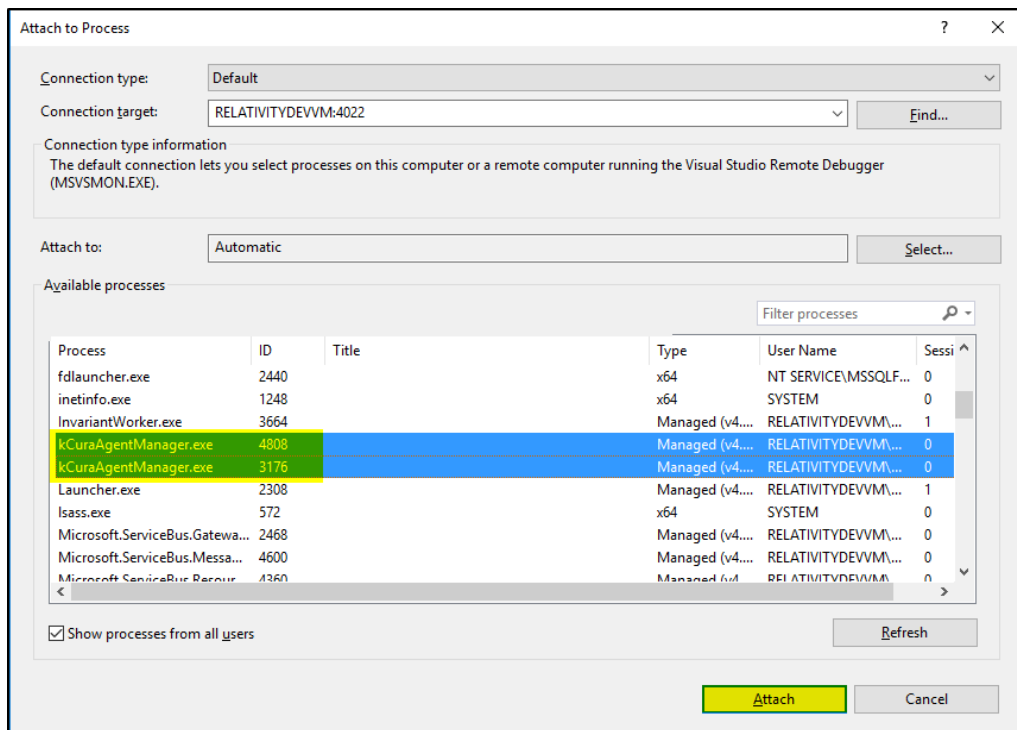
5. If prompted for credentials, enter the **RelativityService** credentials provided in the windows account section of this document.



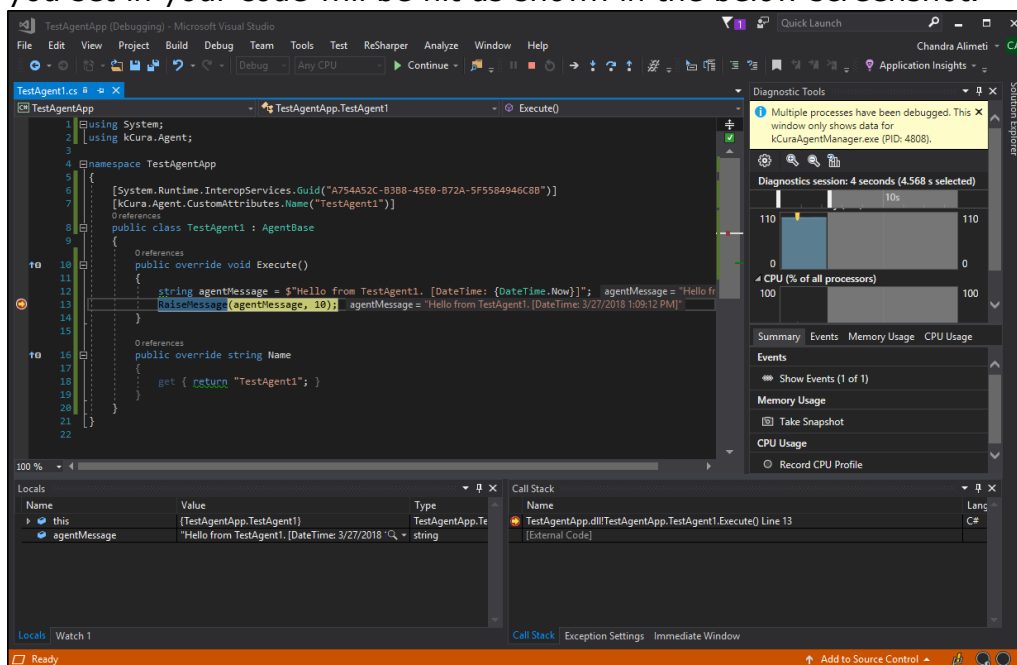
- Next check the box for option **Show processes from all users**.



- 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.



- 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>