

Relativity Dev VM  
Running Vagrant and Chef Scripts  
Documentation  
[October 16, 2019]

# Table of Contents

1	Requirements.....	3
2	Downloads .....	3
2.1.1	Relativity installation.....	4
2.1.2	SQL Server 2016 Developer Edition .....	4
2.1.3	Service Bus 1.1 Defect Windows Update .....	4
2.1.4	Install Visual Studio Code text editor .....	5
3	Setup Workstation .....	5
3.1	Login.....	5
3.2	Setup PowerShell .....	5
3.3	Install Chocolatey.....	7
3.4	Install GIT Version 2.14.1 .....	7
3.5	Clone Dev VM repository from Github .....	8
3.6	Install Chef Development Kit Version 2.4.17 .....	8
3.7	Enable the Hyper-V Windows feature .....	9
3.8	Setup Hyper-V Virtual Switch.....	12
3.9	Share Internet Connection.....	15
3.10	Install Vagrant Version 2.0.1.....	17
3.11	Install Vagrant Reboot plugin.....	17
3.12	Install Vagrant berkshelf plugin .....	18
3.13	Add Dev VM Windows Base machine to Vagrant boxes list .....	18
4	Create New Dev VM.....	19
4.1	Login.....	19

4.2	Navigate to the Dev VM GIT repository .....	19
4.3	Switch to vagrant-poc branch .....	19
4.4	Copy the config.yaml file required for Dev VM .....	19
4.5	Update Configuration file.....	22
4.6	Run the Dev VM Creation script .....	22
4.6.1	Navigate to the Dev VM GIT repository .....	22
4.6.2	List all the files in Chef folder .....	23
4.6.3	Run the Creation script .....	23
5	Clean Workstation .....	24
5.1	Login.....	24
5.2	Remove Vagrant Box Image .....	24
5.3	Uninstall Vagrant berkshelf plugin.....	24
5.4	Uninstall Vagrant reload plugin .....	24
5.5	Uninstall Vagrant.....	24
5.6	Uninstall Chocolatey .....	24
5.7	Remove Hyper-V Virtual Switch.....	25
5.8	Uninstall Hyper-V .....	25
5.9	Uninstall Chef Development Kit.....	25

## 1 Requirements

- CPU cores: 4 cores
- RAM: 12GB
- Storage: 140GB
- Licenses
  - Windows Server 2012 R2
  - Relativity
- Windows 10 Pro with Hyper-V features enabled.
- Windows Admin account.
- Internet connection

## 2 Downloads

- List of files needed for running the Vagrant/Chef recipes.
  - Relativity installer
  - Invariant installer

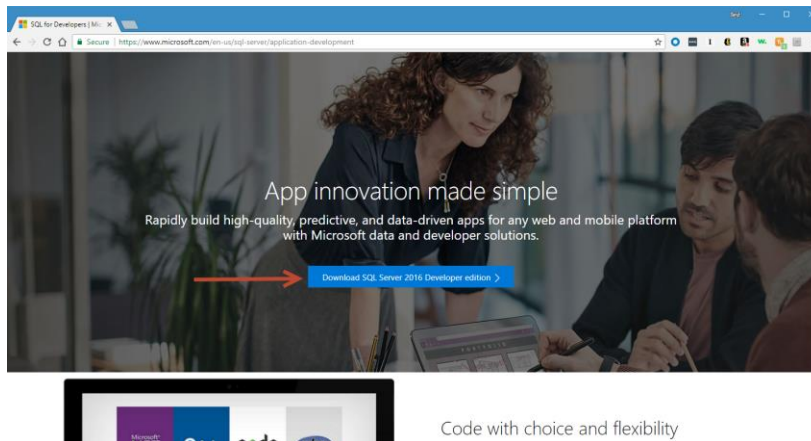
- Secret Store
- Datagrid
- SQL Server 2016 Developer Edition
- Service Bus Defect Windows update
- Visual Studio Code text editor
- MS Office Professional Plus 2010
- MS Works Converter
- Lotus Notes
- Jungum

### 2.1.1 Relativity installation

- Please contact [support@relativity.com](mailto:support@relativity.com) to get the installer files for the following:
  - Relativity
  - Invariant
  - Secret Store
  - Datagrid

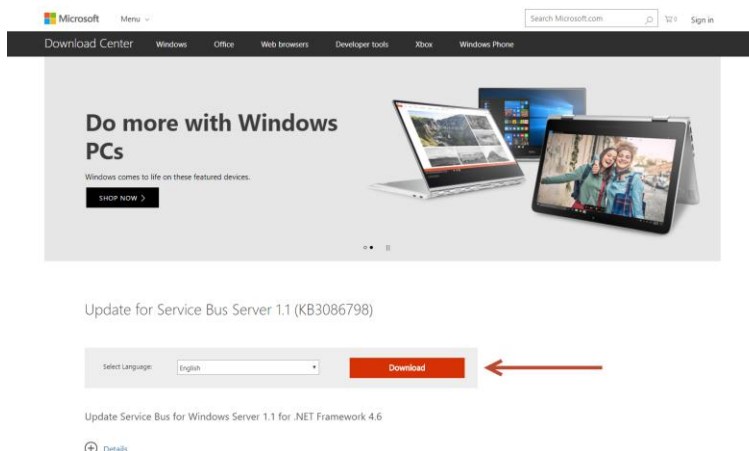
### 2.1.2 SQL Server 2016 Developer Edition

- You can download it at this link - <https://www.microsoft.com/en-us/sql-server/application-development>



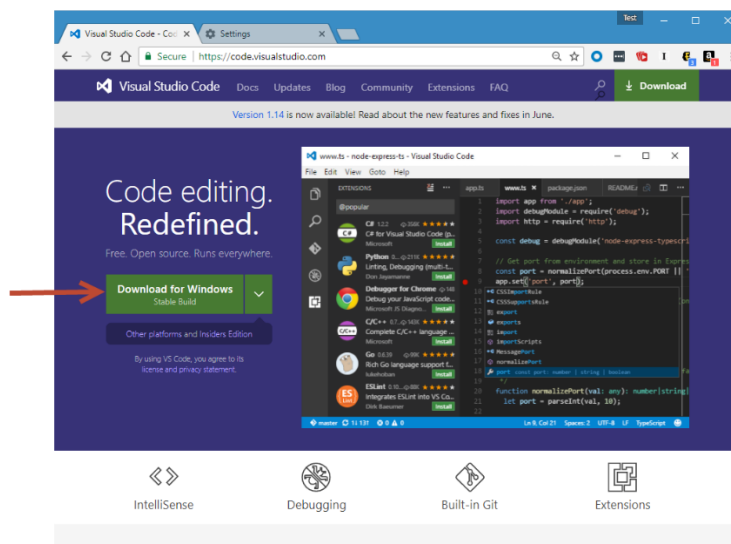
### 2.1.3 Service Bus 1.1 Defect Windows Update

- You can download it at this link - <https://www.microsoft.com/en-us/download/details.aspx?id=49496>



## 2.1.4 Install Visual Studio Code text editor

- Download and Install Visual Studio Code from this link [Visual Studio Code](https://code.visualstudio.com)



## 3 Setup Workstation

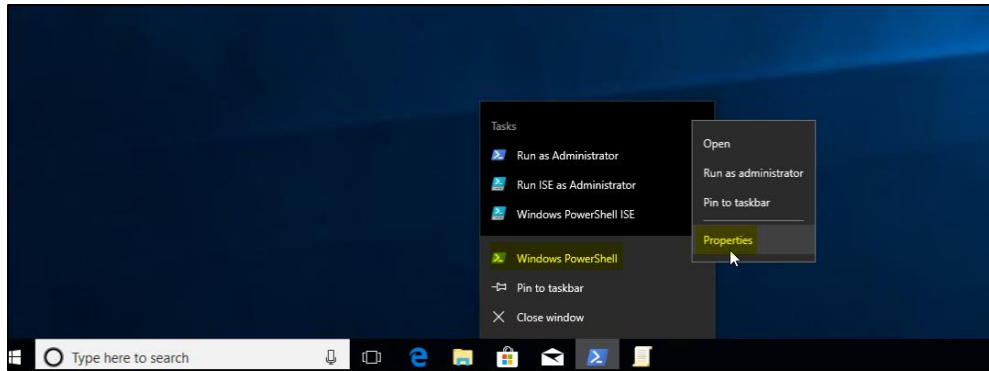
- It's important that you run the below steps in order.

### 3.1 Login

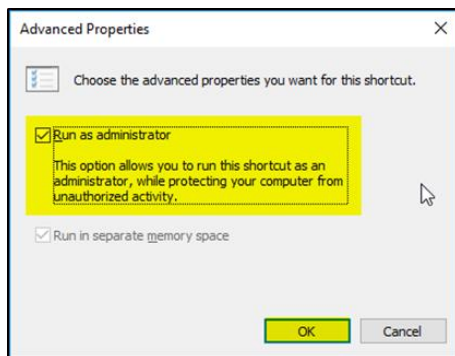
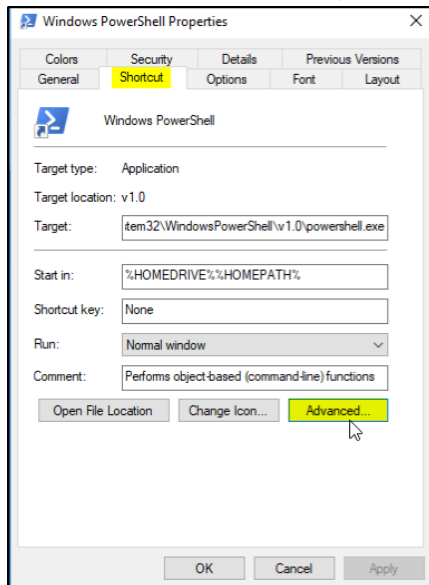
- Login into your workstation with a Windows Administrator account.

### 3.2 Setup PowerShell

- Let's configure PowerShell to always open as an Administrator.
- Pin **PowerShell** to your Task Bar
- Right click on **Windows PowerShell** then **Properties**



- Find the tab called **Shortcut**, then **Advanced**, then **Run As Administrator**



- Open a PowerShell window (should be opening as an Administrator now) and run the following command:

Set-ExecutionPolicy Unrestricted

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Set-ExecutionPolicy Unrestricted
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
http://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
PS C:\WINDOWS\system32>
```

### 3.3 Install Chocolatey

- Run the following command in PowerShell window.

Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
Getting latest version of the Chocolatey package for download.
Getting Chocolatey from https://chocolatey.org/api/v2/package/chocolatey/0.10.9.
Downloading 7-Zip commandline tool prior to extraction.
Extracting C:\Users\CHANDR~1.ALI\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\Users\CHANDR~1.ALI\AppData\Local\Temp\chocolatey\chocInstall...
Installing chocolatey on this machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
Creating Chocolatey folders if they do not already exist.
WARNING: You can safely ignore errors related to missing log files when
upgrading from a version of Chocolatey less than 0.9.9.
'Batch file could not be found' is also safe to ignore.
'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\chandra.alimeti\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\WINDOWS\system32>
```

### 3.4 Install GIT Version 2.14.1

- Run the following command in PowerShell window.

choco install git --version 2.14.1

```
Select Administrator: Windows PowerShell
PS C:\WINDOWS\system32> choco install git --version 2.14.1
Chocolatey v0.10.9
Installing the following packages:
git
By installing you accept licenses for the packages.
Progress: Downloading git.install 2.14.1... 100%
Progress: Downloading chocolatey-core.extension 1.3.3... 100%
Progress: Downloading git 2.14.1... 100%

chocolatey-core.extension v1.3.3 [Approved]
chocolatey-core.extension package files install completed. Performing other installation steps.
Installed/updated chocolatey-core extensions.
The install of chocolatey-core.extension was successful.
Software installed to 'C:\ProgramData\chocolatey\extensions\chocolatey-core'

git.install v2.14.1 [Approved]
git.install package files install completed. Performing other installation steps.
The package git.install wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[N]o/[P]rint): y

Using Git LFS
Installing 64-bit git.install...
git.install has been installed.
git.install installed to 'C:\Program Files\Git'
git.install can be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
The install of git.install was successful.
Software installed to 'C:\Program Files\Git\'

git v2.14.1 [Approved]
git package files install completed. Performing other installation steps.
The install of git was successful.
Software install location not explicitly set, could be in package or
default install location if installer.

Chocolatey installed 3/3 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32>
```

### 3.5 Clone Dev VM repository from Github

- Run the follo
- wing command in PowerShell window.

cd "C:\"

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> cd "C:\"
PS C:\>
```

- Run the following command in PowerShell window.

git clone https://github.com/relativitydev/relativity-dev-vm.git

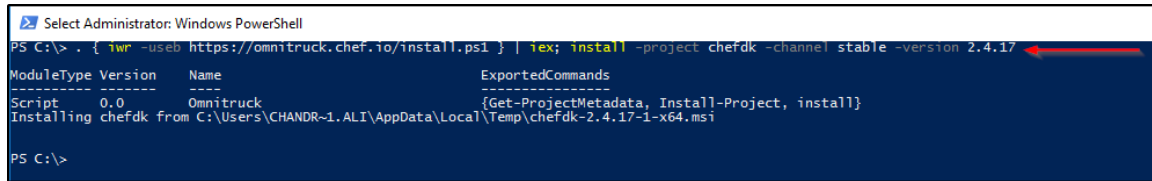
```
Administrator: Windows PowerShell
PS C:\> git clone https://github.com/relativitydev/relativity-dev-vm.git
Cloning into 'relativity-dev-vm'...
remote: Counting objects: 1238, done.
remote: Compressing objects: 100% (196/196), done.
remote: Total 1238 (delta 227), reused 285 (delta 141), pack-reused 848
Receiving objects: 100% (1238/1238), 37.74 MiB | 3.45 MiB/s, done.
Resolving deltas: 100% (744/744), done.
PS C:\>
```

### 3.6 Install Chef Development Kit Version 2.4.17

- Run the following command in PowerShell window.



```
. { iwr -useb https://omnitruck.chef.io/install.ps1 } | iex; install -project chefdk -channel stable -version 2.4.17
```

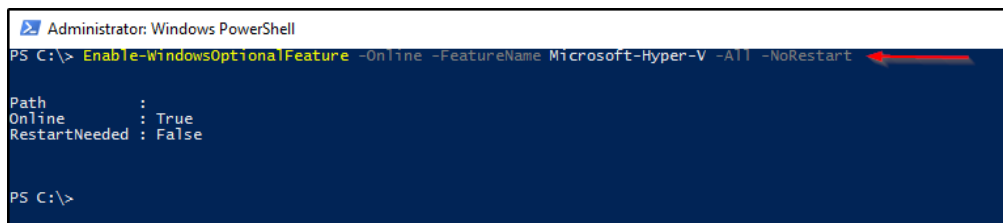


```
Select Administrator: Windows PowerShell
PS C:\> . { iwr -useb https://omnitruck.chef.io/install.ps1 } | iex; install -project chefdk -channel stable -version 2.4.17
ModuleType Version Name ExportedCommands
-----
Script 0.0 Omnitruck {Get-ProjectMetadata, Install-Project, install}
Installing chefdk from C:\Users\CHANDR~1.ALI\AppData\Local\Temp\chefdk-2.4.17-1-x64.msi
PS C:\>
```

### 3.7 Enable the Hyper-V Windows feature

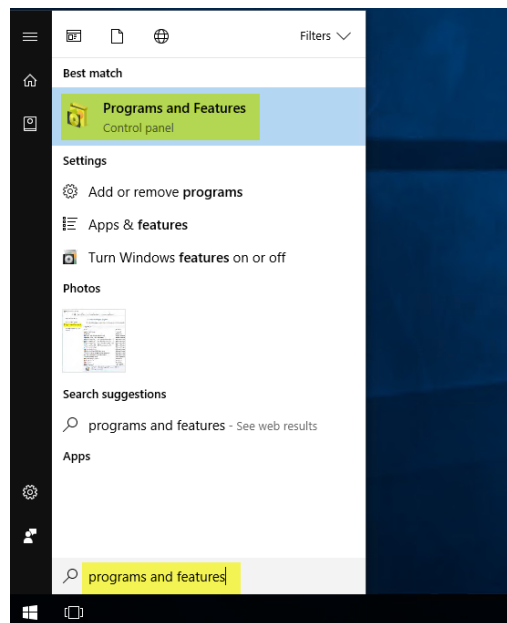
- Run the following command in PowerShell window.

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All -NoRestart
```



```
Administrator: Windows PowerShell
PS C:\> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All -NoRestart
Path :
Online : True
RestartNeeded : False
PS C:\>
```

- **Restart your workstation.**
- Verify Windows Hyper-V feature installation
  - Once your workstation finished restart, Go to **Programs and Features**



- Click on **Turn Windows features on or off**

Programs and Features

Control Panel

All Control Panel Items

Programs and Features

Control Panel Home

View installed updates

Turn Windows features on or off

Install a program from the network

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Rep

Organize

Uninstall/Change

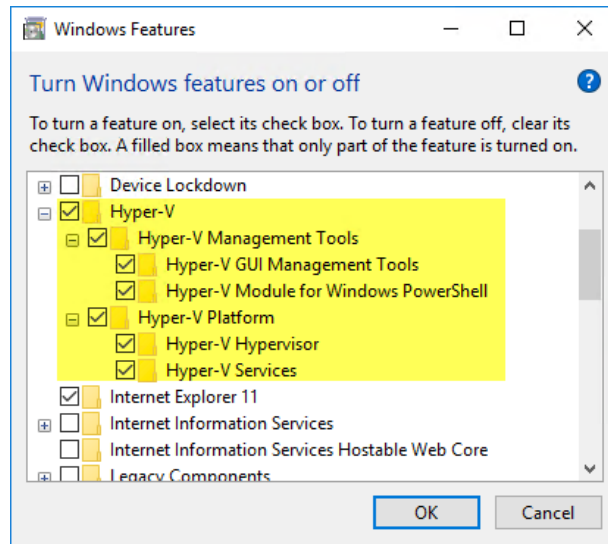
Name	Publisher
Microsoft .NET Version Manager (x64) 1.0.0-beta5	Microsoft Corporation
Microsoft .NET Framework 4.6.1 Targeting Pack (ENU)	Microsoft Corporation
Microsoft .NET Framework 4.6.1 Targeting Pack	Microsoft Corporation
Microsoft .NET Framework 4.6.1 SDK	Microsoft Corporation
Microsoft .NET Framework 4.6 Targeting Pack (ENU)	Microsoft Corporation
Microsoft .NET Framework 4.6 Targeting Pack	Microsoft Corporation
Microsoft .NET Framework 4.6 SDK	Microsoft Corporation
Microsoft .NET Framework 4.5.2 Multi-Targeting Pac...	Microsoft Corporation
Microsoft .NET Framework 4.5.2 Multi-Targeting Pack	Microsoft Corporation
Microsoft .NET Framework 4.5.1 SDK	Microsoft Corporation
Microsoft .NET Framework 4.5.1 Multi-Targeting Pac...	Microsoft Corporation
Microsoft .NET Framework 4.5.1 Multi-Targeting Pack	Microsoft Corporation
Microsoft .NET Framework 4.5 Multi-Targeting Pack	Microsoft Corporation

Currently installed programs

Total size: 12.4 GB

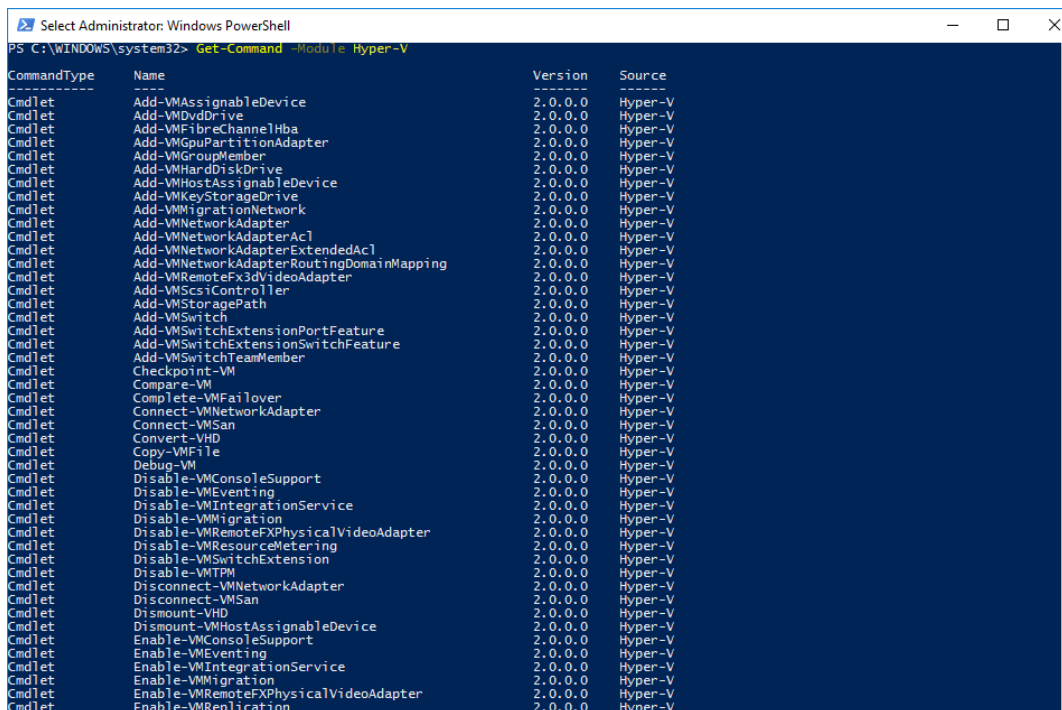
77 programs installed

- Verify if Hyper-V is checked as shown in the below screenshot



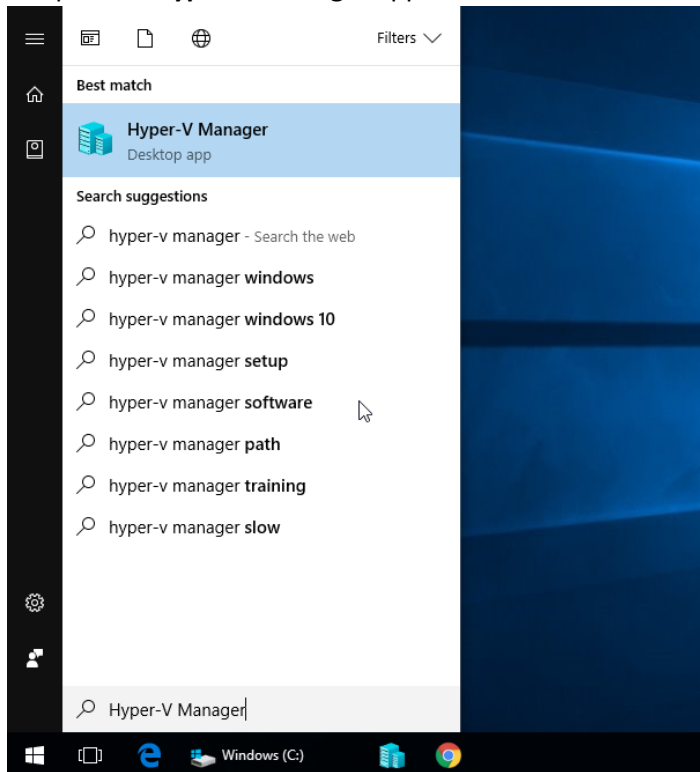
- You can also verify Hyper-V installation via PowerShell
  - Run the following command in PowerShell window. You should see results as shown in the below screenshot.

### Get-Command -Module Hyper-V

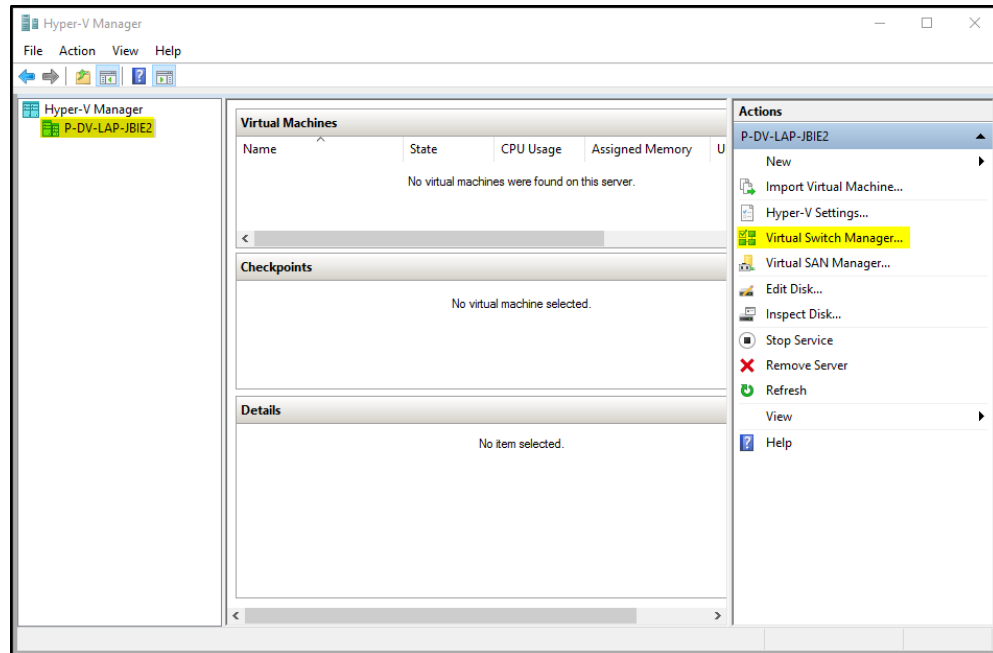


### 3.8 Setup Hyper-V Virtual Switch

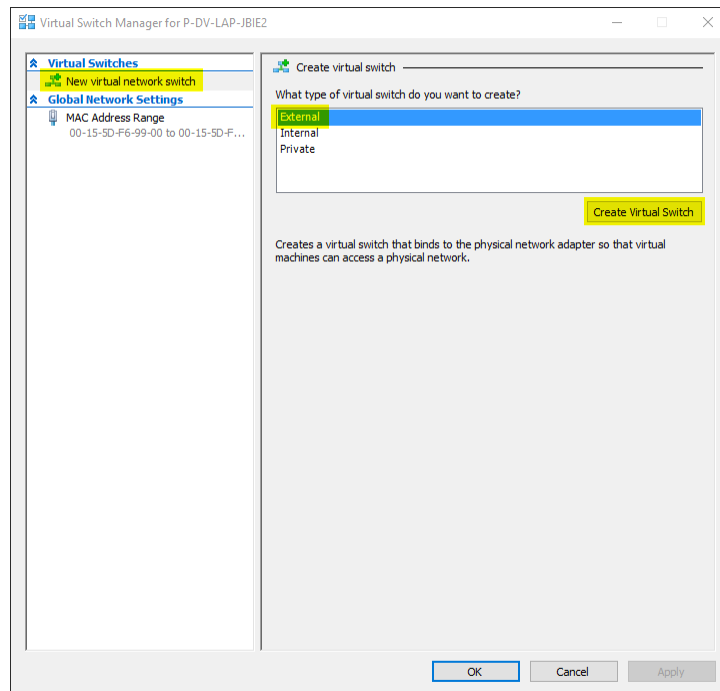
- Open the **Hyper-V Manager** application.



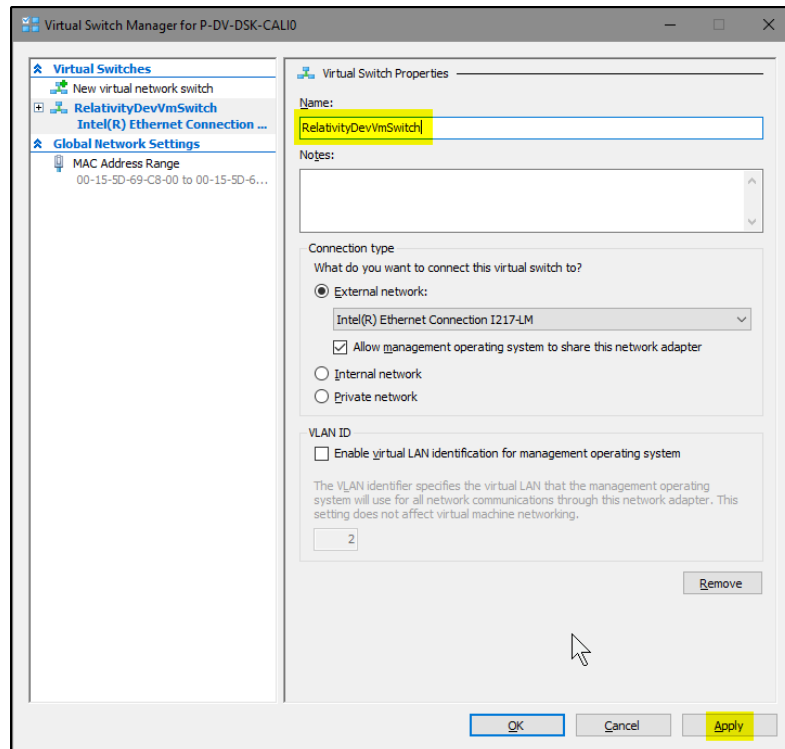
- Make sure you have an **External Virtual Switch** which lets you access your VM from your local machine.
  - Select your local machine from the left pane.
  - In the Actions pane located in **Hyper-V Manager**, click on the **Virtual Switch Manager** link, see screenshot below.



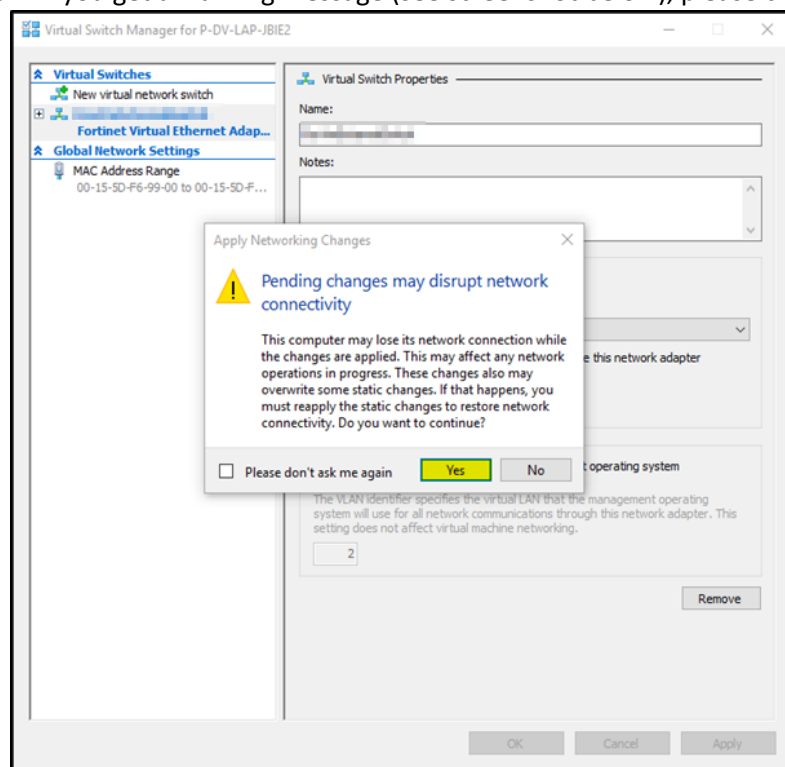
- If you don't already have an external network switch please click the **New virtual network switch** link under **Virtual Switches**. Select **External** for the type of switch and click on the **Create Virtual Switch** button.



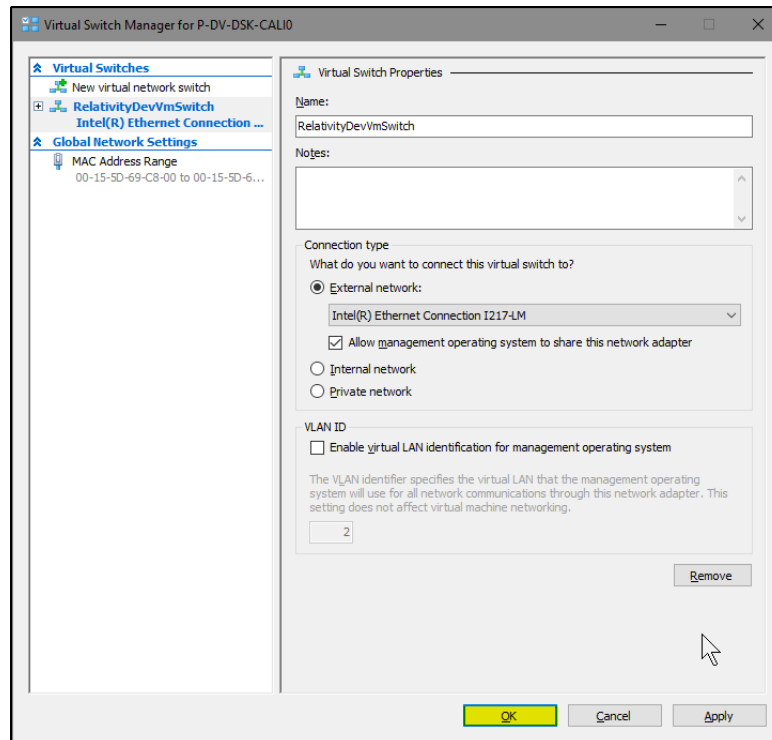
- Provide a friendly name for the new switch (For example: **RelativityDevVmSwitch**) and click **Apply** button.



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

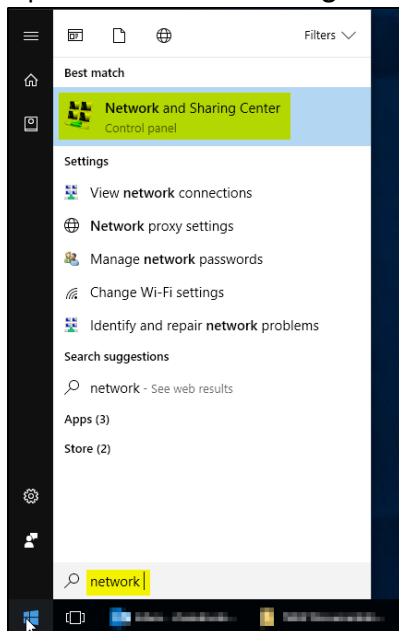


- Next click the **OK** button to create your new virtual switch.

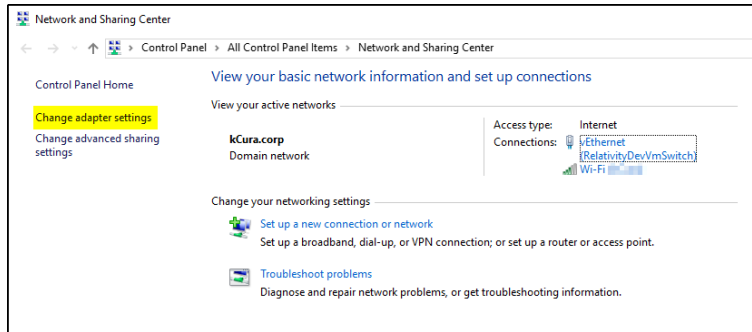


### 3.9 Share Internet Connection

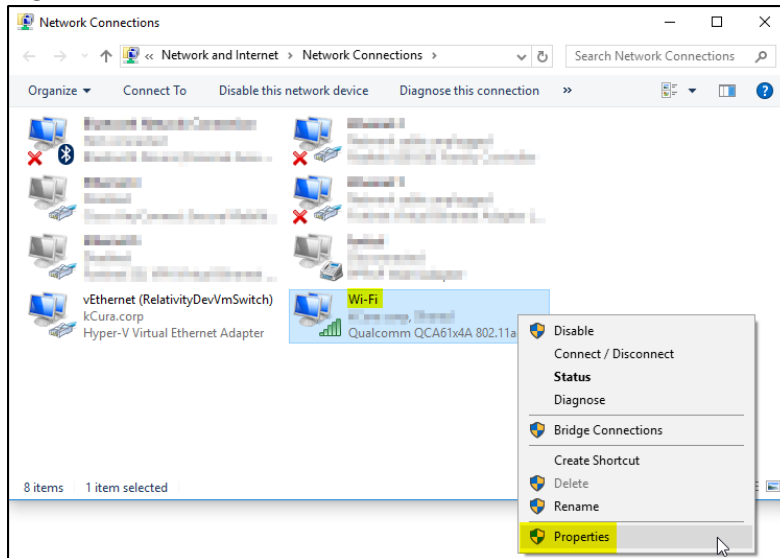
- Sharing the internet connection of your workstation with the Dev VM will make it easy to access it.
- Open **Network and Sharing Center** application on your workstation.



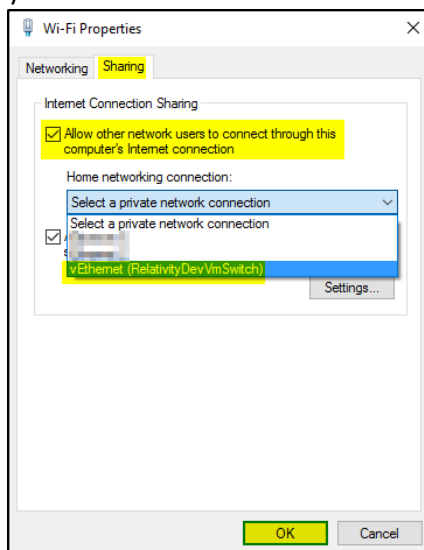
- Click on **Change adapter settings** link on the left pane.



- Right click on the Network connection which has the internet access and select **Properties**.



- Select **Sharing** tab, select the checkbox for **Allow other network users to connect through this computer's Internet connection** option. From the dropdown, select the virtual switch you created for Dev VM and click OK.

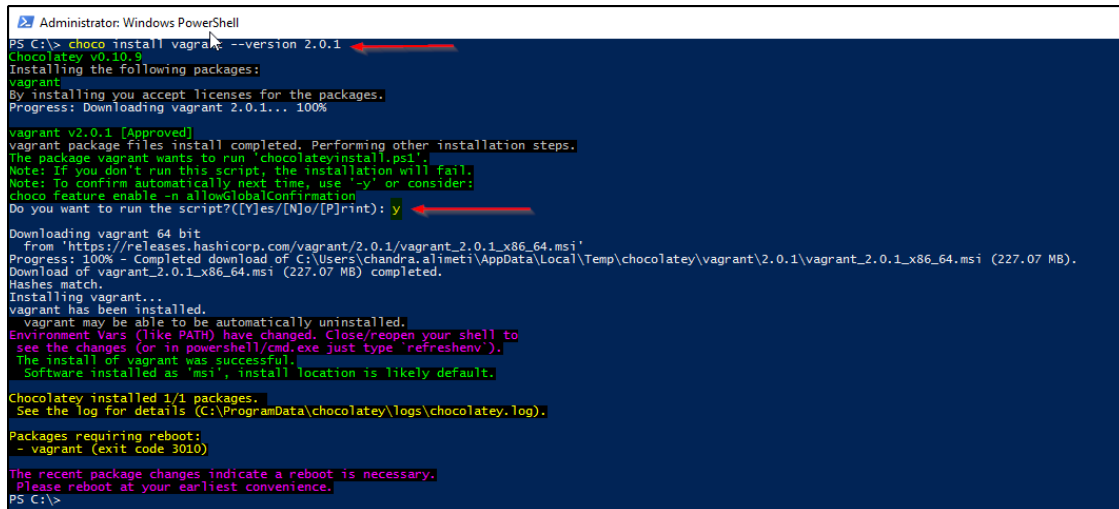




### 3.10 Install Vagrant Version 2.0.1

- Run the following command in PowerShell window.

`choco install vagrant --version 2.0.1`



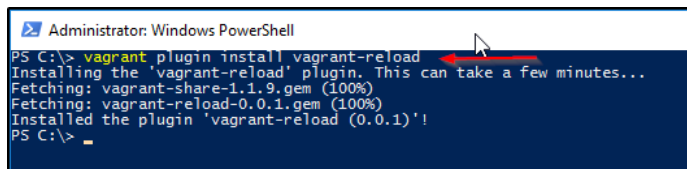
```
Administrator: Windows PowerShell
PS C:\> choco install vagrant --version 2.0.1
Chocolatey v0.10.9
Installing the following packages:
vagrant
By installing you accept licenses for the packages.
Progress: Downloading vagrant 2.0.1... 100%
vagrant v2.0.1 [Approved]
vagrant package files install completed. Performing other installation steps.
The package vagrant wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[N]o/[P]rint): y
Downloading vagrant 64 bit
from 'https://releases.hashicorp.com/vagrant/2.0.1/vagrant_2.0.1_x86_64.msi'
Progress: 100% - Completed download of C:\Users\chandra.alimeti\AppData\Local\Temp\chocolatey\vagrant\2.0.1\vagrant_2.0.1_x86_64.msi (227.07 MB).
Download of vagrant_2.0.1_x86_64.msi (227.07 MB) completed.
Hashes match.
Installing vagrant...
vagrant has been installed.
vagrant may be able to be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
The install of vagrant was successful.
Software installed as 'msi', install location is likely default.
Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
Packages requiring reboot:
- vagrant (exit code 3010)
The recent package changes indicate a reboot is necessary.
Please reboot at your earliest convenience.
PS C:\>
```

- Restart your workstation.

### 3.11 Install Vagrant Reboot plugin

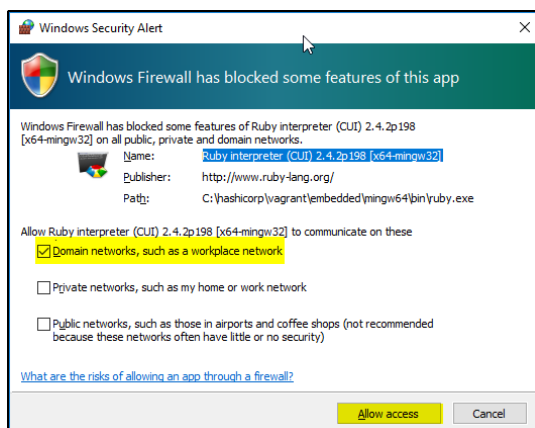
- Run the following command in PowerShell window.

`vagrant plugin install vagrant-reload`



```
Administrator: Windows PowerShell
PS C:\> vagrant plugin install vagrant-reload
Installing the 'vagrant-reload' plugin. This can take a few minutes...
Fetching: vagrant-share-1.1.9.gem (100%)
Fetching: vagrant-reload-0.0.1.gem (100%)
Installed the plugin 'vagrant-reload (0.0.1)'!
PS C:\>
```

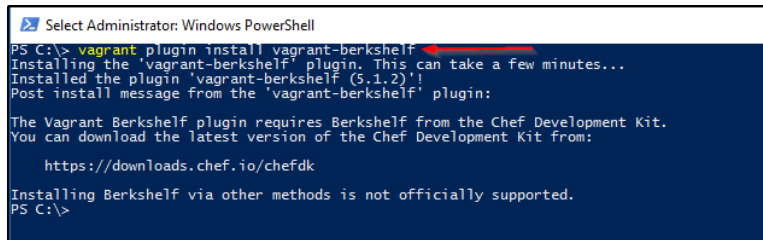
- If you see a popup, select **Allow access** as shown in below screenshot.



### 3.12 Install Vagrant berkshelf plugin

- Run the following command in PowerShell window.

vagrant plugin install vagrant-berkshelf



```
Select Administrator: Windows PowerShell
PS C:\> vagrant plugin install vagrant-berkshelf
Installing the 'vagrant-berkshelf' plugin. This can take a few minutes...
Installed the plugin 'vagrant-berkshelf (5.1.2)'!
Post install message from the 'vagrant-berkshelf' plugin:

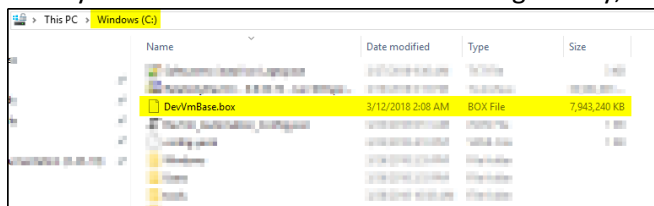
The Vagrant Berkshelf plugin requires Berkshelf from the Chef Development Kit.
You can download the latest version of the Chef Development Kit from:

  https://downloads.chef.io/chefdk

Installing Berkshelf via other methods is not officially supported.
PS C:\>
```

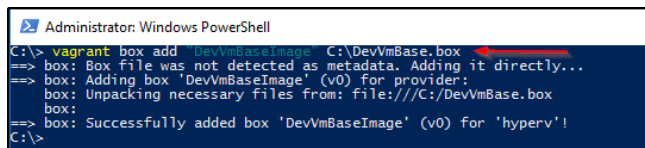
### 3.13 Add Dev VM Windows Base machine to Vagrant boxes list

- Follow the instructions provided in a different document to create a Windows Base machine. This Windows Base image is needed to run the Vagrant/Chef scripts.
- Once you have the Windows Base box image ready, copy it to your C drive.



- Next we need to add the Windows Base box image to the Vagrant boxes list. Run the following command in PowerShell window.

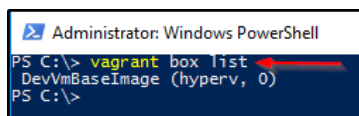
vagrant box add "DevVmBaseImage" C:\DevVmBase.box



```
Administrator: Windows PowerShell
C:\> vagrant box add DevVmBaseImage C:\DevVmBase.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'DevVmBaseImage' (v0) for provider:
box: Unpacking necessary files from: file:///C:/DevVmBase.box
box:
==> box: Successfully added box 'DevVmBaseImage' (v0) for 'hyperv'!
C:\>
```

- Confirm the vagrant box was successfully added. Run the following command in PowerShell window.

vagrant box list



```
Administrator: Windows PowerShell
PS C:\> vagrant box list
DevVmBaseImage (hyperv, 0)
PS C:\>
```

## 4 Create New Dev VM

- It's important that you run the below steps in order.

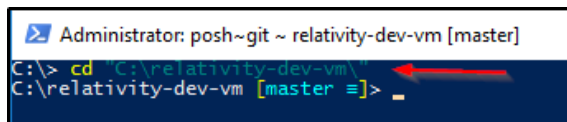
### 4.1 Login

- Login into your workstation with a Windows Administrator account.

### 4.2 Navigate to the Dev VM GIT repository

- Run the following command in PowerShell window.

`cd "C:\relativity-dev-vm\"`

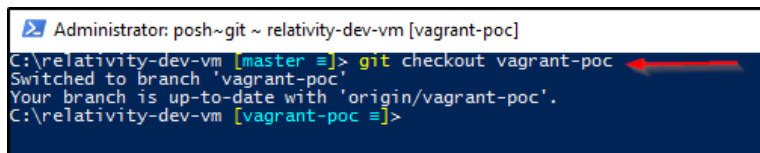


```
Administrator: posh~git ~ relativity-dev-vm [master]
C:\> cd "C:\relativity-dev-vm\"
C:\relativity-dev-vm [master =>]>
```

### 4.3 Switch to vagrant-poc branch

- Run the following command in PowerShell window.

`git checkout vagrant-poc`

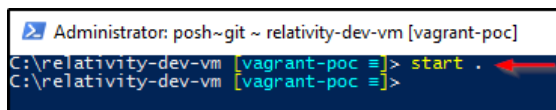


```
Administrator: posh~git ~ relativity-dev-vm [vagrant-poc]
C:\relativity-dev-vm [master =>]> git checkout vagrant-poc
Switched to branch 'vagrant-poc'
Your branch is up-to-date with 'origin/vagrant-poc'.
C:\relativity-dev-vm [vagrant-poc =>]>
```

### 4.4 Copy the config.yaml file required for Dev VM

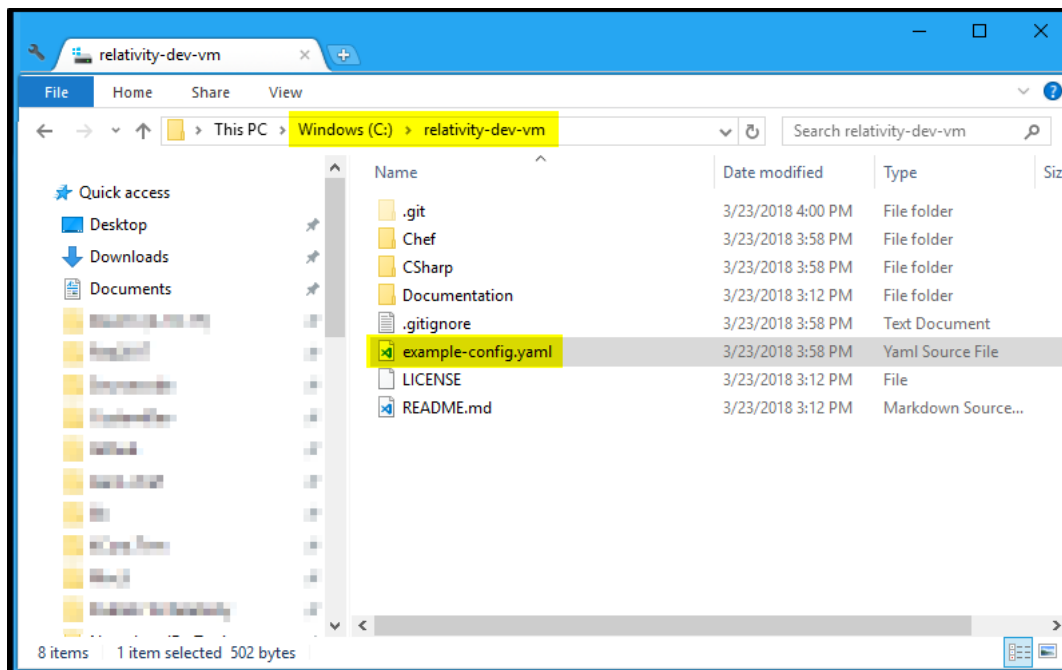
- Run the following command in PowerShell window.

`start .`

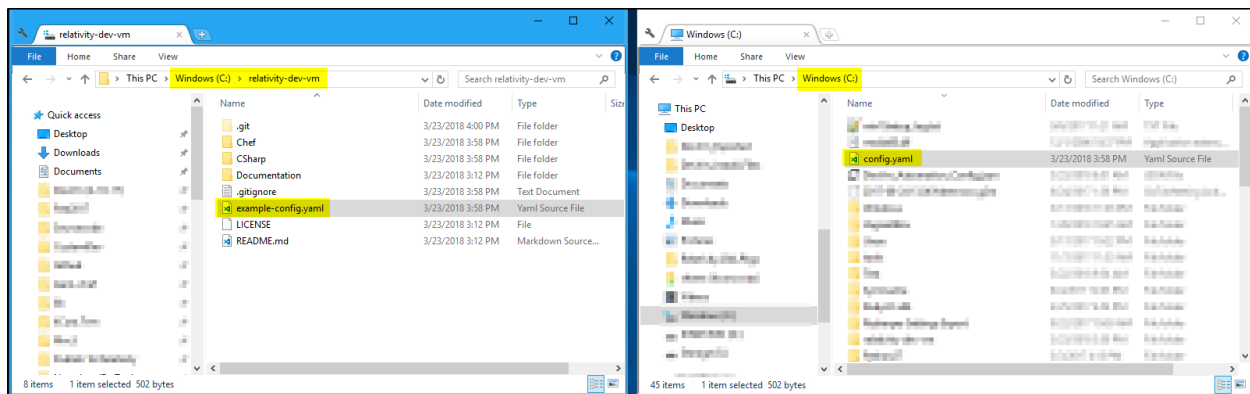


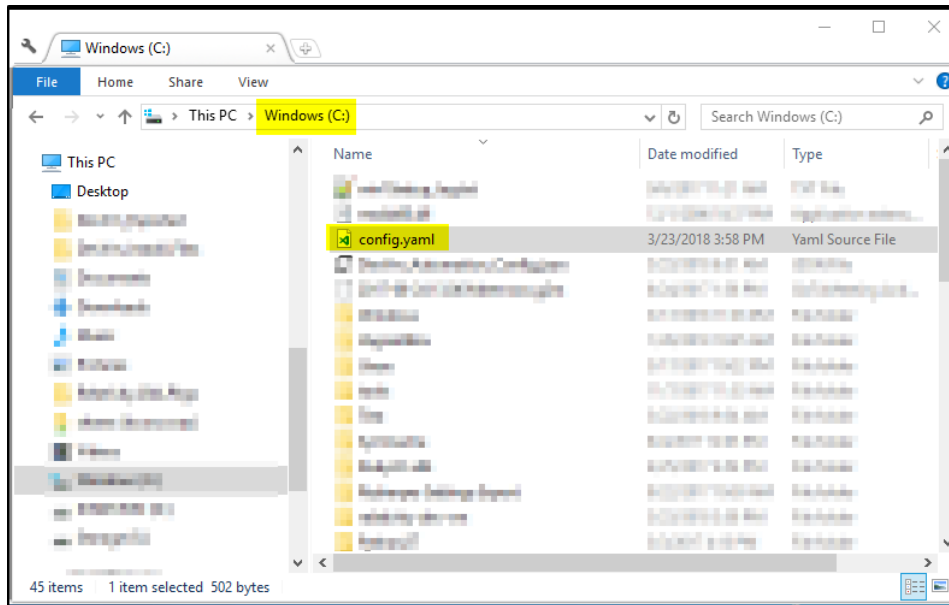
```
Administrator: posh~git ~ relativity-dev-vm [vagrant-poc]
C:\relativity-dev-vm [vagrant-poc =>]> start .
C:\relativity-dev-vm [vagrant-poc =>]>
```

- This opens the current folder in Windows Explorer.



- Copy the **example-config.yaml** file from C:\relativity-dev-vm folder to C drive and rename it to **config.yaml**.





- Open the **config.yaml** file from **C drive** in Notepad.

```
config.yaml - Notepad
File Edit View Help
# Copy this file to the parent directory of the repopulate
# Populate the variables
# Rename this file to config.yaml

credentials:
  vm_username: 'Administrator'
  vm_password: 'Test1234!'
  smb_domain: 'relativity'
  smb_username: 'calimeti'
  smb_password: 'MyWorkstationPassword'

email:
  smtp_server: 'smtp.relativity.com'
  smtp_port: '25'
  email_from: 'noreply@relativity.com'
  email_to: 'test.user1@relativity.com;test.user2@relativity.com'
```

- Update the properties in the **config.yaml** file. Please find below a brief description of each property.
  - **vm\_username**: Windows Admin username of the Dev VM.
  - **vm\_password**: Windows Admin password of the Dev VM.
  - **smb\_domain**: Domain of your workstation. If your workstation is not part of a domain, use your workstation computer name.
  - **smb\_username**: Windows Admin username of your workstation.

- **smb\_password**: Windows Admin password of your workstation.
- **smtp\_server**, **smtp\_port**, **email\_from**, **email\_to** will be values for the email server on your domain. These values will be used to send the report of the Smoke tests run after the Dev VM creation.
- Save the changes to the **config.yaml** file.

## 4.5 Update Configuration file

- Navigate to the configuration file directory. Run the following command in PowerShell window.

```
cd "C:\relativity-dev-vm\Chef\Cookbooks\Relativity\attributes"
```

```
Administrator: posh-git ~ relativity-dev-vm [vagrant-poc]
C:\> cd "C:\relativity-dev-vm\Chef\Cookbooks\Relativity\attributes"
C:\relativity-dev-vm\Chef\Cookbooks\Relativity\attributes [vagrant-poc =>] ls

Directory: C:\relativity-dev-vm\Chef\Cookbooks\Relativity\attributes

Mode                LastWriteTime         Length Name
----                -
-a----            3/28/2018 11:27 AM         21064 default.rb

C:\relativity-dev-vm\Chef\Cookbooks\Relativity\attributes [vagrant-poc =>]
```

- The directory contains one file name **default.rb**.
- Open the **default.rb** file and update the necessary properties.

```
default['timeout']['default'] = 3600

default['windows']['hostname'] = node['machinename']
default['windows']['new_computer_name'] = 'RelativityDevVm'
default['windows']['user']['admin']['login'] = 'Administrator'
default['windows']['user']['admin']['password'] = 'Test1234!'

default['file']['installers']['default_destination_folder'] = 'C:\\Chef_Install'
default['file']['log']['default_destination_folder'] = 'C:\\Chef_Logs'
default['file']['log']['name'] = 'log.txt'
default['file']['result']['name'] = 'result_file.txt'
default['file']['result']['destination_folder'] = 'C:\\vagrant'
```

## 4.6 Run the Dev VM Creation script

### 4.6.1 Navigate to the Dev VM GIT repository

- Run the following command in PowerShell window.

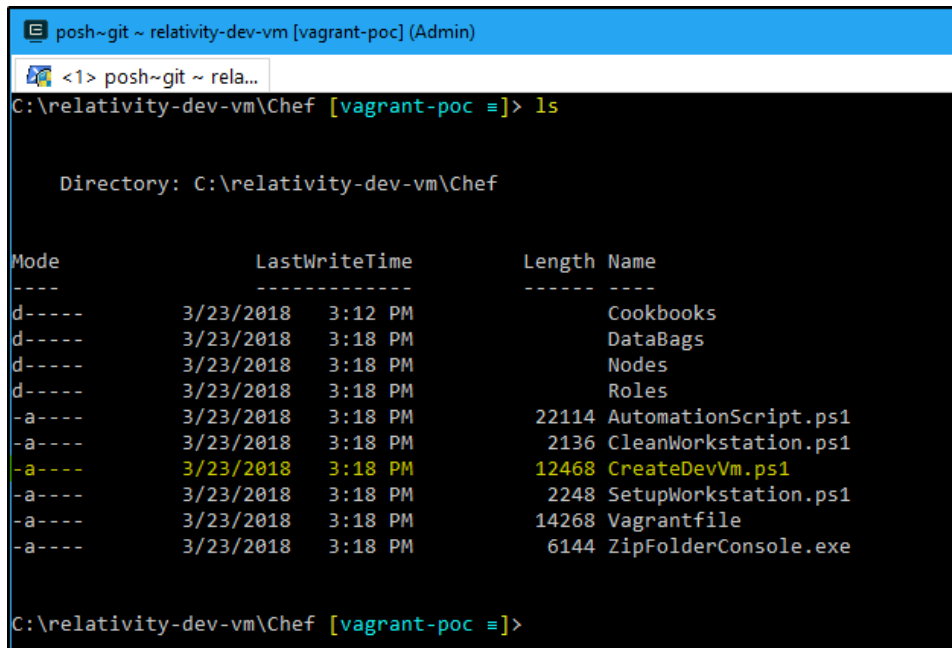
```
cd "C:\relativity-dev-vm\Chef\"
```

```
Administrator: posh-git ~ relativity-dev-vm [vagrant-poc]
C:\relativity-dev-vm [vagrant-poc =>] cd "C:\relativity-dev-vm\Chef\"
C:\relativity-dev-vm\Chef [vagrant-poc =>]
```

#### 4.6.2 List all the files in Chef folder

- Run the following command in PowerShell window.

ls



```
posh~git ~ relativity-dev-vm [vagrant-poc] (Admin)
<1> posh~git ~ rela...
C:\relativity-dev-vm\Chef [vagrant-poc ==]> ls

Directory: C:\relativity-dev-vm\Chef

Mode                LastWriteTime         Length Name
----                -
d-----          3/23/2018   3:12 PM             Cookbooks
d-----          3/23/2018   3:18 PM             DataBags
d-----          3/23/2018   3:18 PM             Nodes
d-----          3/23/2018   3:18 PM             Roles
-a-----          3/23/2018   3:18 PM        22114 AutomationScript.ps1
-a-----          3/23/2018   3:18 PM        2136 CleanWorkstation.ps1
-a-----          3/23/2018   3:18 PM       12468 CreateDevVm.ps1
-a-----          3/23/2018   3:18 PM        2248 SetupWorkstation.ps1
-a-----          3/23/2018   3:18 PM       14268 Vagrantfile
-a-----          3/23/2018   3:18 PM        6144 ZipFolderConsole.exe

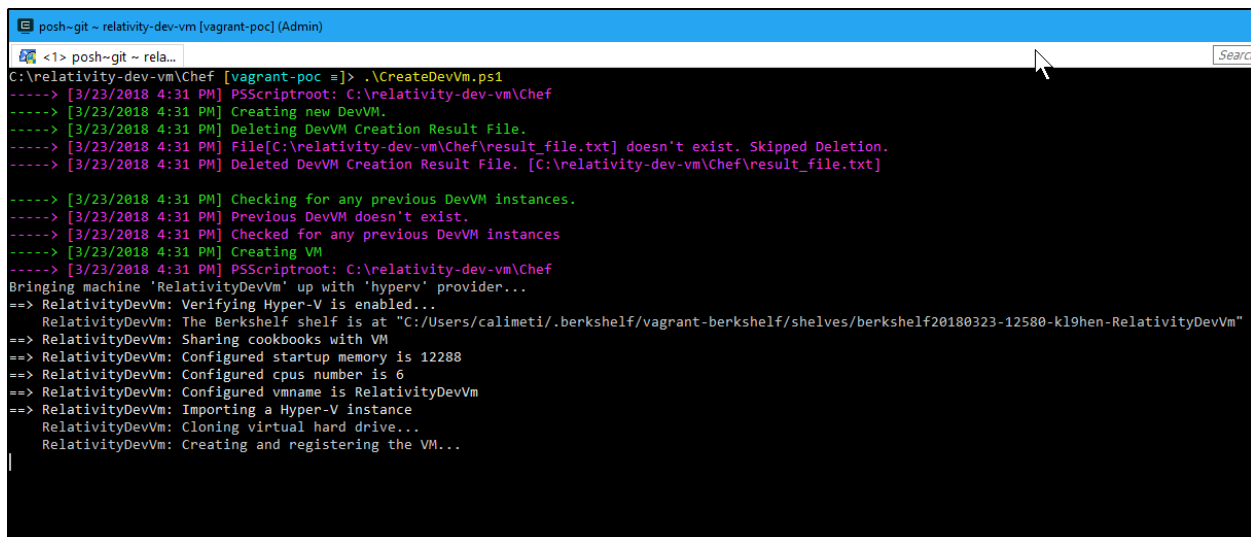
C:\relativity-dev-vm\Chef [vagrant-poc ==]>
```

- **CreateDevVm.ps1** PowerShell script has the code to create a new Dev VM.

#### 4.6.3 Run the Creation script

- Run the following command in PowerShell window.

.\CreateDevVm.ps1



```
posh~git ~ relativity-dev-vm [vagrant-poc] (Admin)
<1> posh~git ~ rela...
C:\relativity-dev-vm\Chef [vagrant-poc ==]> .\CreateDevVm.ps1
-----> [3/23/2018 4:31 PM] PSScriptroot: C:\relativity-dev-vm\Chef
-----> [3/23/2018 4:31 PM] Creating new DevVM.
-----> [3/23/2018 4:31 PM] Deleting DevVM Creation Result File.
-----> [3/23/2018 4:31 PM] File[C:\relativity-dev-vm\Chef\result_file.txt] doesn't exist. Skipped Deletion.
-----> [3/23/2018 4:31 PM] Deleted DevVM Creation Result File. [C:\relativity-dev-vm\Chef\result_file.txt]

-----> [3/23/2018 4:31 PM] Checking for any previous DevVM instances.
-----> [3/23/2018 4:31 PM] Previous DevVM doesn't exist.
-----> [3/23/2018 4:31 PM] Checked for any previous DevVM instances
-----> [3/23/2018 4:31 PM] Creating VM
-----> [3/23/2018 4:31 PM] PSScriptroot: C:\relativity-dev-vm\Chef
Bringing machine 'RelativityDevVm' up with 'hyperv' provider...
==> RelativityDevVm: Verifying Hyper-V is enabled...
    RelativityDevVm: The Berkshelf shelf is at "C:/Users/calimeti/.berkshelf/vagrant-berkshelf/shelves/berkshelf20180323-12580-k19hen-RelativityDevVm"
==> RelativityDevVm: Sharing cookbooks with VM
==> RelativityDevVm: Configured startup memory is 12288
==> RelativityDevVm: Configured cpus number is 6
==> RelativityDevVm: Configured vmname is RelativityDevVm
==> RelativityDevVm: Importing a Hyper-V instance
    RelativityDevVm: Cloning virtual hard drive...
    RelativityDevVm: Creating and registering the VM...
```

- Once the script finishes execution successfully, you should be able to see a

## 5 Clean Workstation

- It's important that you run the below steps in order.

### 5.1 Login

- Login into your workstation with a Windows Administrator account.

### 5.2 Remove Vagrant Box Image

- Run the following command in PowerShell window.

```
vagrant box remove "DevVmBaseImage" -force
```

### 5.3 Uninstall Vagrant berkshelf plugin

- Run the following command in PowerShell window.

```
vagrant plugin uninstall vagrant-berkshelf
```

### 5.4 Uninstall Vagrant reload plugin

- Run the following command in PowerShell window.

```
vagrant plugin uninstall vagrant-reload
```

### 5.5 Uninstall Vagrant

- Run the following command in PowerShell window.

```
choco uninstall vagrant -force
```

### 5.6 Uninstall Chocolatey

- Run the following command in PowerShell window.

```
Remove-Item -Recurse -Force "$env:ChocolateyInstall"
```

- Run the following command in PowerShell window.

```
[System.Text.RegularExpressions.Regex]::Replace([Microsoft.Win32.Registry]::CurrentUser.OpenSubKey('Environment').GetValue('PATH', "", [Microsoft.Win32.RegistryValueOptions]::DoNotExpandEnvironmentNames).ToString(), [System.Text.RegularExpressions.Regex]::Escape("$env:ChocolateyInstall\bin") + '{?>}?', "",
```



```
[System.Text.RegularExpressions.RegexOptions]::IgnoreCase) | %  
{[System.Environment]::SetEnvironmentVariable('PATH', $_, 'User')}
```

- Run the following command in PowerShell window.

```
[System.Text.RegularExpressions.Regex]::Replace([Microsoft.Win32.Registry]::LocalMachine.OpenSubKey('SYSTEM\CurrentControlSet\Control\Session Manager\Environment\').GetValue('PATH', '[Microsoft.Win32.RegistryValueOptions]::DoNotExpandEnvironmentNames').ToString(),  
[System.Text.RegularExpressions.Regex]::Escape("$env:ChocolateyInstall\bin") + '{?>}?', '[System.Text.RegularExpressions.RegexOptions]::IgnoreCase) | %  
{[System.Environment]::SetEnvironmentVariable('PATH', $_, 'Machine')}
```

- Run the following command in PowerShell window.

```
if ($env:ChocolateyBinRoot -ne "" -and $env:ChocolateyBinRoot -ne $null) { Remove-Item -Recurse -Force "$env:ChocolateyBinRoot" }
```

- Run the following command in PowerShell window.

```
if ($env:ChocolateyToolsRoot -ne "" -and $env:ChocolateyToolsRoot -ne $null) { Remove-Item -Recurse -Force "$env:ChocolateyToolsRoot" }
```

- Run the following command in PowerShell window.

```
[System.Environment]::SetEnvironmentVariable("ChocolateyBinRoot", $null, 'User')
```

- Run the following command in PowerShell window.

```
[System.Environment]::SetEnvironmentVariable("ChocolateyToolsLocation", $null, 'User')
```

## 5.7 Remove Hyper-V Virtual Switch

- Run the following command in PowerShell window.

```
Remove-VMSwitch "RelativityDevVmSwitch" -Force
```

## 5.8 Uninstall Hyper-V

- Run the following command in PowerShell window.

```
Disable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -NoRestart
```

- **Restart your workstation.**

## 5.9 Uninstall Chef Development Kit

- Open **Programs and Features** application.
- Search for **Chef Development Kit**, select it and click Uninstall.

Control Panel Home

View installed updates

Turn Windows features on or off

Install a program from the network

Control Panel > All Control Panel Items > Programs and Features

chef

Uninstall or change a program


To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.


Organize

Uninstall

Change

Repair

Name	Publisher	Installed On	Size	Version
 Chef Development Kit v2.4.17	Chef Software, Inc.	1/26/2018	680 MB	2.4.17.1



Chef Software, Inc. Product version: 2.4.17.1

Size: 680 MB

Help link: <http://www.getchef.com>