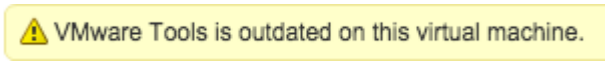
VMware Tools and VM Compatibility Upgrade

Table of contents

VMware Tools and VM Compatibility Upgrade

Walkthrough: Updating Tools in a vSphere Environment



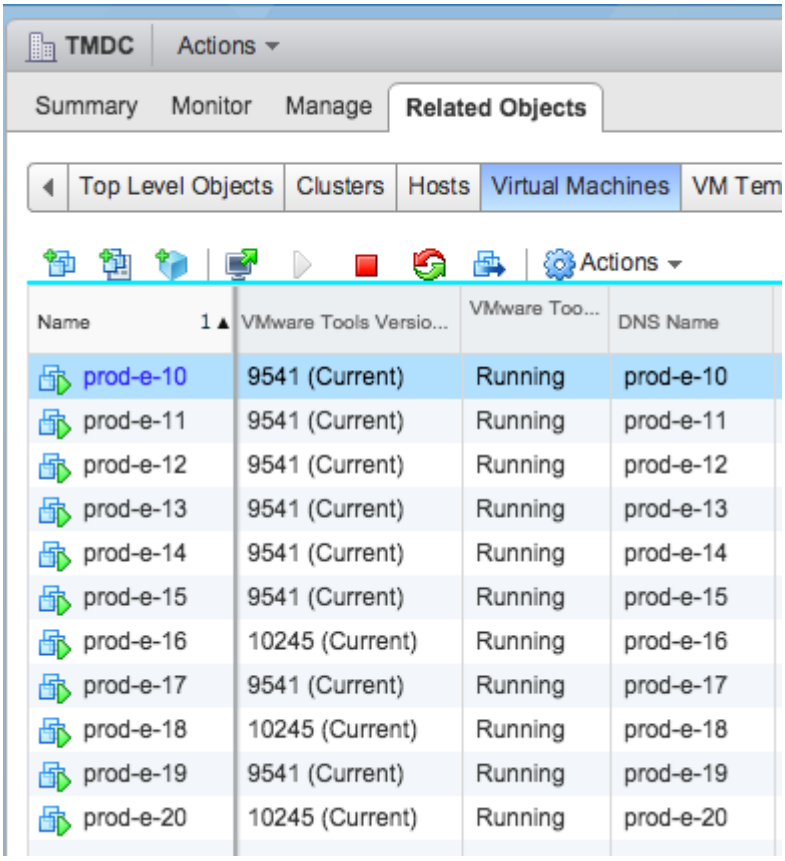
For keeping VMware Tools up to date, there are six different approaches that vSphere administrators can use to accommodate nearly any workflow required for flexible datacenter operations. These different techniques allow optimizing either for automation and standardization or for separation of responsibilities. A previous article provides an overview of the three types of VM Tools.

VMware Tools Status is Relative to Underlying Host

Each ESXi host has a storage location for VM Tools installers, which is a configurable option and visibly referenced by the /productLocker symlink. The target can be either local to each host or point to a centralized repository of VM Tools on a shared datastore. For more information about setting up a shared Tools repository, see this earlier post or KB 2004018.

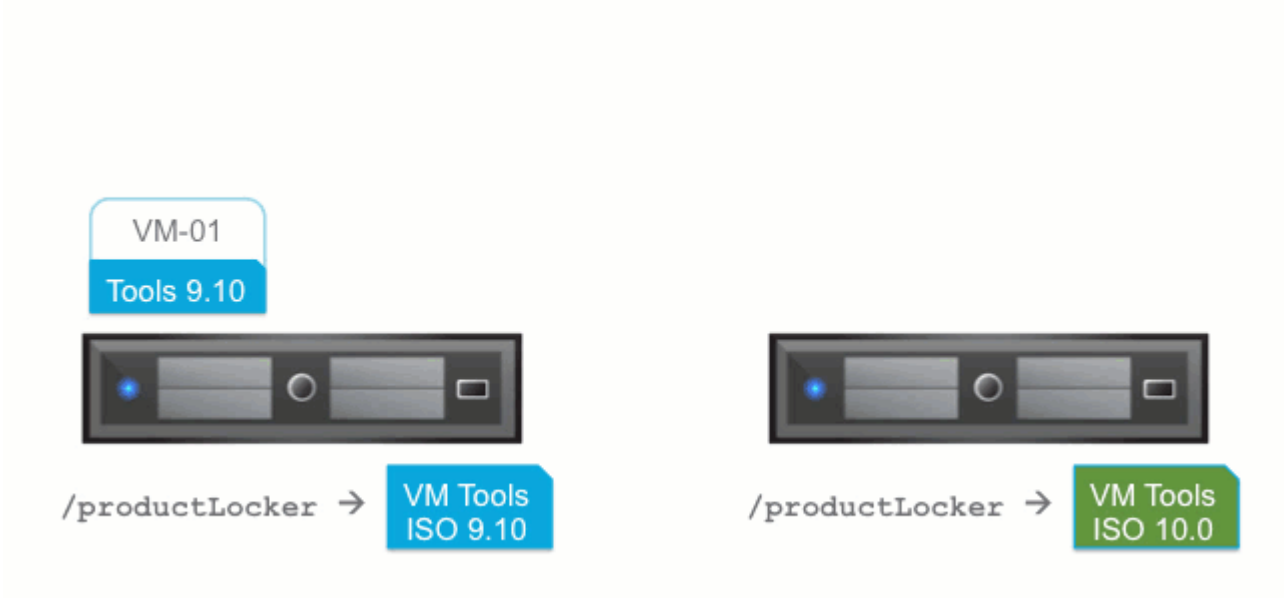
```
productLocker -> /vmfs/volumes/NFS-C/vmtools_shared
```

The VM Tools status for any given VM is always in the context of the underlying host. As demonstrated below, two different versions of Tools are considered “Current” because the underlying hosts are not identical.



In vSphere 6.5, the version of VM Tools running on each guest is compared to the version associated with the underlying ESXi host on a periodic schedule. In vSphere 6.0 and prior this check is performed when certain virtual machine events occur, such as power-on or vMotion,. If the host has a newer version, the VM is considered out of date.

Note that there is no mechanism for VMs running on vSphere to *contact the mother ship* and learn about new versions of VM Tools – only the VM-to-host relationship is relevant. This explains why a VM may suddenly complain about having outdated VM Tools after migrating from one host to another – it is because the destination host has a more recent version in the product locker.



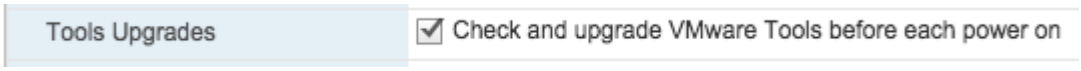
VM Tools Type Determines Update Choices

A previous article explains why there are three types of VM Tools: the familiar Tools ISOs for all supported operating systems, plus two additional offerings in the form of binary packages for Linux. There are several ways to initiate VM Tools updates from vSphere or from within a guest. The following applies **only** to Windows and Linux guests using VM Tools ISOs except where noted. The VM Tools Linux packages – OVT and OSP – are not managed via vSphere, so they can only be installed and updated from within each guest OS using native package management tools.

Six Ways to Keep VM Tools Updated

1. Automatic update on VM boot

The easiest way to keep VM Tools up to date is to check a box and forget about managing this element of infrastructure. Upon VM reboot, such as after installing guest OS patches, the VM Tools status will be checked and updated when necessary. In many cases, this will result in one additional reboot after the VM Tools installation completes.



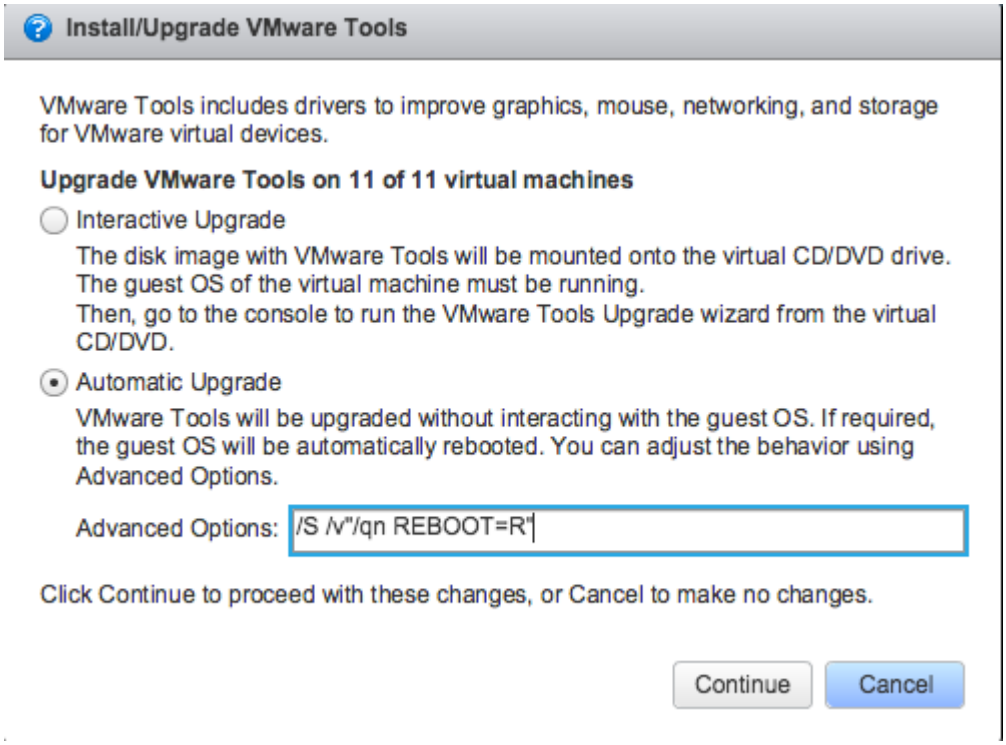
This approach may be viable for less-critical workloads, such as labs or test/dev environments. Imagine a scenario where VMs are rebooting unexpectedly due to a widespread infrastructure outage. After scrambling to get applications back online, administrators could find themselves facing unanticipated subsequent reboots if a VM Tools update happened to be available. This is an edge case, but one to keep in mind.

2. Initiate update to one or more VMs through the vSphere UI

In the vSphere Web Client, when a VM indicates that VM Tools are outdated, the adjacent button can be used to automatically initiate an update. This can be done interactively or in a completely hands-off fashion. In the latter case, administrators also have the option of suppressing any potential reboots on Windows VMs – this is a good option that enables coordination of reboots required after routine guest OS patching.

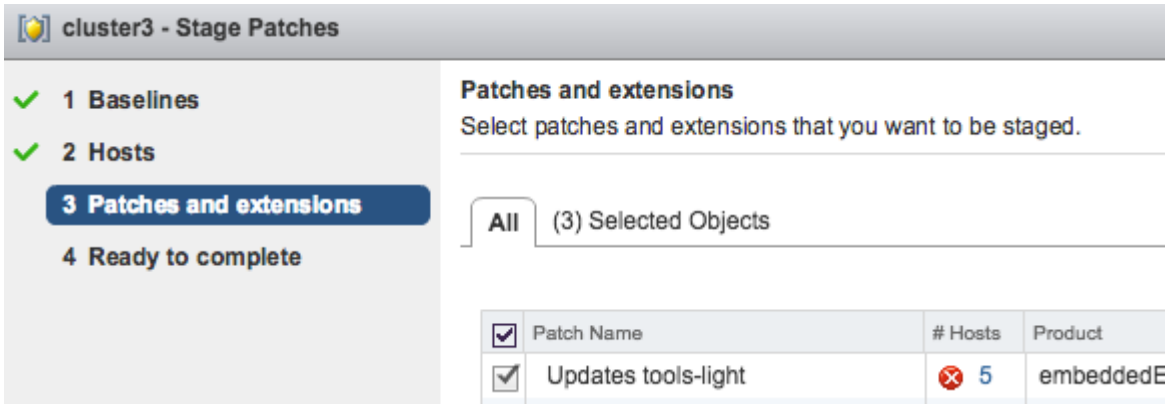
Important note for guests other than Windows and Linux: Solaris, FreeBSD, and Mac OS - VM Tools can only be updated using the manual interactive method. Currently, there is no automatic Tools update for these guests.

Going a step further, it is also possible to select multiple VMs in the Web Client UI and initiate a VM Tools update on all of them at once.

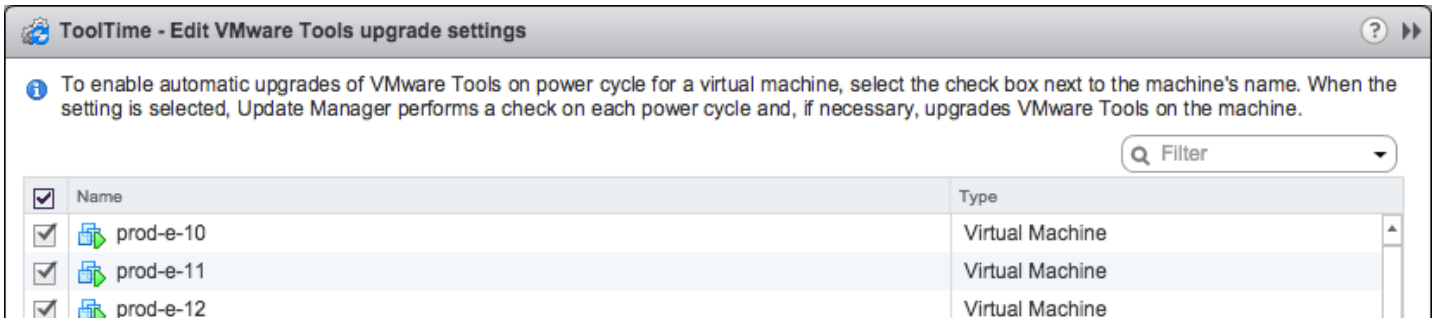


3. VMware Update Manager: immediate, scheduled, or on boot

VMware Update Manager (VUM) has two very different roles to play when it comes to updating VM Tools. The first one fetches updated VM Tools ISOs in the form of the 'tools-light' VIB that is offered when needed in the normal ESXi patch stream. This patch is then pushed out to all managed hosts, in accordance to the baselines established by administrators. Once this occurs, individual VMs will begin to detect that a new version of VM Tools is available and ready for update.



The second role VUM has in managing VM Tools is to trigger updates for individual VMs in accordance to baselines. Keep in mind that VUM does this work by leveraging the vSphere methods described in the two previous sections. In one mode, VUM can be used to make various configuration changes to multiple VMs so that a Tools update is checked and performed as necessary on each guest reboot, just like an administrator can do using the technique shown in #1 above. The advantage of using VUM is that many VMs can be configured or un-configured for this option at once.



The other mode VUM uses is to trigger a VM Tools update either immediately or at a scheduled time, just as an administrator can do manually as described in #2 above. One added benefit of using VUM to initiate these updates in this way is the ability to also remediate powered-off or suspended VMs, subsequently returning them to their initial state after update.

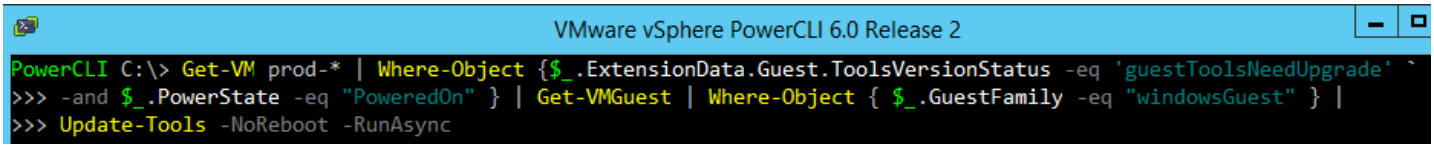
4. In-guest update – delegating control to app owners

For scenarios where application owners demand tight control over activities that occur in the guest OS, there is an option to allow in-guest updates of VM Tools. A tray icon in Windows will indicate that an update is available, and the VM Tools configuration dialog box will permit interactive initiation of an update at a convenient time.

For equivalent functionality from a command-line utility, vmware-toolbox-cmd is offered for Linux as well as Windows guests. Keep in mind that for Linux this is only for the VM Tools ISOs, since OVT and OSP use a different process, as described in #6 below.

5. Mass updates through PowerCLI automation

For very large environments or for those that have established more mature operational processes, PowerCLI provides a powerful option for updating VM Tools. This approach can target particular groups of VMs in many convenient ways, such as by cluster, by guest OS version, tags, VM state, or other vSphere attributes.



6. Native Linux package management processes

By nature of their design, Linux guests running OSPs or OVTs update VM Tools as part of a broader patching and updating workflow used for other components. This allows administrators to leverage existing Linux package managers or broader patch management and monitoring solutions without the need to coordinate with vSphere administrators.

```
# yum update open-vm-tools
```

7. BONUS: VM Tools Upgrade Method (pardon the pun)

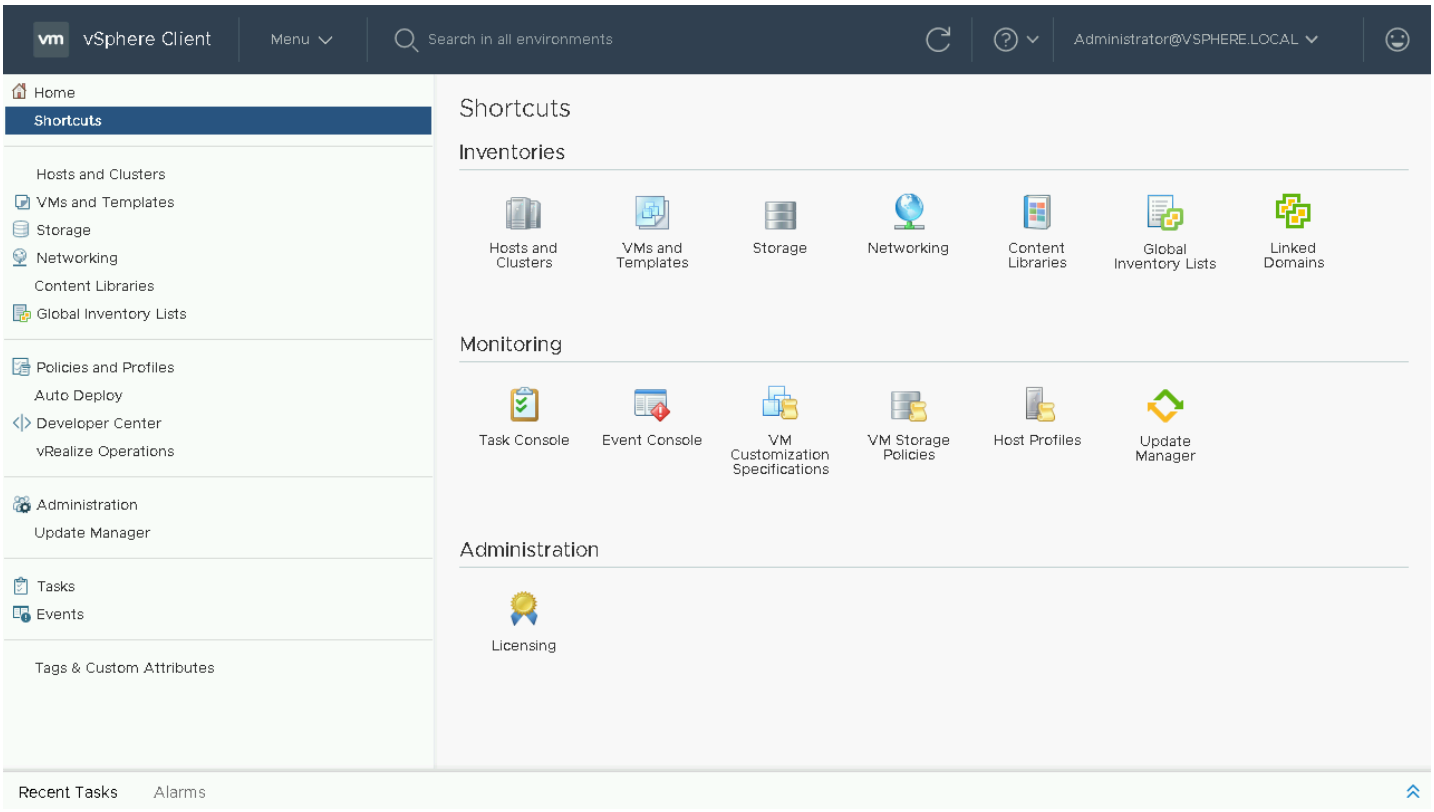
For advanced use cases where vSphere APIs are being used for deeper integration with other processes, consider the UpgradeTools_Task for programmatic upgrades of VM Tools.

Summary

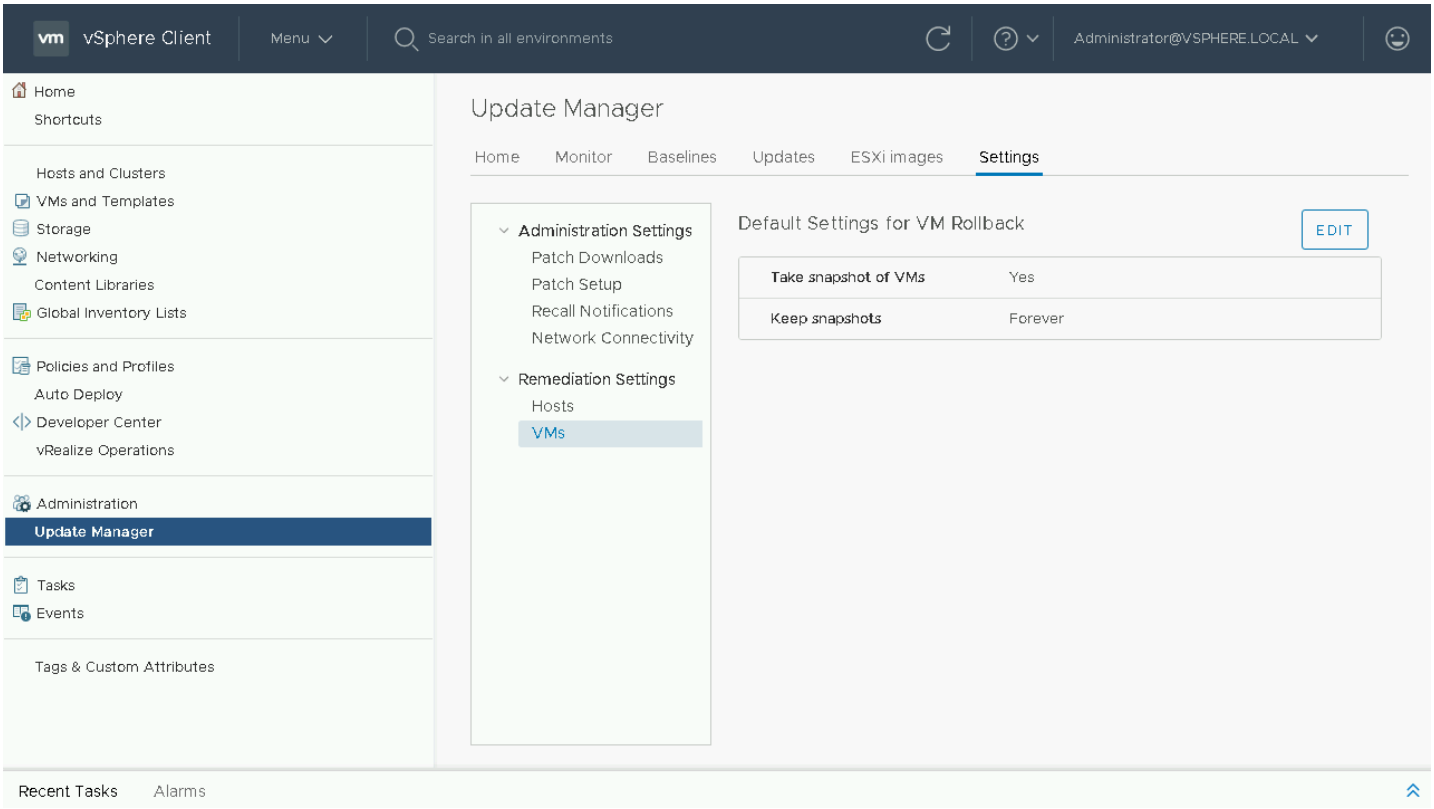
With these flexible means of updating VM Tools, there is a suitable approach for any VMware datacenter, whether the requirement is centralized control, automation, delegation to app owners, or integration with existing patch management processes.

Walkthrough: Upgrading VM Tools using vSphere Update Manager

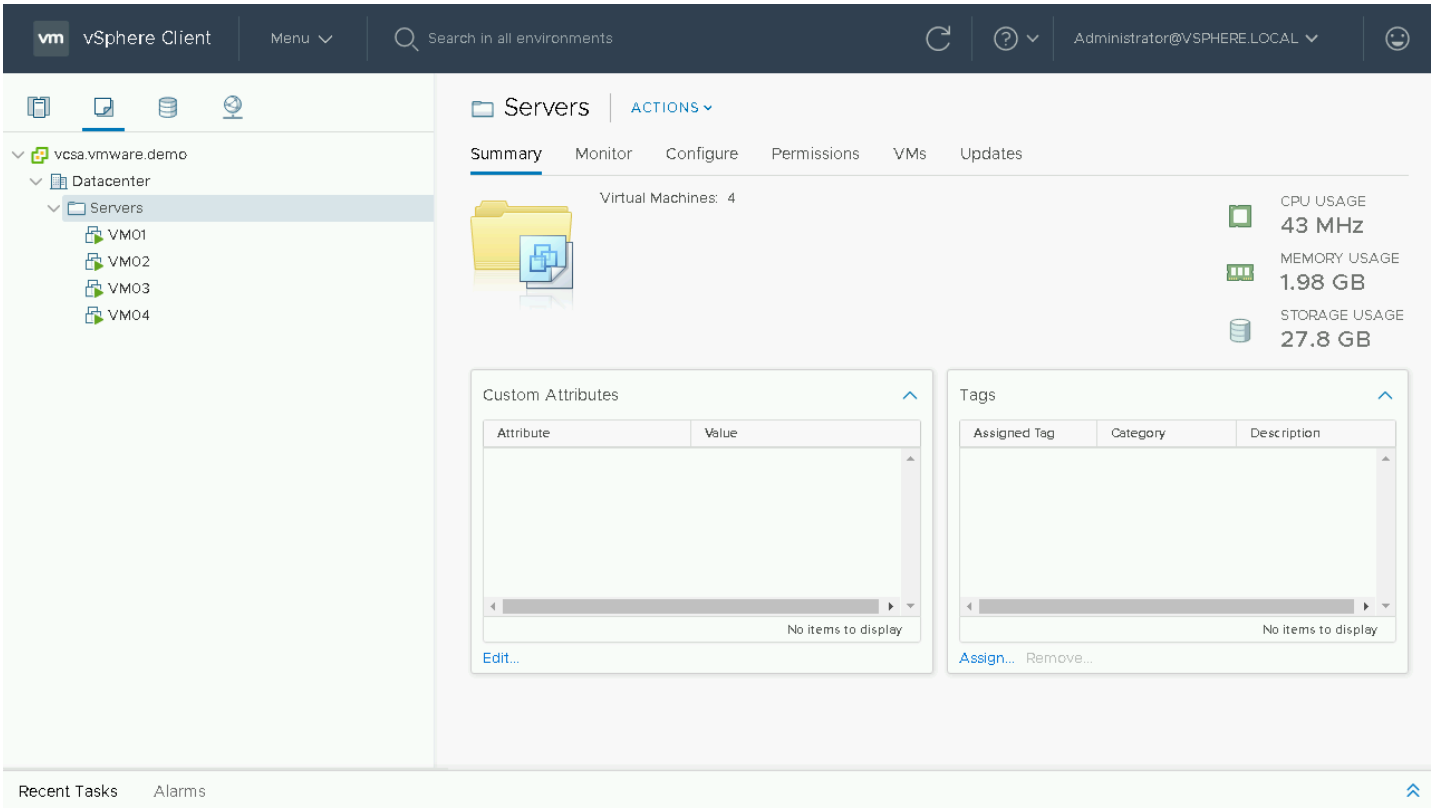
This product walkthrough will cover the steps to upgrade VM Tools using vSphere Update Manager



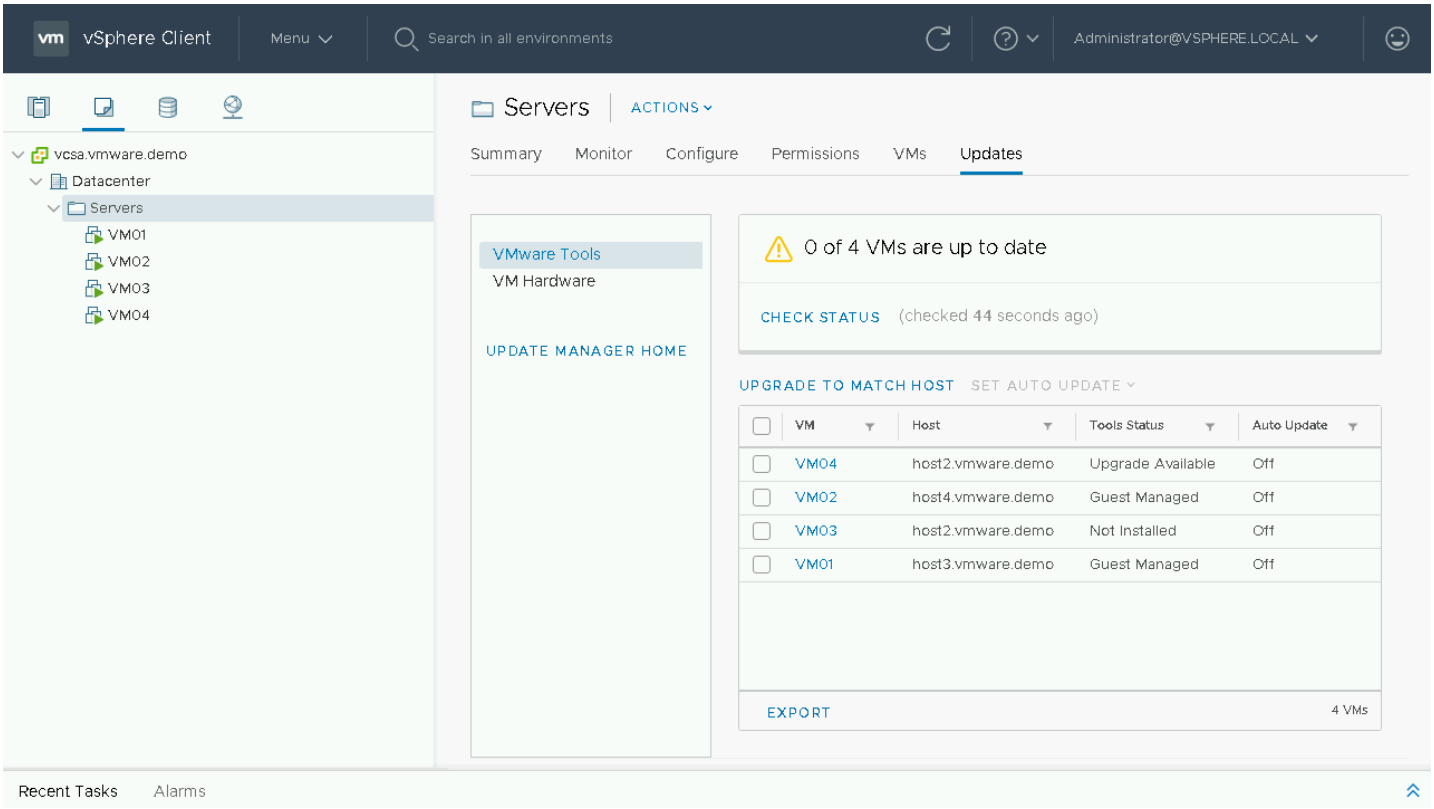
Before we begin the upgrade of VM Tools, lets navigate to the Update Manager Settings for VMs.



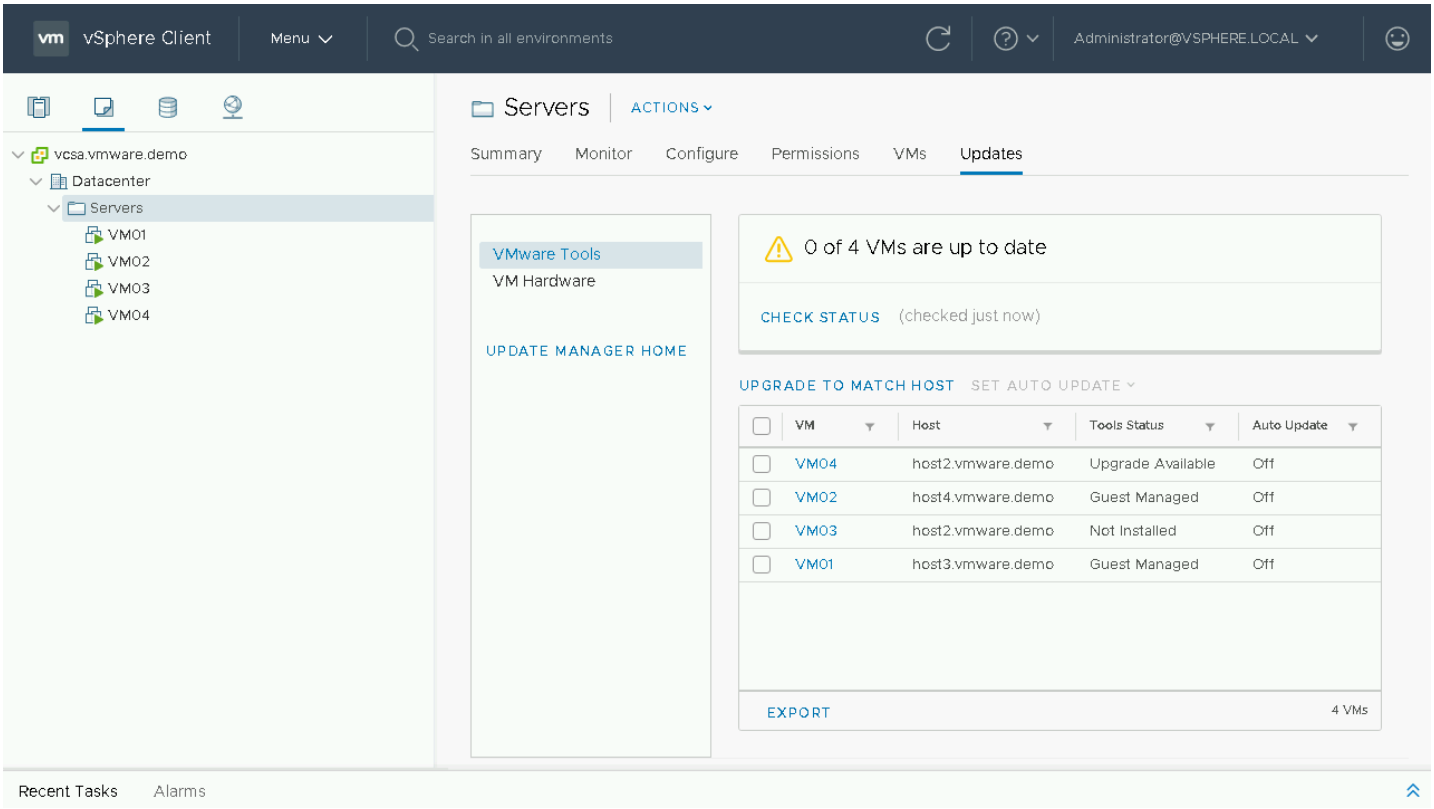
On the current screen you can set the default settings for VM Rollback. This includes whether or not a snapshot should be taken of the VM and the time to retain the snapshot. Let's navigate to VMs and Templates view to begin our VM Tools Upgrade



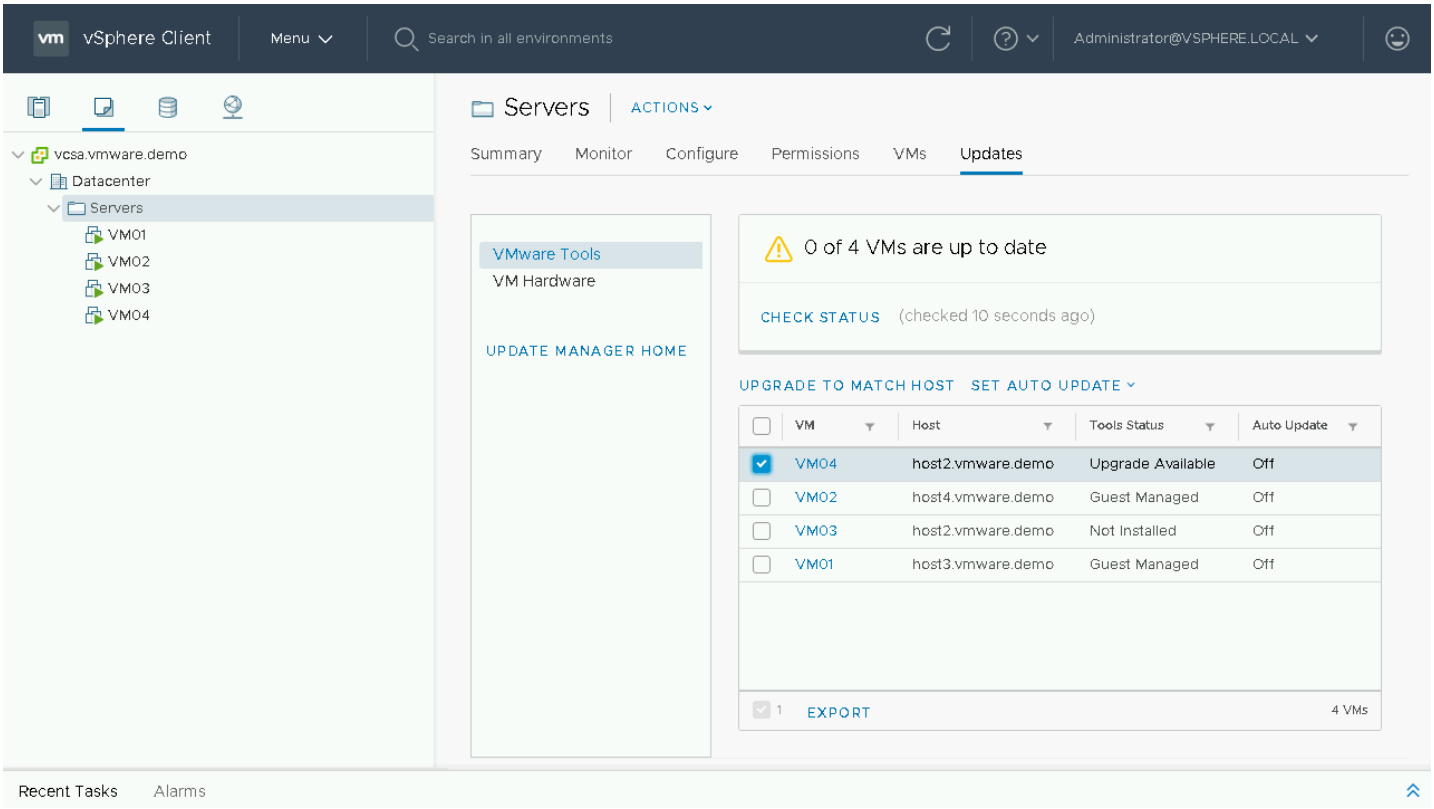
Select the VM Folder or VM Object you wish to update and select the Updates Tab.



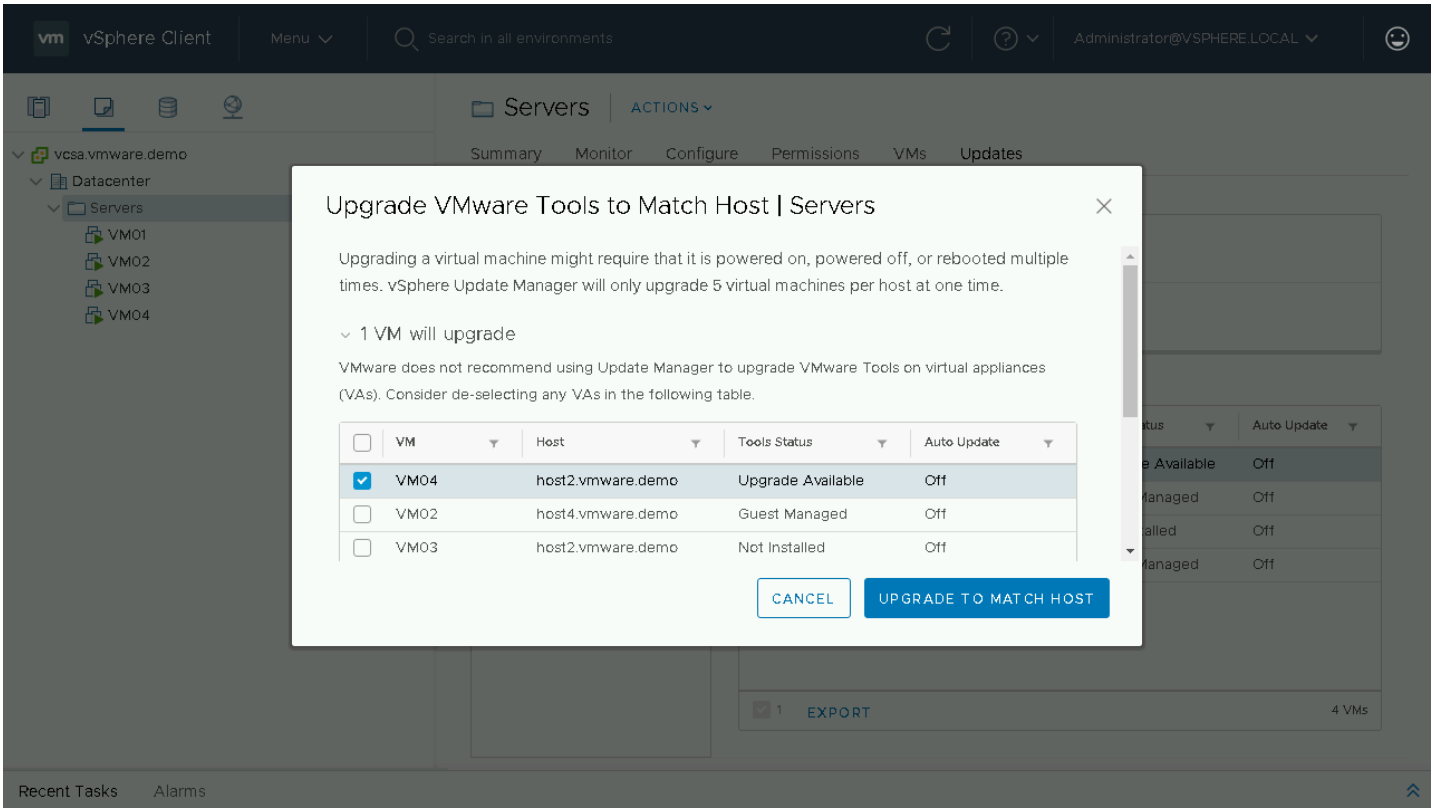
If the status is unknown or not current, click on check status to get the latest versions of tools



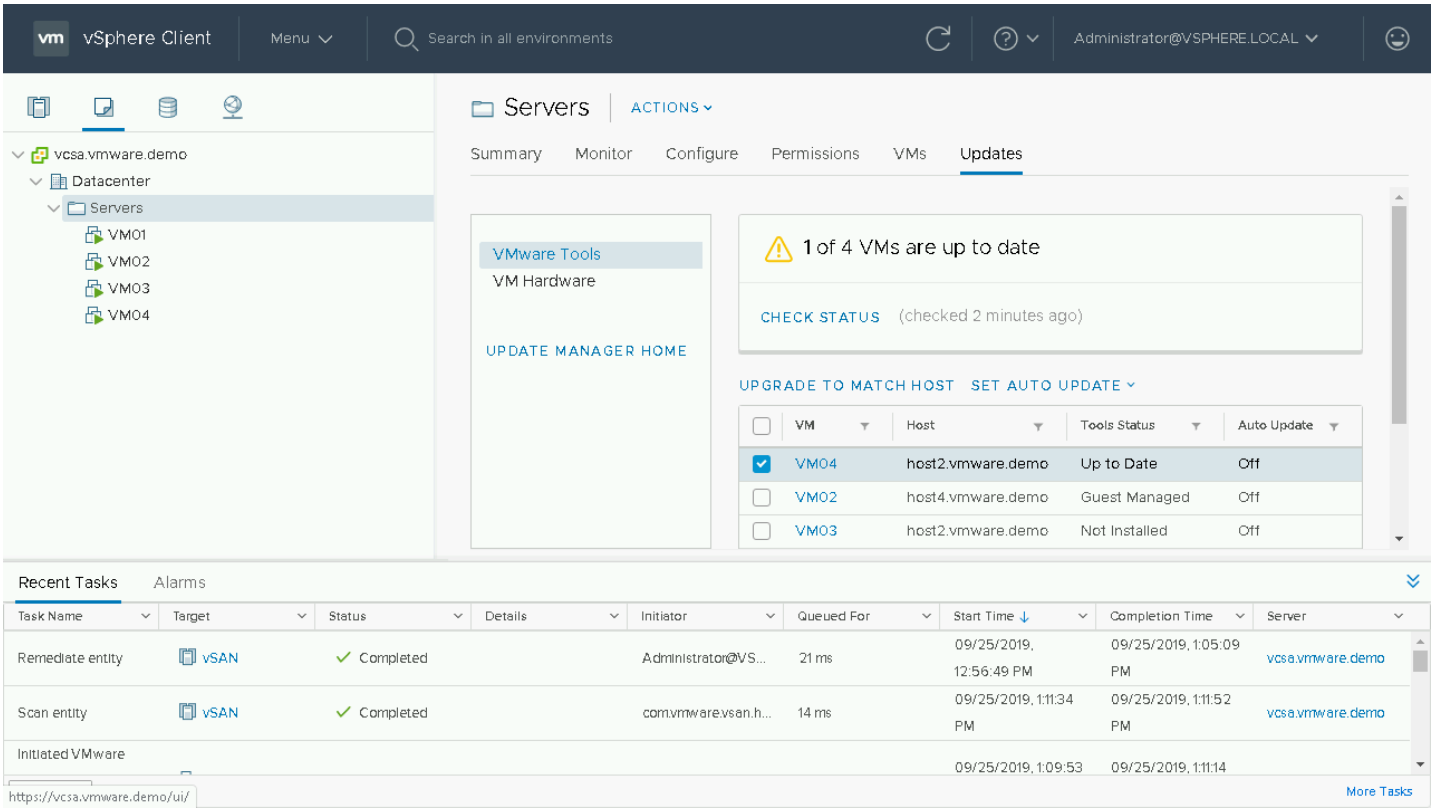
VM04 currently has an upgrade available, so we will select it to perform our upgrade. If tools are not installed or are guest managed they cannot be upgraded through Update Manager



To perform our upgrade we will chose "Upgrade to Match Host"



On the following screen will be a summary of changes being done. We see that 1 VM will be upgraded and that is VM04.

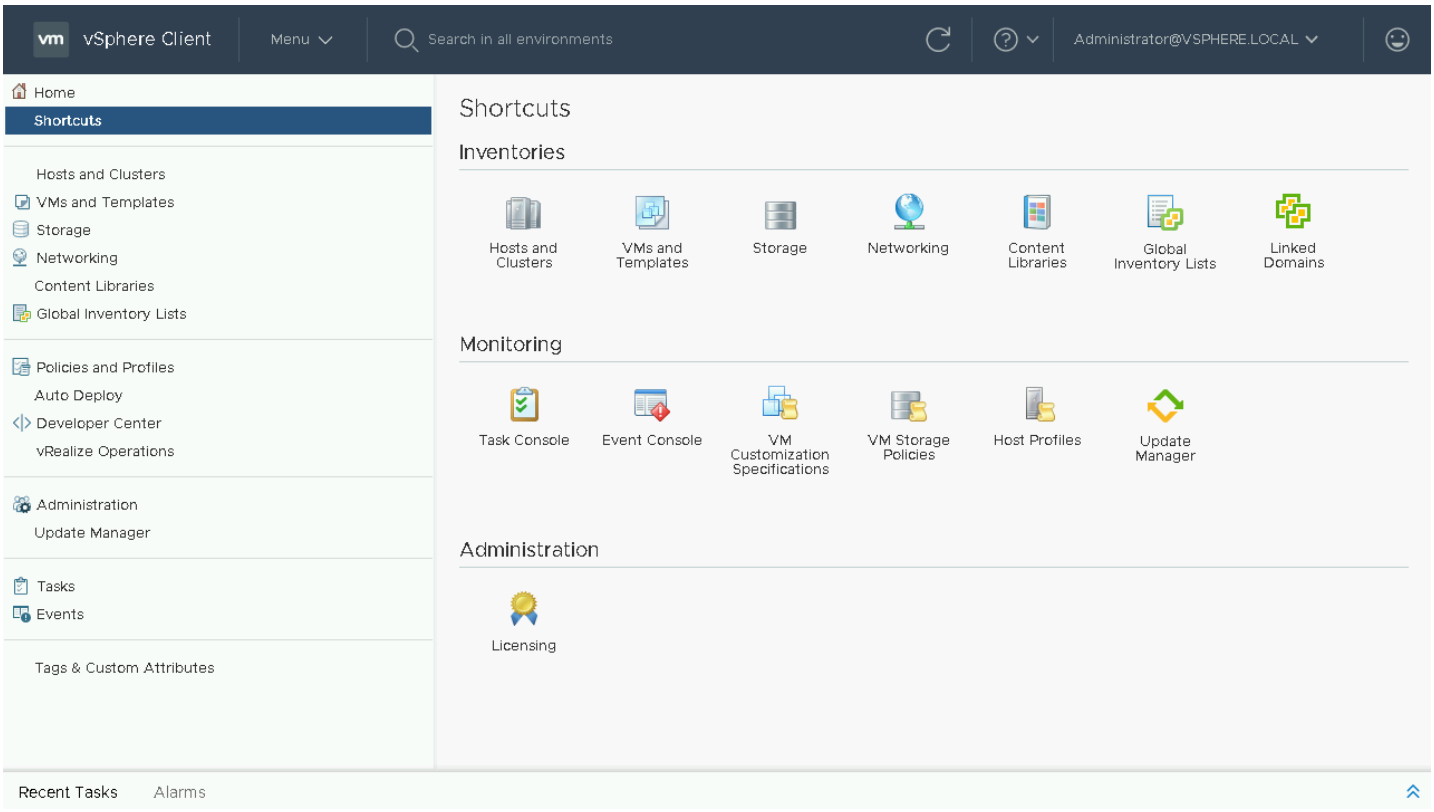


During the upgrade process the VM Tools will be updated and the VM will be rebooted if required.

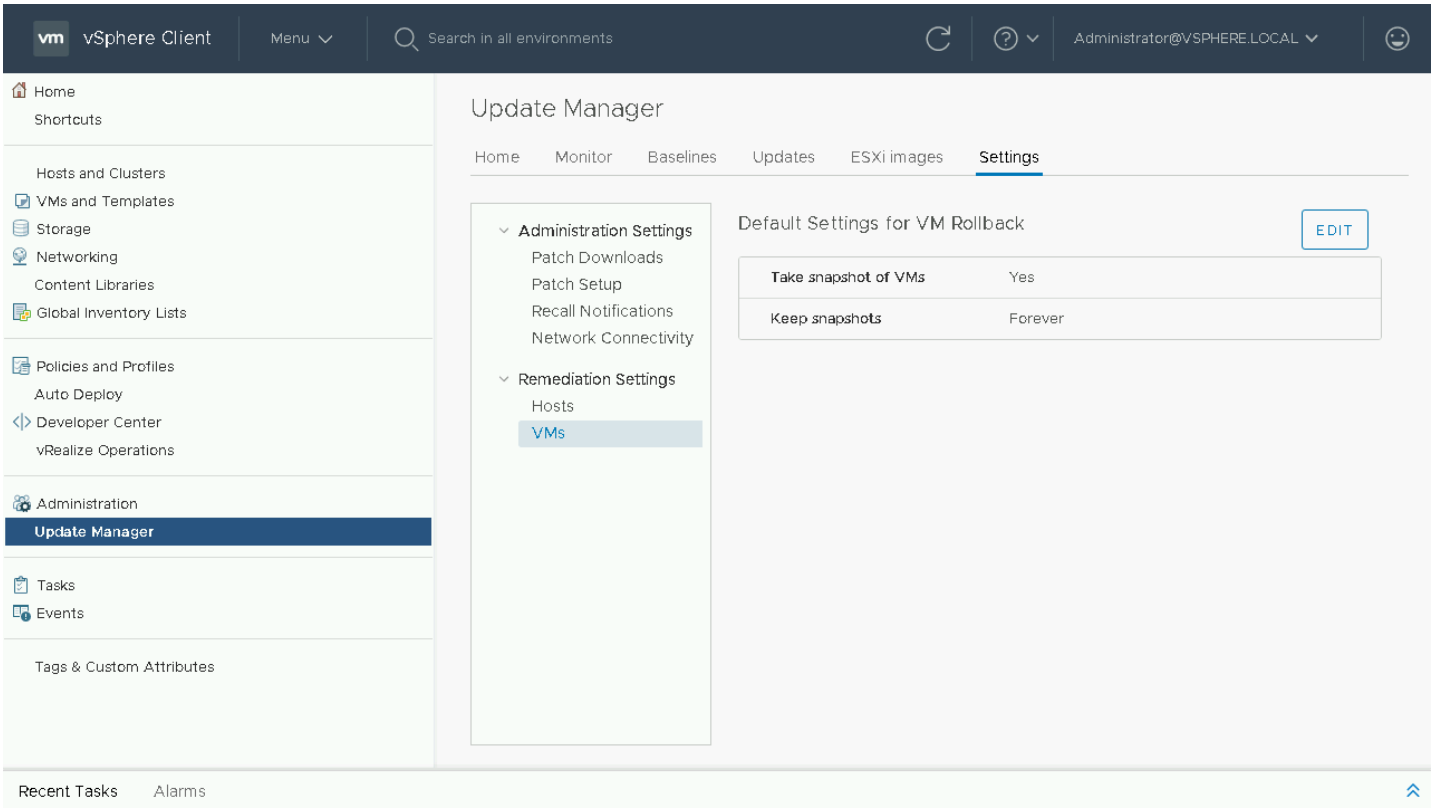
To learn more visit blogs.vmware.com/vsphere

Walkthrough: Upgrading VM Compatibility using vSphere Update Manager

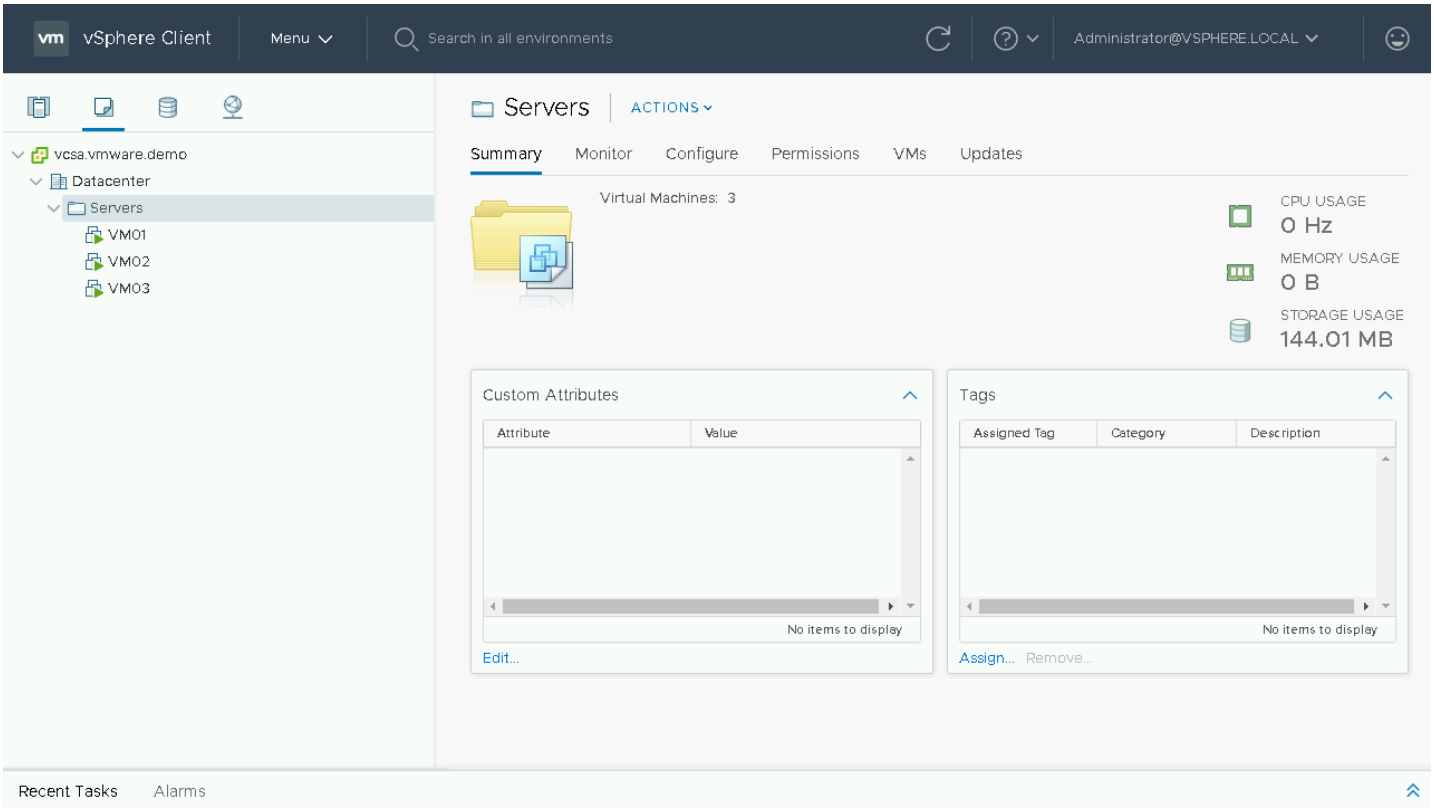
This product walkthrough will cover the steps to upgrade VM Compatibility using vSphere Update Manager



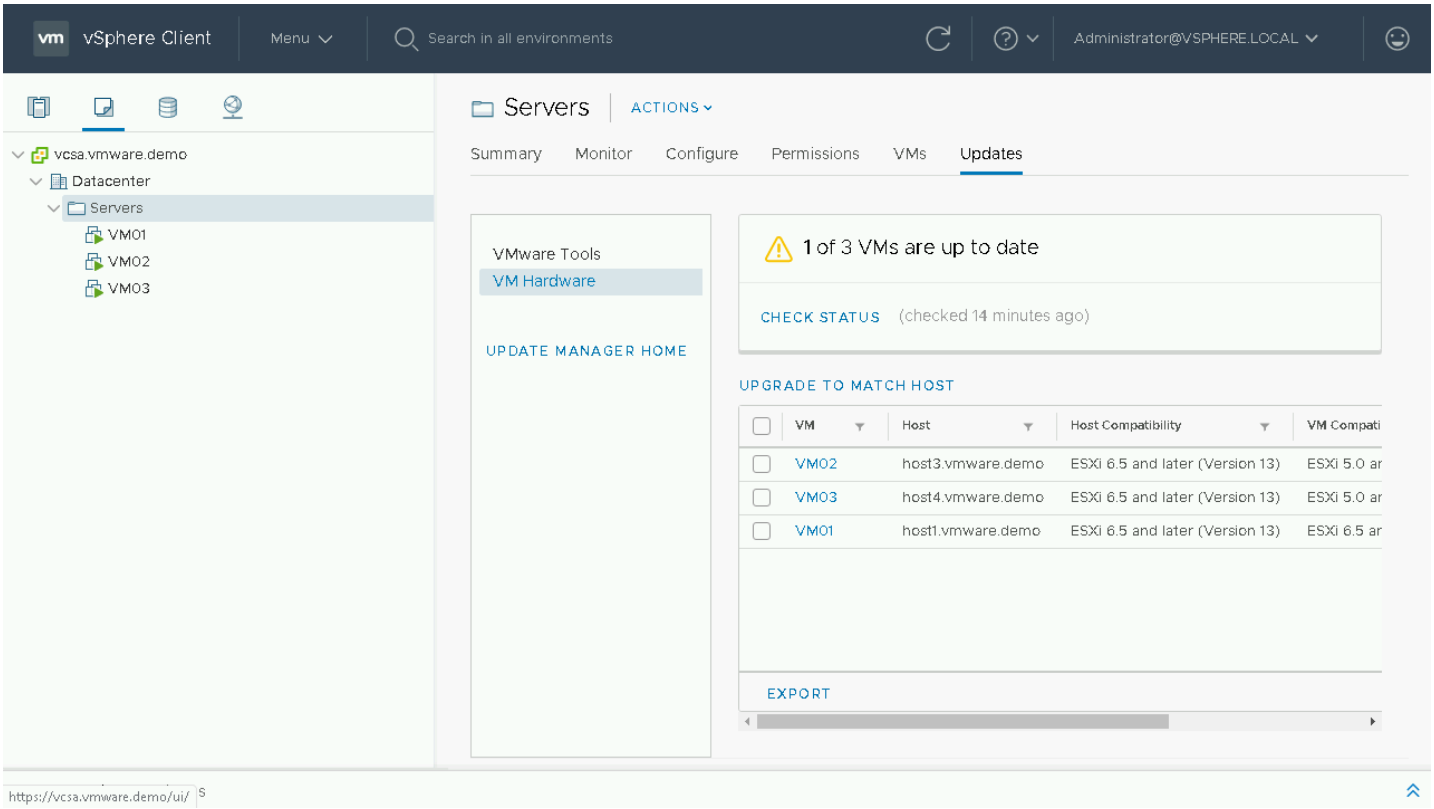
Before we begin the upgrade of VM Compatibility, lets navigate to the Update Manager Settings for VMs



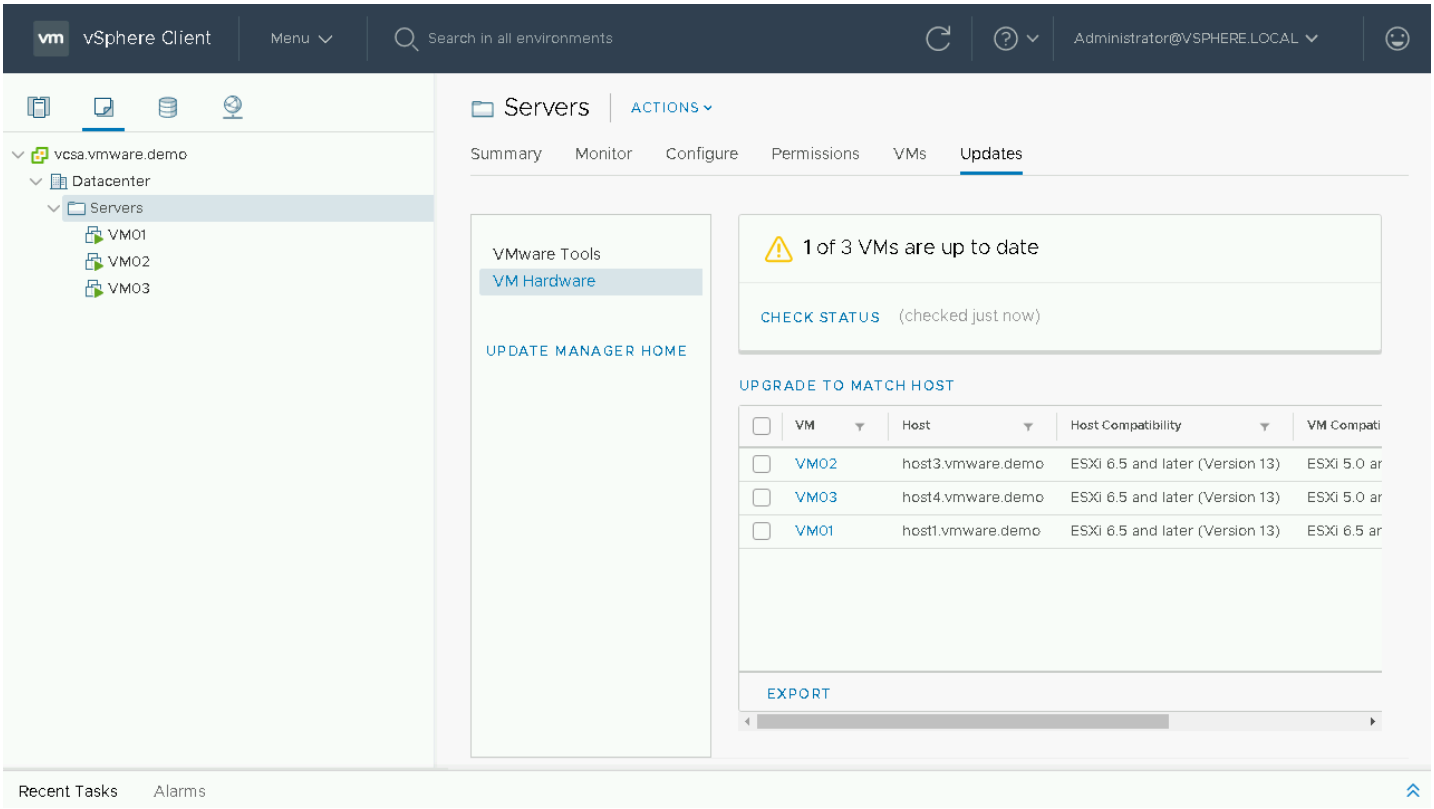
On the current screen you can set the default settings for VM Rollback. This includes whether or not a snapshot should be taken of the VM and the time to retain the snapshot. Let's navigate to VMs and Templates view to begin our VM Compatibility Update.



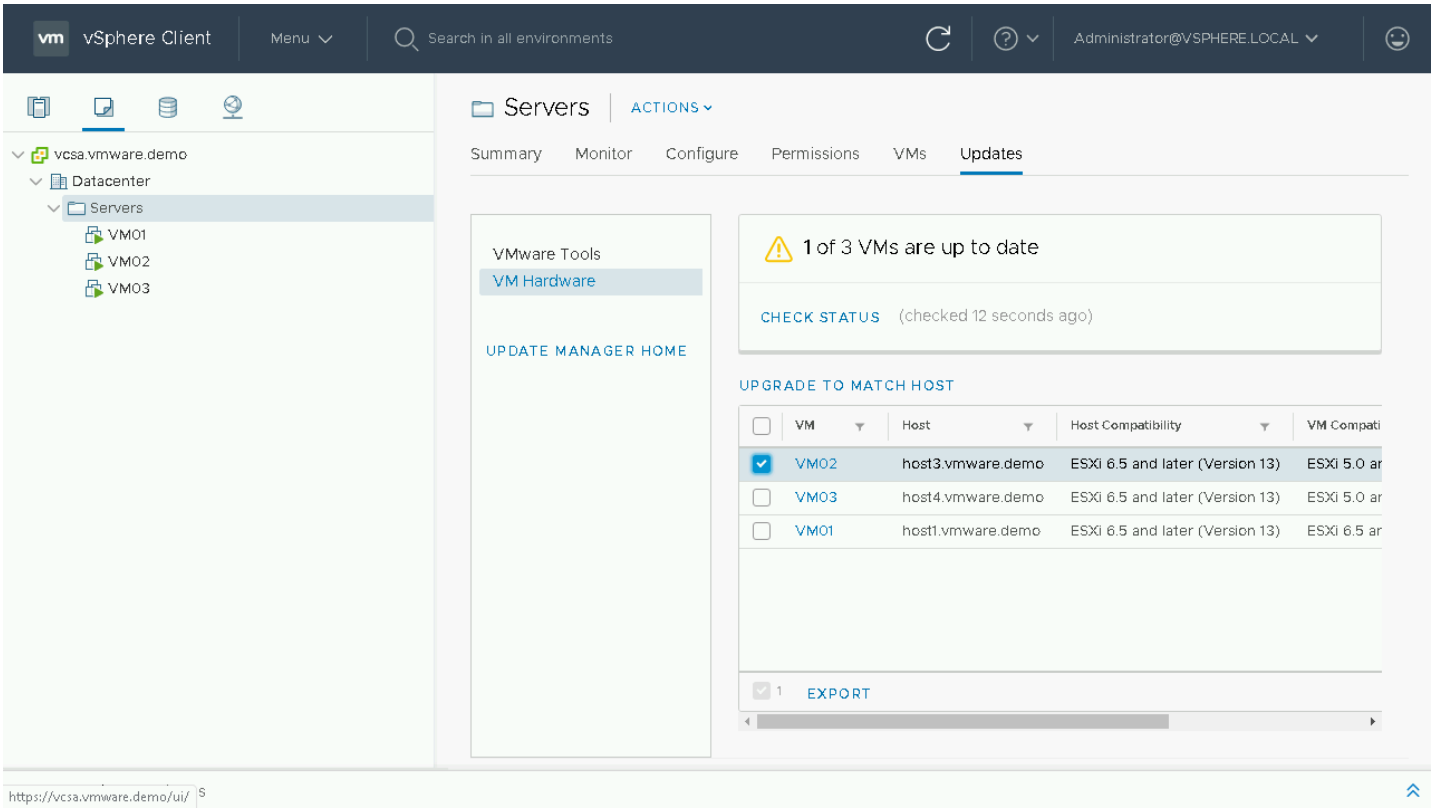
Select the VM Folder or VM Object you wish to update and select the Updates Tab.



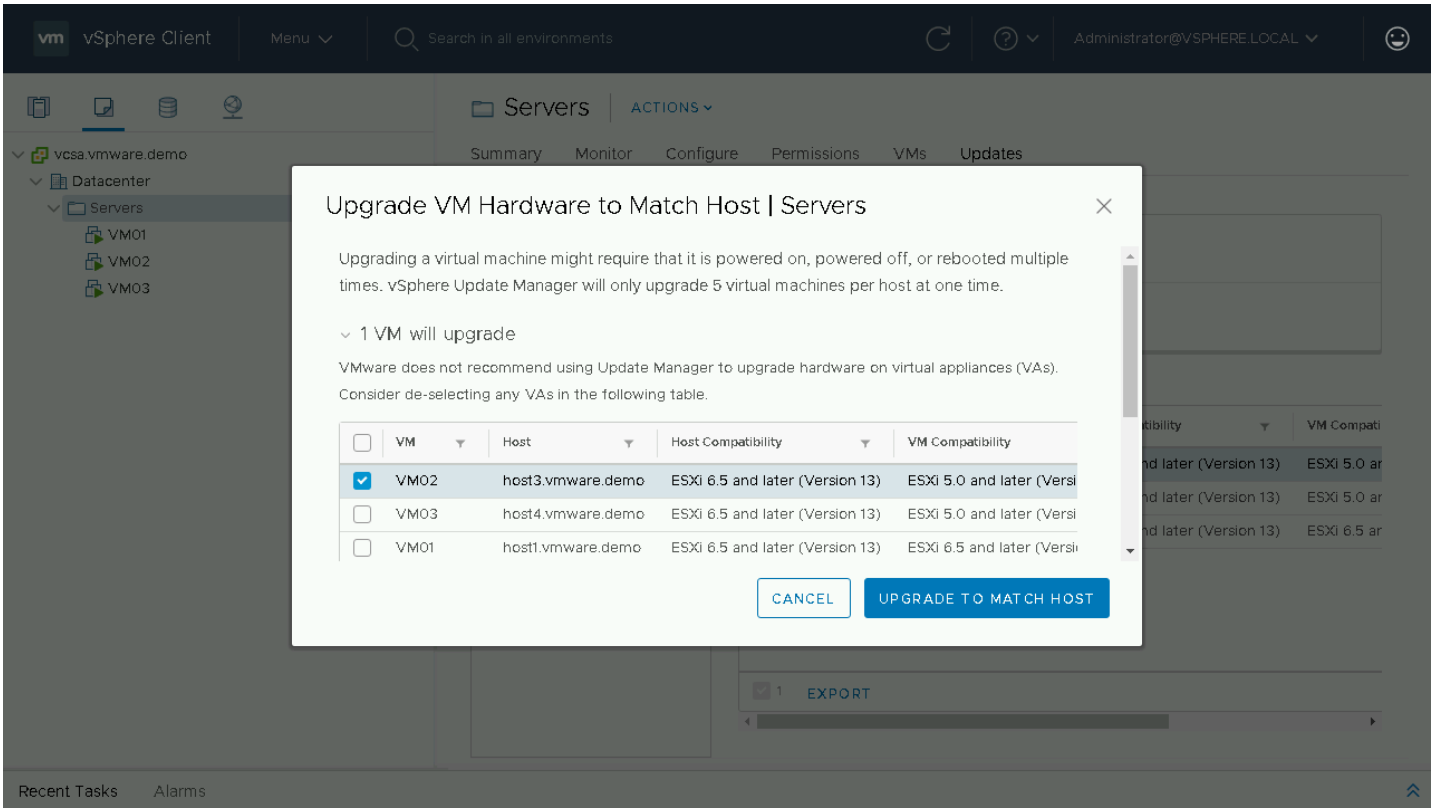
If the status is unknown or not current, click on check status to get the latest versions of compatibility.



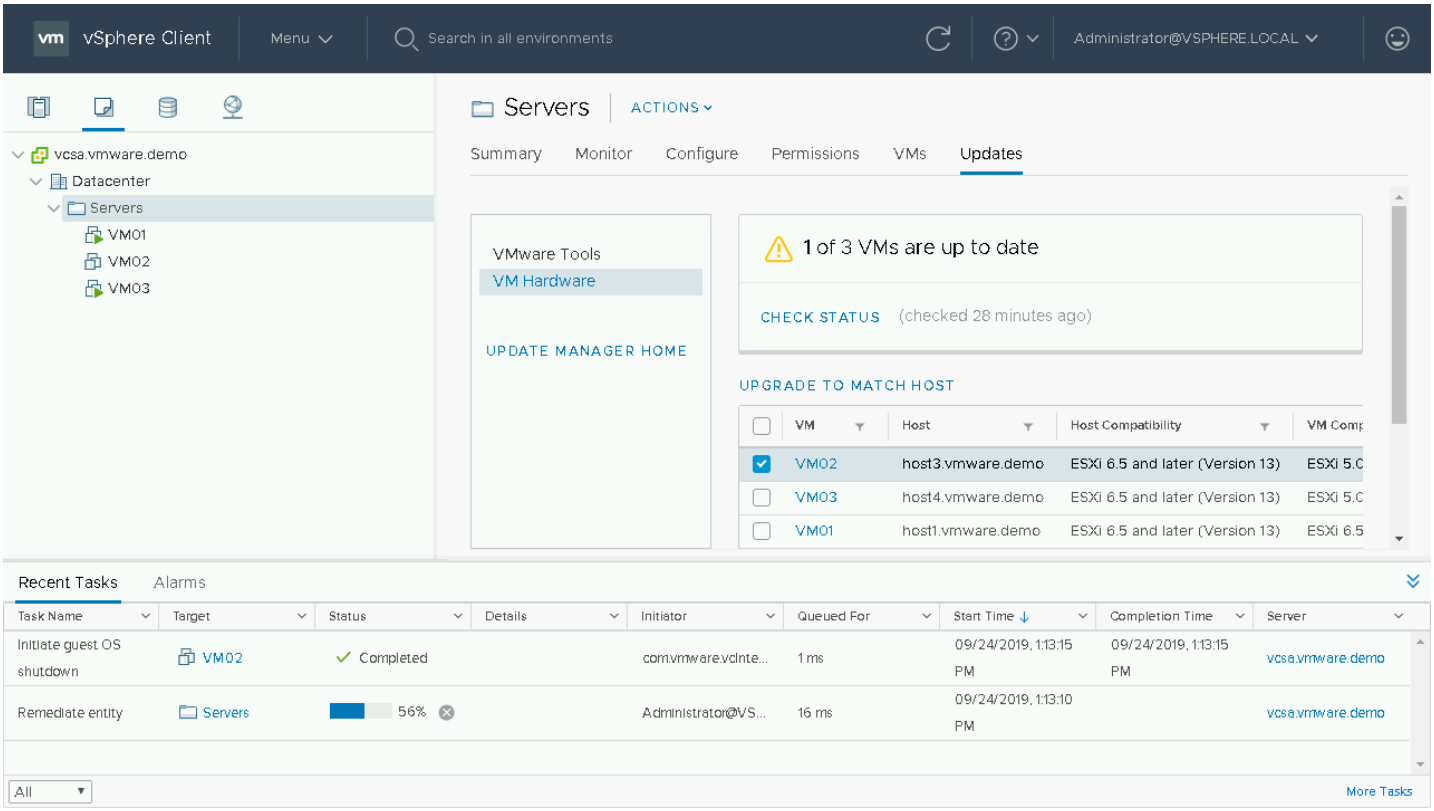
VM02 is currently out of date, so we will select it to perform our upgrade.



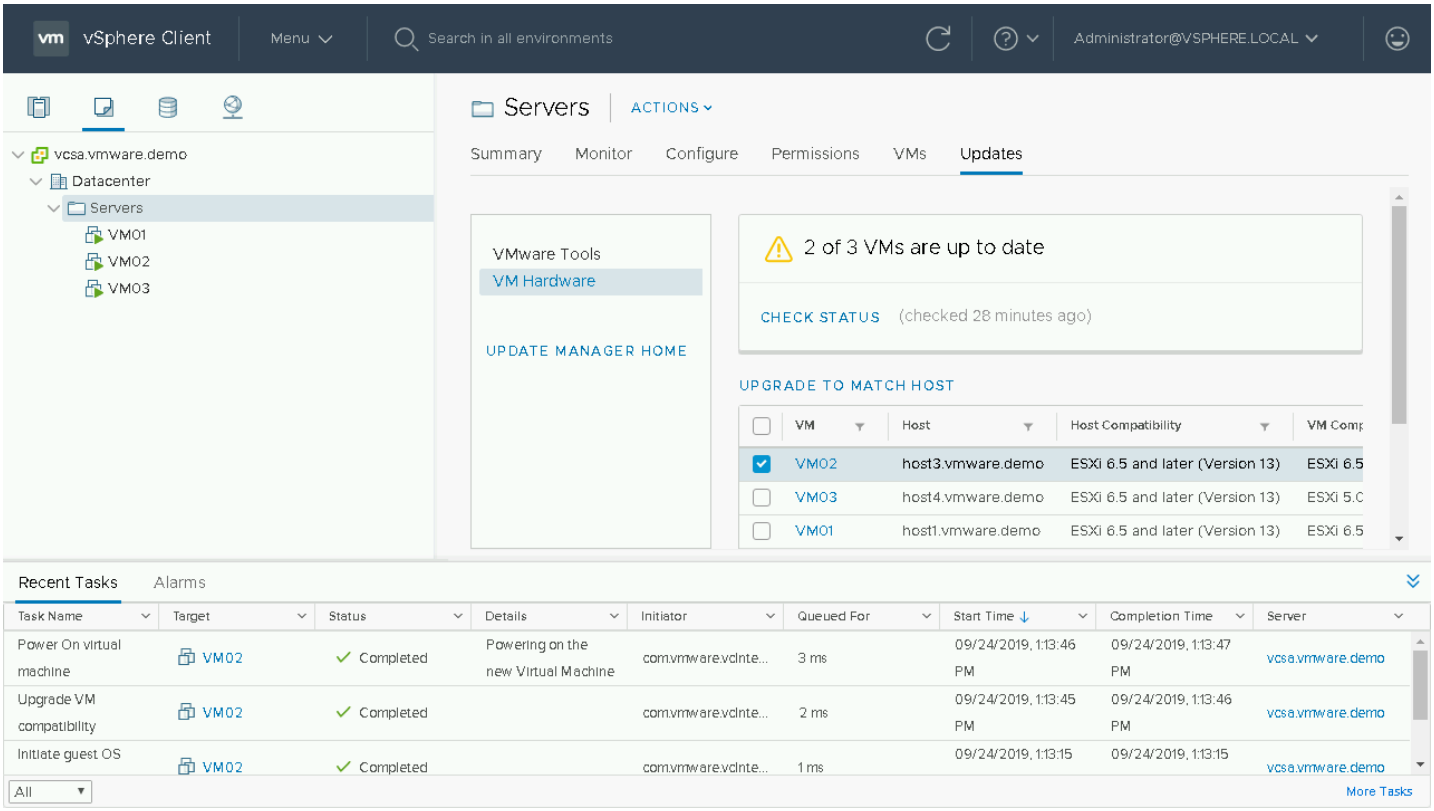
To perform our upgrade we will chose "Upgrade to Match Host"



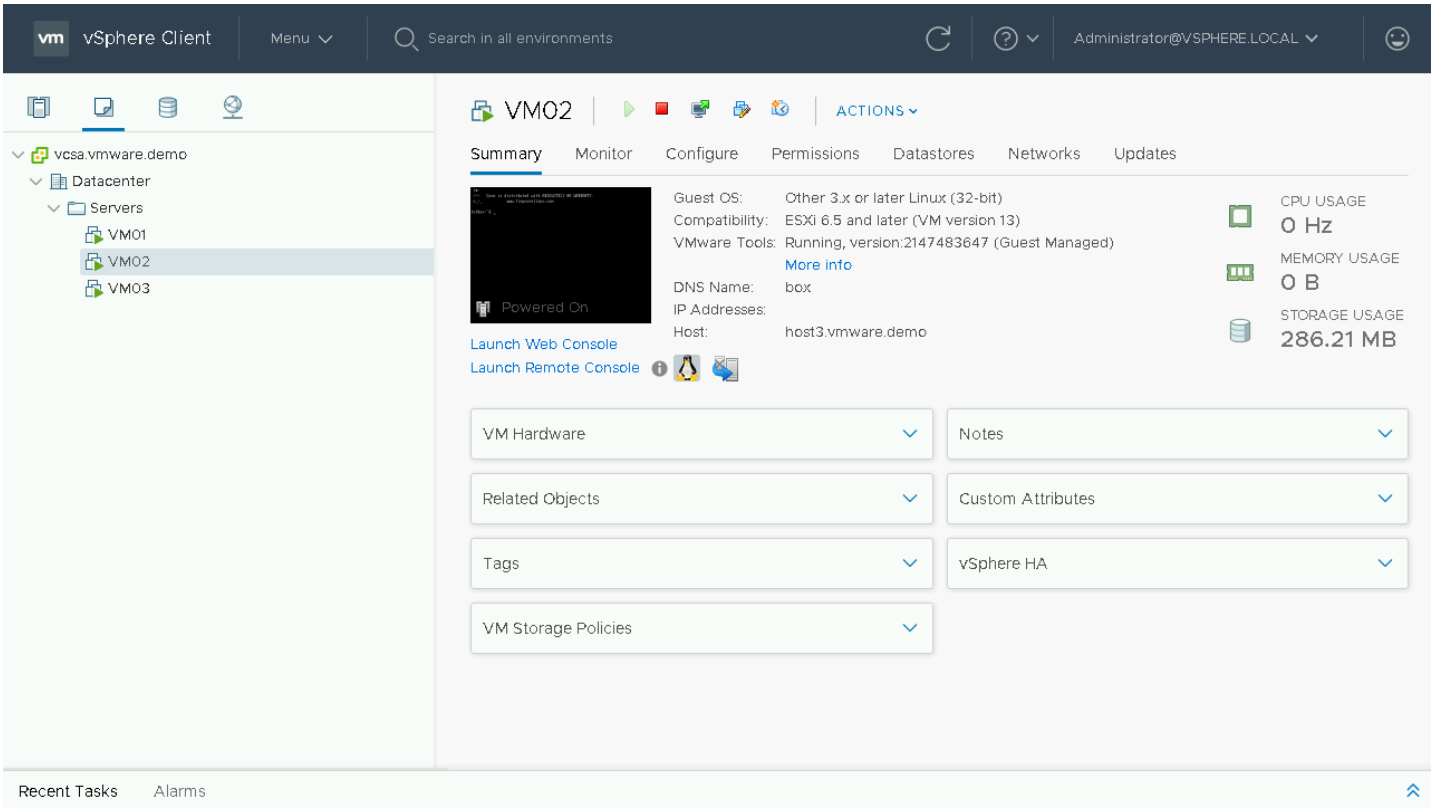
On the following screen will be a summary of changes being done. We see that 1 VM will upgrade and that VM02 is currently VM Version 10 and is going to be upgraded to VM Version 13



During the upgrade process the VM will be shutdown, compatibility upgraded, and then powered back on.



Once the Upgrade is complete, click on the Summary Tab to View the current version of VM Compatibility.



VM Compatibility upgrade is now complete.

Blog: Six Methods for Keeping VM Tools Up to Date

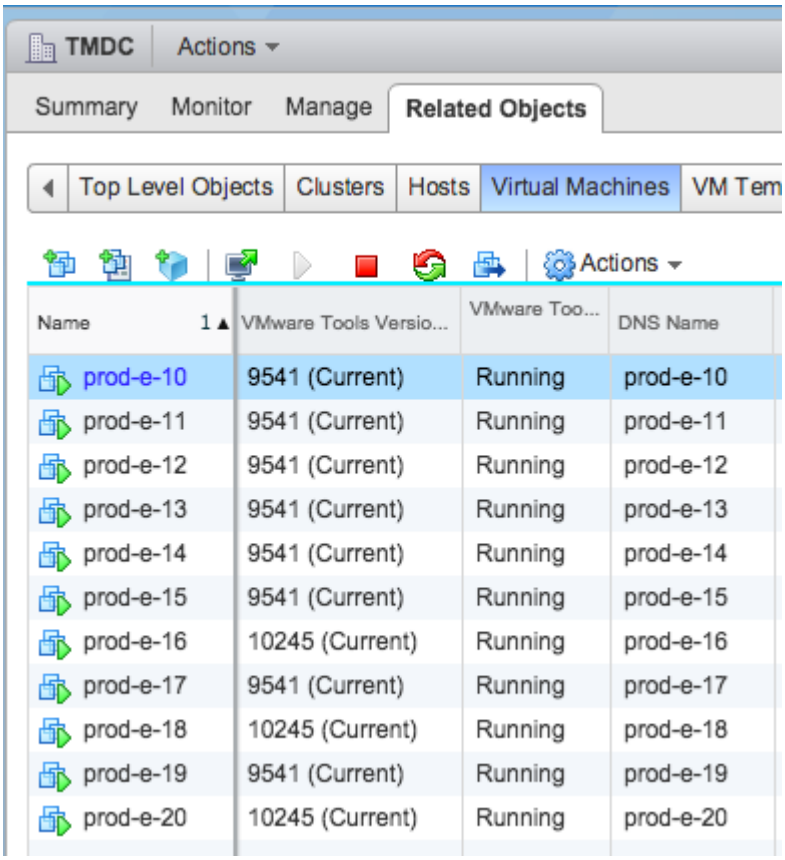
When it comes to keeping VM Tools up to date, there are six different approaches that vSphere administrators can use. That may sound like a lot, but after seeing each of the various options, it is clear that the intention is to accommodate nearly any workflow customers require for flexible datacenter operations. These different techniques allow optimizing either for automation and standardization or for separation of responsibilities. A previous article provides an overview of the three types of VM Tools.

VMware Tools Status is Relative to Underlying Host

Recall that each ESXi host has a storage location for VM Tools installers, which is a configurable option and visibly referenced by the /productLocker symlink. The target can be either local to each host or point to a centralized repository of VM Tools on a shared datastore. For more information about setting up a shared Tools repository, see this earlier post or KB 2004018.

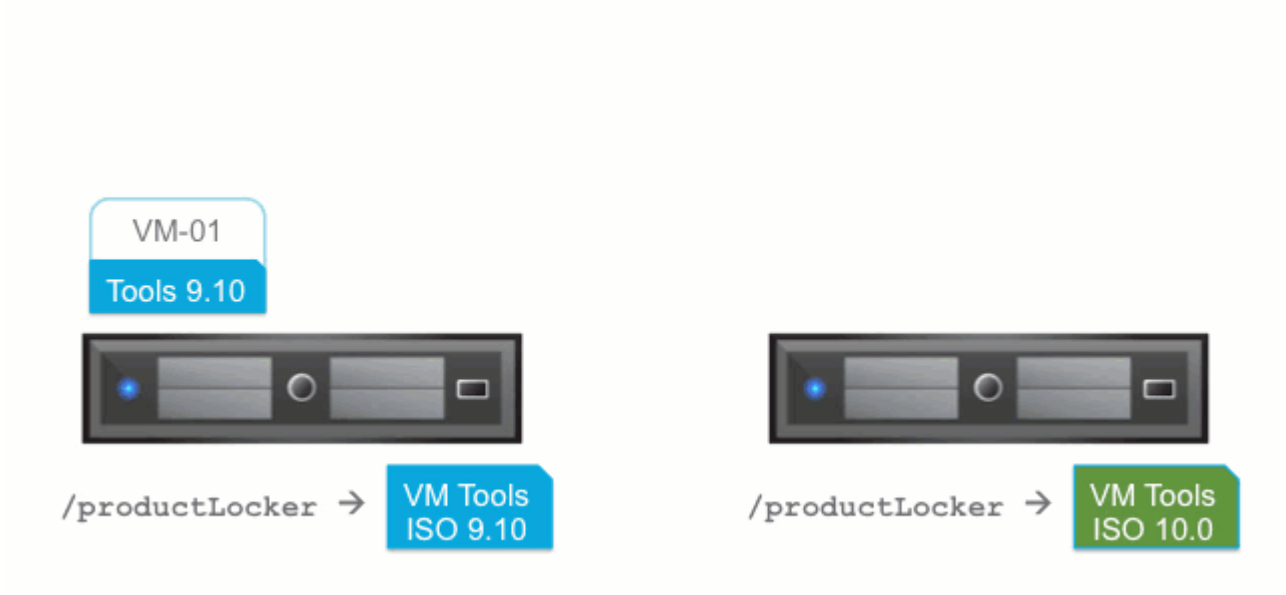
```
productLocker -> /vmfs/volumes/NFS-C/vmtools_shared
```

VM Tools status for any given VM is always in the context of the underlying host. As demonstrated below, two different versions of Tools are considered “Current” because the underlying hosts are not identical.



When certain virtual machine events occur, such as power-on or vMotion, the version of VM Tools running on that guest is compared to the version associated with the underlying ESXi host. If the host has a newer version, the VM is considered out of date.

Note that there is no mechanism for VMs running on vSphere to *contact the mother ship* and learn about new versions of VM Tools – only the VM to host relationship is relevant. This explains why a VM may suddenly complain about having out of date VM Tools after migrating from one host to another – the destination host has a more recent version in the product locker.



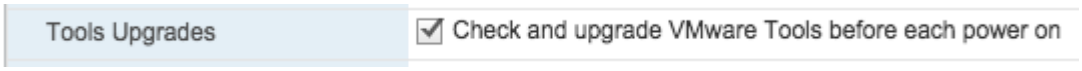
VM Tools Type Determines Update Choices

Recall from the previous post that there are three types of VM Tools – the familiar Tools ISOs for all supported operating systems, plus two additional offerings in the form of binary packages for Linux. There are several ways to initiate VM Tools updates from vSphere or from within a guest. The following applies **only** to Windows and Linux guests using VM Tools ISOs except where noted. The VM Tools Linux packages – OVT and OSP – are not managed via vSphere, so they can only be installed and updated from within each guest OS using native package management tools.

Six Ways to Keep VM Tools Updated

1. Automatic update on VM boot

The simplest way to keep VM Tools up to date is to check a box and forget about managing this element of infrastructure. Upon VM reboot, such as after installing guest OS patches, VM Tools status will be checked and an update installed if needed. In many cases, this will result in one additional reboot after VM Tools installation completes.



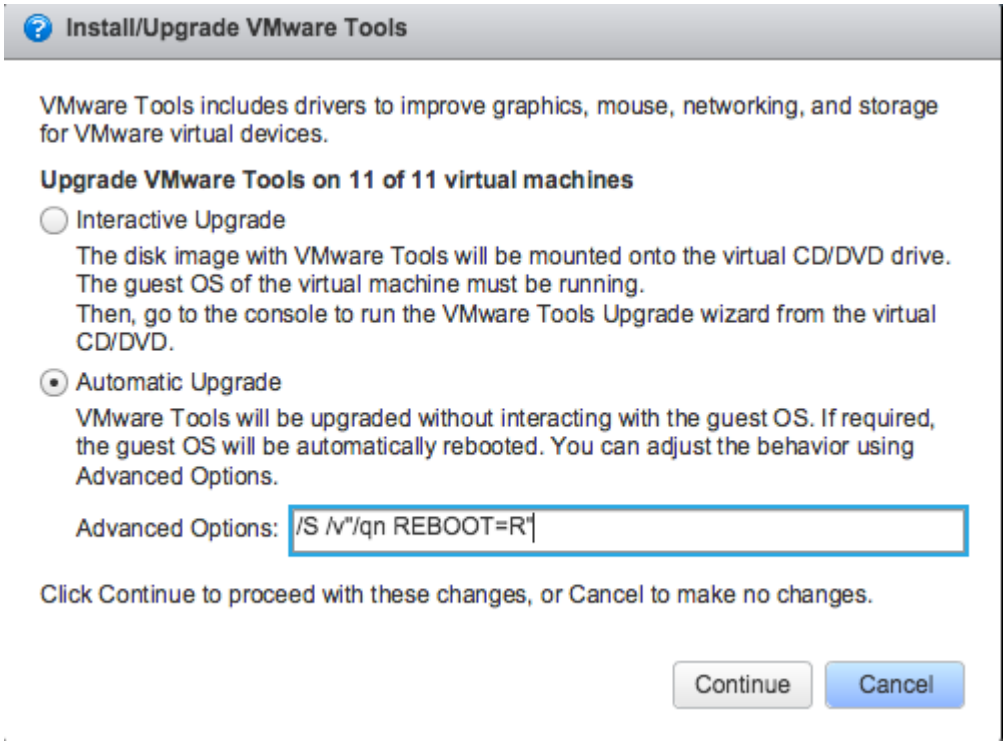
This approach may be viable for less-critical workloads, perhaps labs or test/dev environments. Imagine a scenario where VMs are rebooting unexpectedly due to a widespread infrastructure outage. After scrambling to get applications back online, administrators could find themselves facing unanticipated subsequent reboots if a VM Tools update happened to be available. This is an edge case, but one to keep in mind.

2. Initiate update to one or more VMs through the vSphere UI

In the vSphere Web Client, when a VM indicates that VM Tools are outdated an adjacent button can be used to automatically initiate an update. This can be done interactively or in a completely hands-off fashion. In the latter case, administrators also have the option of suppressing any potential reboots on Windows VMs – this is a good option that enables coordination of reboots required after routine guest OS patching.

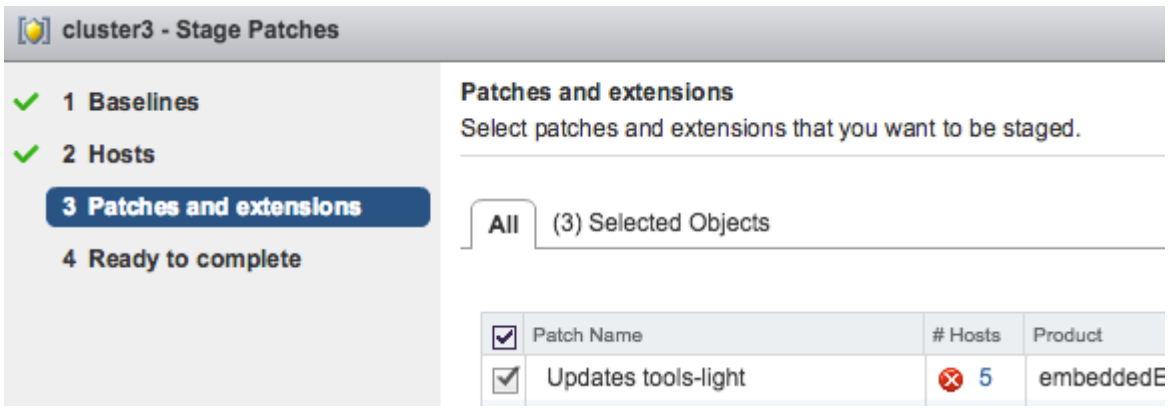
Important note for guests other than Windows and Linux: Solaris, FreeBSD, and Mac OS VM Tools can only be updated using the manual interactive method. There is currently no automatic Tools update for these guests.

Going a step further, it is also possible to select multiple VMs in the Web Client UI and initiate a VM Tools update on all of them at once.

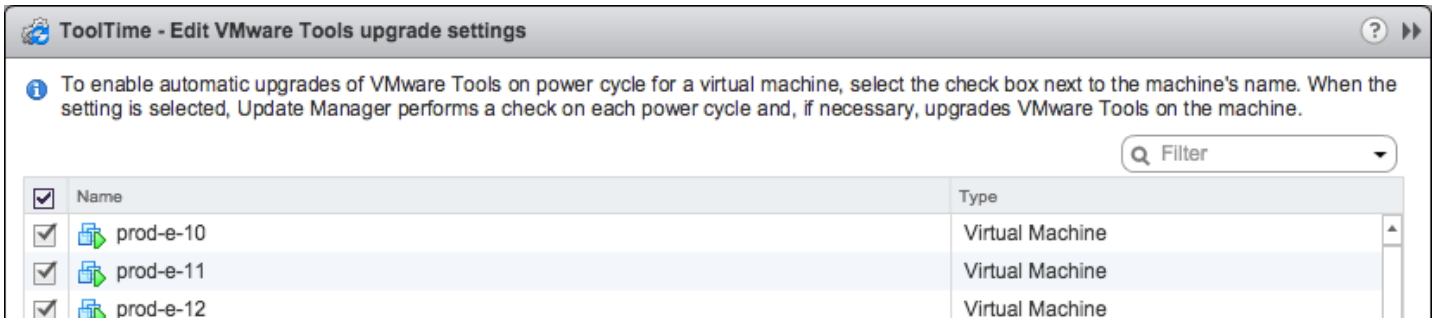


3. VMware Update Manager: immediate, scheduled, or on boot

VMware Update Manager (VUM) has two very different roles to play when it comes to updating VM Tools. The first one has to do with fetching updated VM Tools ISOs in the form of the 'tools-light' VIB that is offered when needed in the normal ESXi patch stream. This patch is then pushed out to all managed hosts according to baselines established by administrators. Once this occurs, individual VMs will begin to detect that a new version of VM Tools is available and will be eligible for update.



The second role VUM has in managing VM Tools is to trigger updates for individual VMs according to baselines. Keep in mind that VUM does this work by leveraging the vSphere methods described in the two previous sections. In one mode, VUM can be used to make a bulk configuration change to multiple VMs so that a Tools update is checked and performed as necessary on each guest reboot, just like an administrator can do using the technique shown in #1 above. The advantage of using VUM is that many VMs can be configured or un-configured for this option at once.



The other mode VUM uses is to trigger a VM Tools update either immediately or at a scheduled time, just as an administrator can do manually as described in #2 above. One added benefit of using VUM to initiate these updates in this way is the ability to also

remediate powered-off or suspended VMs, subsequently returning them to their initial state after update.

4. In-guest update – delegating control to app owners

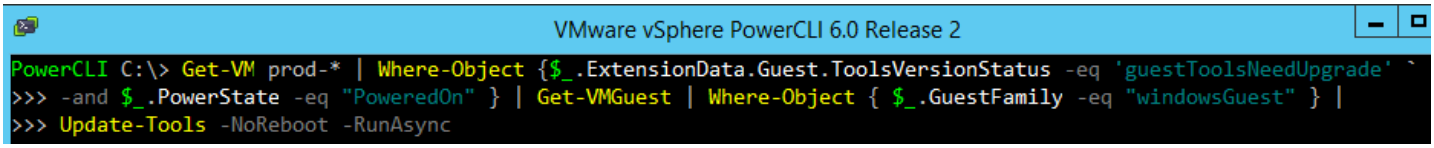
For scenarios where application owners demand tight control over activities that occur in the guest OS, there is an option to allow in-guest updates of VM Tools. A tray icon in Windows will indicate that an update is available, and the VM Tools configuration dialog box will permit interactive initiation of an update at a convenient time.

For equivalent functionality from a command-line utility, vmware-toolbox-cmd is offered for Linux as well as Windows guests. Keep in mind that for Linux this is only for the VM Tools ISOs, since OVT and OSP use a different process, as described in #6 below.

Enable guest-initiated updates by modifying the **isolation.tools.guestInitiatedUpgrade.disable** VM setting, which can be done easily to one or more VMs with PowerCLI:

5. Mass updates through PowerCLI automation

In very large environments or for those that have established more mature operational processes, PowerCLI provides a powerful option for updating VM Tools. This approach can target particular groups of VMs in many convenient ways, such as by cluster, by guest OS version, tags, VM state, or other vSphere attributes.



6. Native Linux package management processes

By nature of their design, Linux guests running OSPs or OVTs update VM Tools as part of a broader patching and updating workflow used for other components. This allows administrators to leverage existing Linux package managers or broader patch management and monitoring solutions without need to coordinate with vSphere administrators.

7. BONUS: VM Tools Upgrade Method (pun intended)

For advanced use cases where vSphere APIs are being used for deeper integration with other processes, consider the UpgradeTools_Task for programmatic upgrades of VM Tools.

Summary

With these flexible means of updating VM Tools, there is a suitable approach for any VMware datacenter, whether the requirement is centralized control, automation, delegation to app owners, or integration with existing patch management processes.

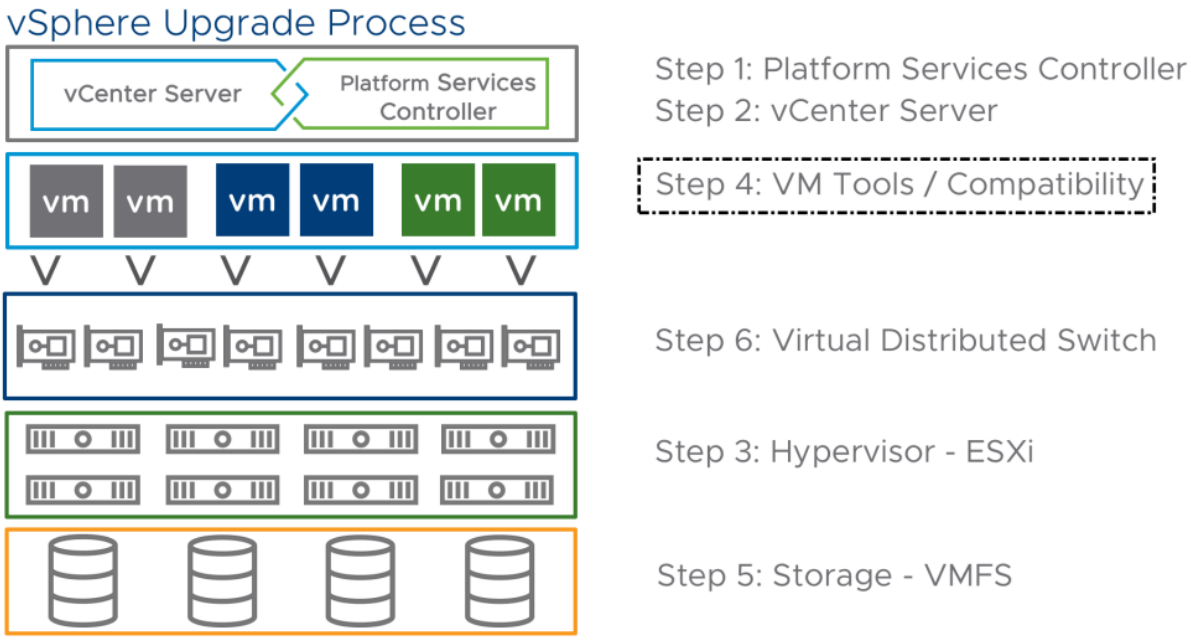
Article: https://blogs.vmware.com/vsphere/2016/03/six-methods-for-keeping-vm-tools-up-to-date.html

Date: 2018-10-29

Blog: Automating Upgrade of VMware Tools and VM Compatibility

Automating Upgrade of VMware Tools and VM Compatibility

Next up in our Automating your vSphere Upgrade blog series is your VMware Tools and VM Compatibility. Upgrading these both have different requirements so we will cover when and how you should upgrade your VMware Tools and VM compatibility in the below post.



Upgrading VMware Tools

When it comes to keeping your VMware Tools up to date we have a few options but I will focus on two of my favorite methods. Keeping VMware tools up to date is very important as VMware includes drivers and tools to make sure your VM's run optimally with the latest features.

Eric Gray of our CPBU Technical Marketing Team covers a few considerations and methods in depth in the following blog post Six Methods for Keeping VM Tools Up to Date.

Checking VMware Tools Compliance

Before we can start to update our VMware Tools it might be a good idea to understand which VMs in our environment are currently out of compliance, the easiest way to check if a VM if out of compliance is to view it via the vSphere Client and it will show you details such as the version and compliance.

However, since we are talking about automation lets show this another way using PowerCLI.

Using this PowerCLI one-liner we are able to see more information at scale, we can see that within our folder we actually have 3 VMs that are out of compliance. To remediate these to the latest version its actually quite simple with PowerCLI as well.

```
PS C:\Users\dstamen> Connect-VIServer upg-vcsa-01.cpbu.lab

Name                           Port  User
----                           ----  ----
upg-vcsa-01.cpbu.lab           443   CPBU\dstamen


PS C:\Users\dstamen> Get-Folder -name Testing | Get-VM | % { get-view $_.id } | select name, @{Name="ToolsVersion"; Expressi
on={$_.config.tools.toolsversion}}, @{ Name="ToolStatus"; Expression={$_.Guest.ToolsVersionStatus}}|Sort-Object Name

Name ToolsVersion ToolStatus
---- ------------ -----------
VM01         10287 guestToolsNeedUpgrade
VM02         10287 guestToolsNeedUpgrade
VM03         10287 guestToolsNeedUpgrade
VM04         10305 guestToolsCurrent
VM05         10305 guestToolsCurrent
```

To make the information clearer on what needs an update we will use the same search only looking for VMs where ToolStatus is guestToolsNeedUpgrade. To do that we will use the following one-liner.


```
PS C:\Users\dstamen> Get-Folder -name Testing | Get-VM | % { get-view $_.id } |Where-Object {$_.Guest.ToolsVersionStatus -lik
e "guestToolsNeedUpgrade"} |select name, @{Name="ToolsVersion"; Expression={$_.config.tools.toolsversion}}, @{ Name="ToolSt
atus"; Expression={$_.Guest.ToolsVersionStatus}}| Sort-Object Name

Name ToolsVersion ToolStatus
---- ------------ -----------
VM01         10287 guestToolsNeedUpgrade
VM02         10287 guestToolsNeedUpgrade
VM03         10287 guestToolsNeedUpgrade
```

Upgrading VMware Tools via PowerCLI

Now that we know which VMs need tools update we can actually go forth and upgrade tools. That is quite simple we can do this using the Update-Tools PowerCLI cmdlet. Using our existing one-liner from above we will just append Update-Tools to update the VMs with tools currently out of compliance.


```
PS C:\Users\dstamen> Get-Folder -name Testing | Get-VM | % { get-view $_.id } |Where-Object {$_.Guest.ToolsVersionStatus -li
ke "guestToolsNeedUpgrade"} |select name, @{Name="ToolsVersion"; Expression={$_.config.tools.toolsversion}}, @{ Name="ToolSt
atus"; Expression={$_.Guest.ToolsVersionStatus}}| Update-Tools -NoReboot -VM {$_.Name} -Verbose
VERBOSE: 9/27/2018 8:54:09 AM Update-Tools Started execution
VERBOSE: Performing operation "Update VMware Tools." on VM "VM03".
VERBOSE: 9/27/2018 8:54:10 AM Update-Tools Finished execution
VERBOSE: 9/27/2018 8:54:10 AM Update-Tools Started execution
VERBOSE: Performing operation "Update VMware Tools." on VM "VM02".
VERBOSE: 9/27/2018 8:54:10 AM Update-Tools Finished execution
VERBOSE: 9/27/2018 8:54:10 AM Update-Tools Started execution
VERBOSE: Performing operation "Update VMware Tools." on VM "VM01".
VERBOSE: 9/27/2018 8:54:10 AM Update-Tools Finished execution
```

You may have noticed that all the VMs had their updates kicked off at the same time and this may not be ideal, this is one way that the Update-Tools cmdlet works, however we can get around this by storing the VMs within a variable and then using a loop process them one at a time. This is definitely preferable as it will limit the impact to the environment.


```
PS C:\Users\dstamen> $OutOfDateVMs = Get-Folder -name Testing | Get-VM | % { get-view $_.id } |Where-Object {$_.Guest.ToolsV
ersionStatus -like "guestToolsNeedUpgrade"} |select name, @{Name="ToolsVersion"; Expression={$_.config.tools.toolsversion}},
 @{ Name="ToolStatus"; Expression={$_.Guest.ToolsVersionStatus}}| Sort-Object Name
PS C:\Users\dstamen> $OutOfDateVMs

Name ToolsVersion ToolStatus
---- ------------ ----------
VM01        10287 guestToolsNeedUpgrade
VM02        10287 guestToolsNeedUpgrade
VM03        10287 guestToolsNeedUpgrade


PS C:\Users\dstamen> ForEach ($VM in $OutOfDateVMs){Update-Tools -NoReboot -VM $VM.Name -Verbose}
VERBOSE: 9/27/2018 9:10:12 AM Update-Tools Started execution
VERBOSE: Performing operation "Update VMware Tools." on VM "VM01".
VERBOSE: 9/27/2018 9:10:12 AM Update-Tools Finished execution
VERBOSE: 9/27/2018 9:10:44 AM Update-Tools Started execution
VERBOSE: Performing operation "Update VMware Tools." on VM "VM02".
VERBOSE: 9/27/2018 9:10:44 AM Update-Tools Finished execution
VERBOSE: 9/27/2018 9:11:19 AM Update-Tools Started execution
VERBOSE: Performing operation "Update VMware Tools." on VM "VM03".
VERBOSE: 9/27/2018 9:11:19 AM Update-Tools Finished execution
```

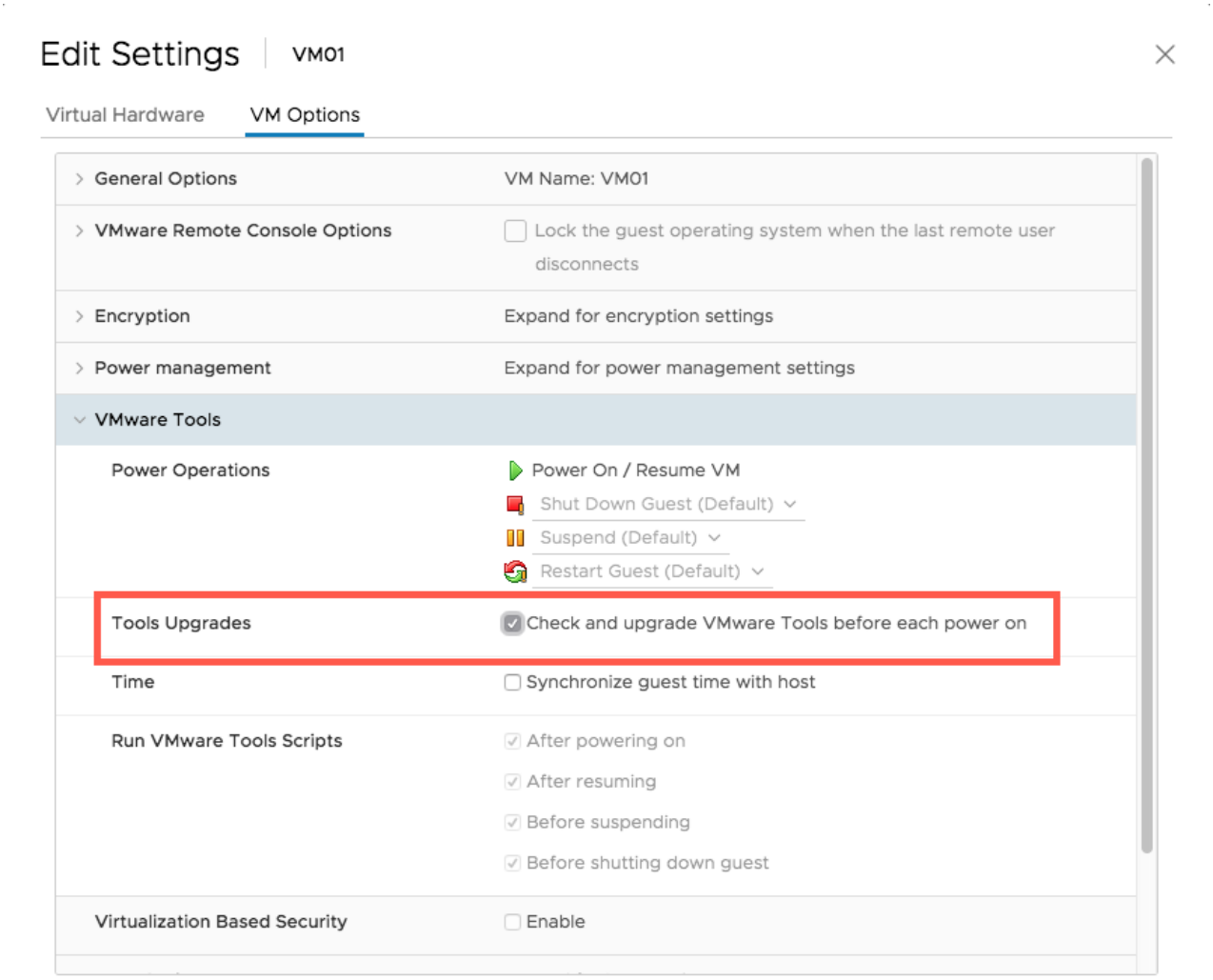
Now our tools are up to date!

For more information on upgrading your VMware Tools via PowerCLI you can find more information here.

Upgrading Tools Automatically on Reboot

Using VM options to keep VMware Tools up to date is also another method to automatically keep VMs up to date. Enabling the “Check and upgrade VMware Tools before each power on” advanced setting use to not be used because of the additional reboot it would cause for virtual machines. Keep in mind that with Windows Server 2016 VMware Tools no longer need a reboot on upgrade, it can be safe to enable this setting and have VMs stay up to date on every reboot. However this may not be applicable to all situations, so another recommendation would be to enable this for a lab environment or non-critical workloads.

The easy way to enable this option is to log into the vSphere Client, edit the VM settings and enable the setting.



After all this a post on automation, so lets see if we can find a way to use PowerCLI to modify the VM's setting so this gets easier when we have a large environment.

```
PS C:\Users\dstamen> Get-Folder Testing|Get-VM|Get-View | select name,@{N='ToolsUpgradePolicy';E={$_.Config.Tools.ToolsUpgra
dePolicy } } |Sort Name

Name ToolsUpgradePolicy
---- ------------------
VM01 upgradeAtPowerCycle
VM02 manual
VM03 manual
VM04 manual
VM05 manual
```

Here we can see which VMs have the automatic upgrade set and which ones are configured for manual. Utilizing a filter we can look for objects that are set for manual and then configure them to be set for upgradeAtPowerCycle


```
PS C:\Users\dstamen> $ManualUpdateVMs = Get-Folder Testing|Get-VM|Get-View | Where-Object {$_.Config.Tools.ToolsUpgradePolic
y -like "manual"}|select name,@{N='ToolsUpgradePolicy';E={$_.Config.Tools.ToolsUpgradePolicy } }
PS C:\Users\dstamen>
PS C:\Users\dstamen> Foreach ($VM in ($ManualUpdateVMs)) {
>>     $VMConfig = Get-View -VIObject $VM.Name
>>     $vmConfigSpec = New-Object VMware.Vim.VirtualMachineConfigSpec
>>     $vmConfigSpec.Tools = New-Object VMware.Vim.ToolsConfigInfo
>>     $vmConfigSpec.Tools.ToolsUpgradePolicy = "UpgradeAtPowerCycle"
>>     $VMConfig.ReconfigVM($vmConfigSpec)
>> }
```

If we do a final check we can see that all VMs are now set to **upgradeAtPowerCycle**.

```
PS C:\Users\dstamen> Get-Folder Testing|Get-VM|Get-View | select name,@{N= ToolsUpgradePolicy';E={$_.Config.Tools.ToolsUpgra
dePolicy } } |Sort Name

Name ToolsUpgradePolicy
---- ------------------
VM01 upgradeAtPowerCycle
VM02 upgradeAtPowerCycle
VM03 upgradeAtPowerCycle
VM04 upgradeAtPowerCycle
VM05 upgradeAtPowerCycle
```

Upgrading VM Compatibility

When it comes to upgrading your VM Compatibility this is something that should be done with caution. Upgrading VM Compatibility aka VM Hardware is like pulling out the motherboard and replacing it with a new one, so this should only be done when features and functionality in a higher level are needed. Our current recommended level is Hardware Version 11 as it handles remediation from current security threats.

Prior to upgrading your VM Compatibility you should always make sure VMware Tools are up to date first as new drivers can be required for the new virtual hardware.

As I mentioned previously prior to upgrading your VM Compatibility I recommend taking a snapshot or a backup of the virtual machine in case a rollback is needed.

Checking VM Compatibility Version

Its quite easy to see the current version of VM Compatibility via the vSphere Client, however when checking the current levels across our entire vCenter Server we may want to automate this Below you will find a quick and easy one-liner to identify the VMs and their current VM Compatibility version.

```
PS C:\Users\dstamen> get-folder Testing|Get-VM|Select Name, Version|sort Name

Name Version
---- -------
VM01     v14
VM02     v14
VM03     v14
VM04     v14
VM05     v14
VM06     v11
VM07     v12
VM08     v11
```

As we can see above a few VMs are currently running v14 which is compatible with vSphere 6.7 only. This was needed for us to take advantage of VBS and TPM security features.

Again, we should only upgrade VM Compatibility when additional functionality is needed.

Upgrading VM Compatibility Version

We have identified a need to upgrade VM08 also to v14 to take advantage of Per-VM EVC so we can handle the automation of the VM compatibility upgrade. We can do this again quite easily with PowerCLI.

If you wish to do this via the vSphere Client my colleague Nigel Hickey covers this in his blog series here. However, we are talking automation here so lets jump into the quick script to accomplish this.


```
PS C:\Users\dstamen> $HardwareUpdateVMs = Get-Folder Testing | Get-VM VM08
PS C:\Users\dstamen>
PS C:\Users\dstamen> Foreach ($VM in ($HardwareUpdateVMs)) {
>>     $VMConfig = Get-View -VIObject $VM.Name
>>     $vmConfigSpec = New-Object VMware.Vim.VirtualMachineConfigSpec
>>     $vmConfigSpec.ScheduledHardwareUpgradeInfo = New-Object -TypeName VMware.Vim.ScheduledHardwareUpgradeInfo
>>     $vmConfigSpec.ScheduledHardwareUpgradeInfo.UpgradePolicy = "always"
>>     $vmConfigSpec.ScheduledHardwareUpgradeInfo.VersionKey = "vmx-14"
>>     $VMConfig.ReconfigVM($vmConfigSpec)
>> }
```

This will not automatically upgrade the VM Compatibility, unlike VMware Tools this can not be done with the virtual machine Powered On. The next time the VM is rebooted it will be shutdown, the compatibility will be upgraded and then be powered back on.

More information on upgrading VM Compatibility can be found here.

Conclusion

Automating your VMware Tools and VM Compatibility upgrades do not need to be hard, we have quite a few ways to help you with this and help this blog has helped educate you on some additional methods. For more information on Automating your vSphere Upgrade be sure to check out the full series here.

Article: https://blogs.vmware.com/vsphere/2018/09/automating-upgrade-of-vmware-tools-and-vmware-compatibility.html

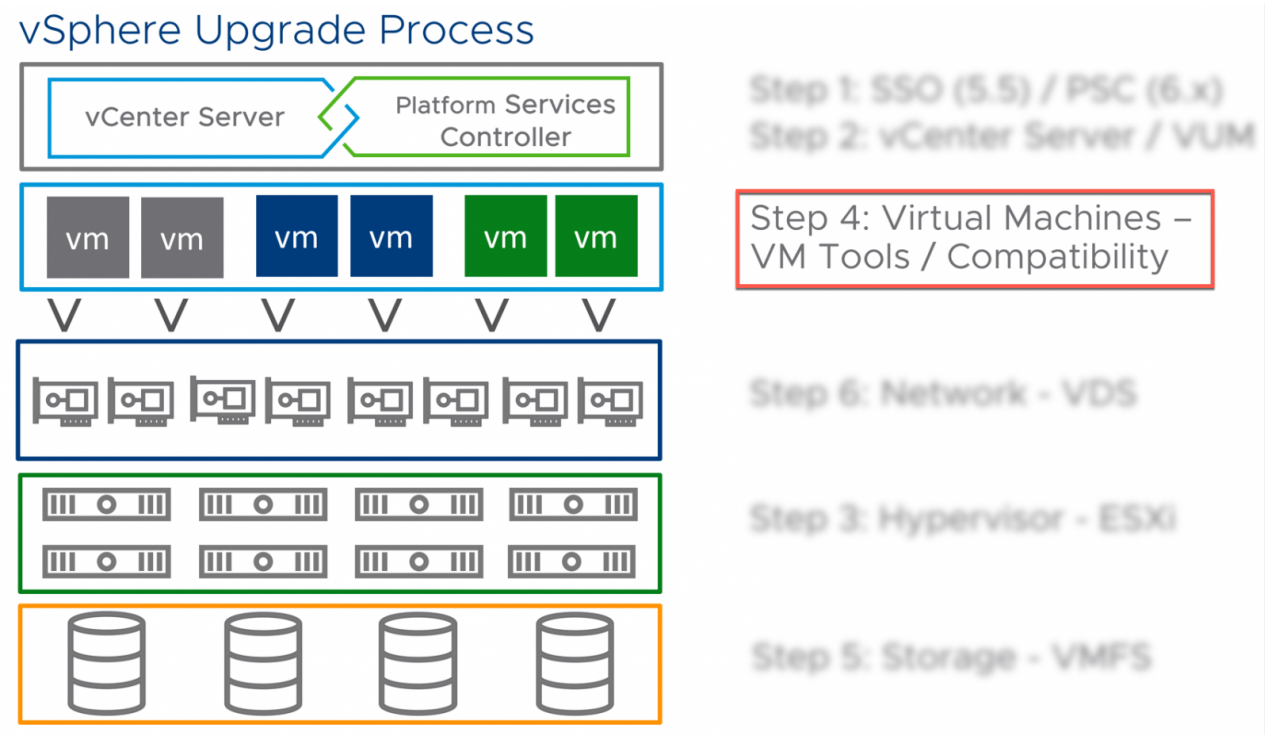
Date: 2018-10-29

Blog: vSphere Upgrade Series Part 4: Upgrading VMware Tools and VM Compatibility

vSphere Upgrade Series Part 4: Upgrading VMware Tools and VM Compatibility

Welcome back to the vSphere Upgrade Blog for the next piece of our Upgrade Journey. We began in **Part 1 of this blog series** by reviewing our prerequisites & compatibility, gathering our data. **In Part 2** we upgraded vCenter Server & migrated VUM from vSphere 6.0 Update3 to 6.7. **Part 3 guided us** through preparing vSphere Update Manager (VUM) by creating an Upgrade Baseline and using that baseline to remediate our vSphere 6.0 Update 3 ESXi hosts to vSphere 6.7.

In part 4 of the vSphere Upgrade Series we will cover Upgrading VMware Tools and VM Compatibility.



VMware Tools and Virtual Machine Compatibility comes in at Step 4 when upgrading a vSphere environment. Although both of these components hold much value for virtual machines (VM) when upgraded, caution should always be at the forefront of upgrading the VM Compatibility version. I mention caution because upgrading the VM Compatibility version may not always be necessary to perform unless specific features are needed.

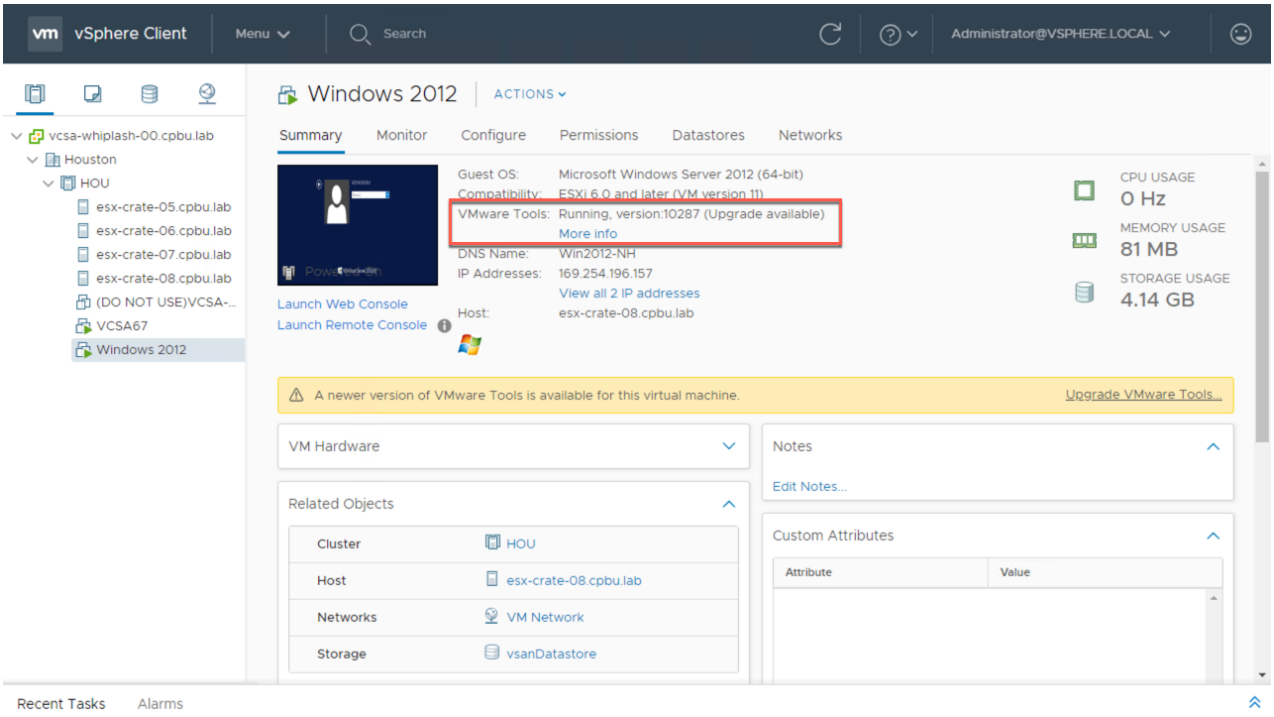
Upgrading VMware Tools

Let's start with VMware Tools. VMware Tools is a set of services and modules that enable several features in VMware products for better management of, and seamless user interactions with, guests operating systems. Although a guest operating system can run without VMware Tools, we suggest to always run the latest version of VMware Tools in your guest operating systems to access the latest features and updates.

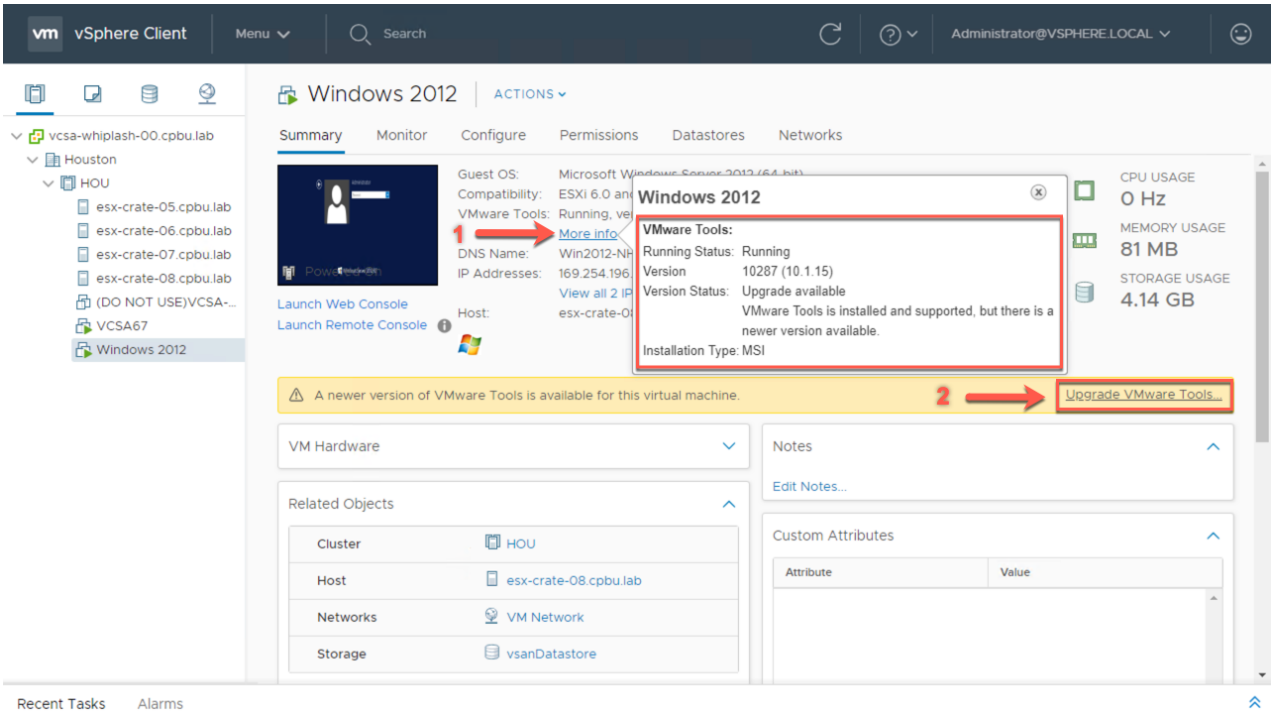
VMware Tools can be upgraded manually, via vSphere Update Manager, **PowerCLI**, or by configuring virtual machines to check and install newer versions of VMware Tools when they reboot. The guest OS checks the version of VMware Tools when you power on a Virtual Machine (VM). The status bar of the VM displays a message when a new version is available.

In my vSphere 6.7 lab environment I will be upgrading VMware Tools on a Windows 2012 server VM via the vSphere Client (HTML5) which is considered a manual upgrade. I will begin by logging into the vSphere Client and check the current version of VMware Tools running on my VM.

Step1: Click on the VM to be upgraded and then from the **Summary** tab review the version of VMware Tools.

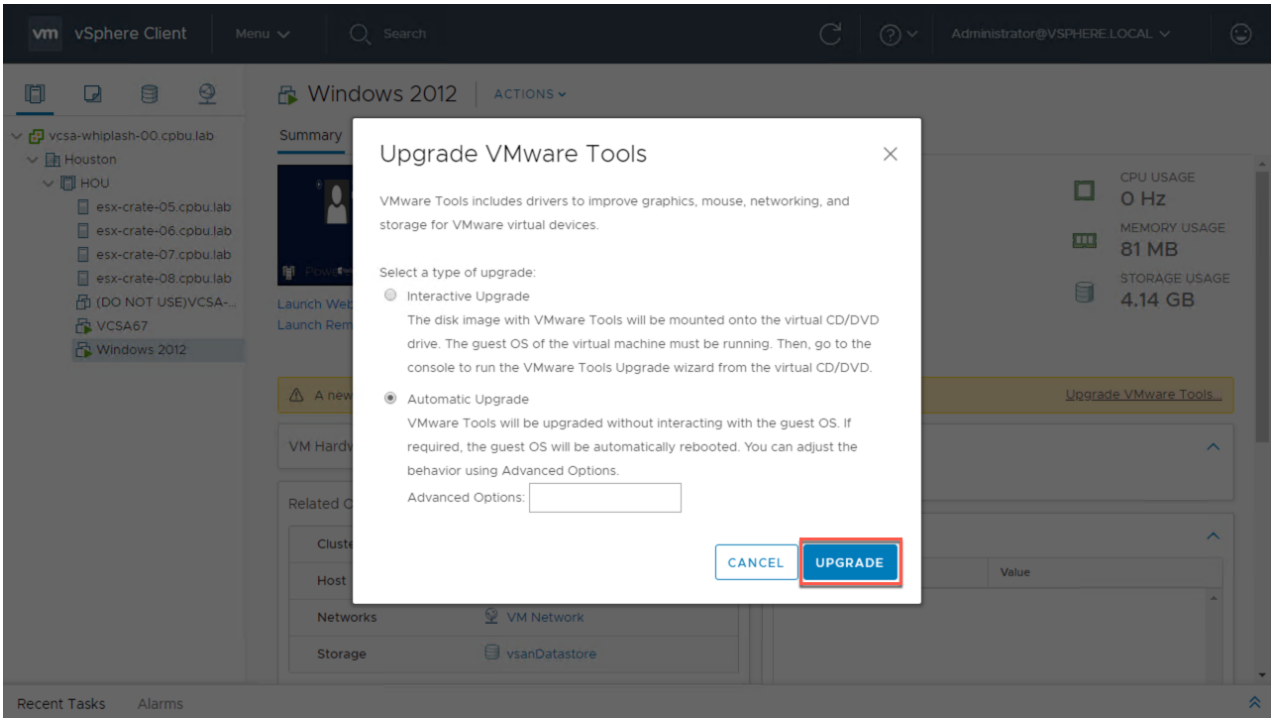


Step 2: Click **More info** to reveal additional information (optional), then click **Upgrade VMware Tools** to begin the process

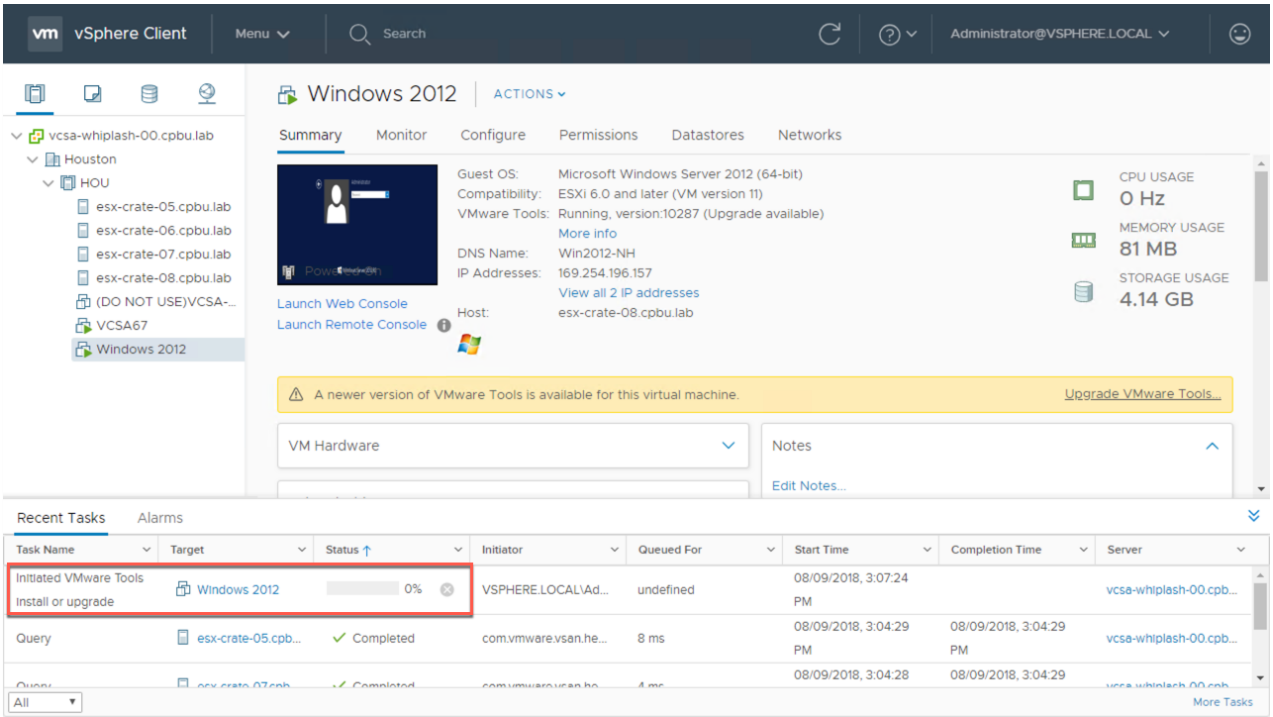


Step 3: Once you click **Upgrade VMware Tools**, you will be presented with 2 types of upgrades. Choose an upgrade type, click **Upgrade** to continue. (In my scenario, I have selected an Automatic Upgrade)

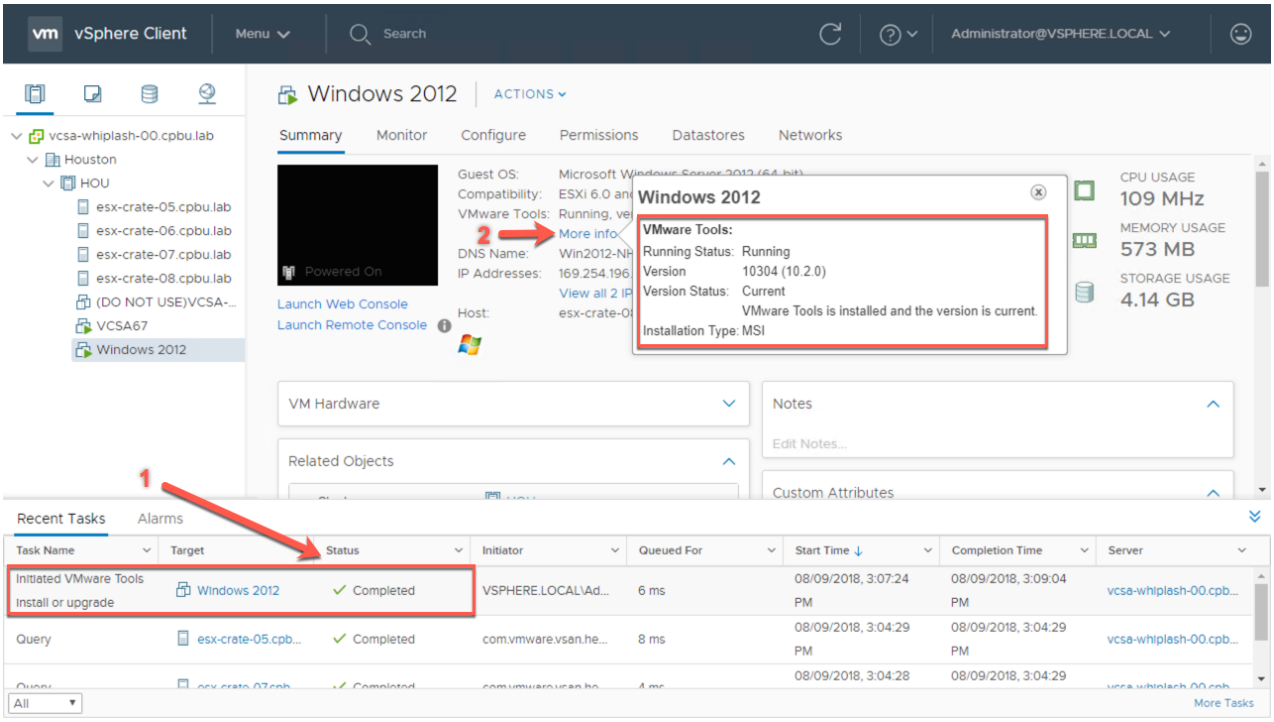
- An ***Interactive Upgrade*** allows the disk image for VMware Tools to be mounted to the guest OS. This allows the admin doing the work to console into the VM to run the install wizard.
- An ***Automatic Upgrade*** does what it implies by automatically upgrading VMware Tools without interacting with the guest OS. It also allows the guest OS to reboot automatically if needed.



Step 4: Review the ***Recent Tasks*** pane to monitor the VMware Tools upgrade progress



Step 5: Once the upgrade is complete, we verify the upgrade by clicking More info to display VMware Tools' version & status.



Upgrading VMware Tools with PowerCLI

VMware Tools updates can also be performed via PowerCLI. The cmdlet "***Update-Tools***" can be used to upgrade the VMware Tools on the specified virtual machine guest OS. VMware Tools must be installed prior to updating it. After VMware Tools is updated, the virtual machine is restarted unless the NoReboot parameter is specified.

An example usage is shown below that can update VMware Tools on a virtual machine specified by its guest operating system.

Example scripts for VMware Tools Management can also be found on our VMware GitHub pages.

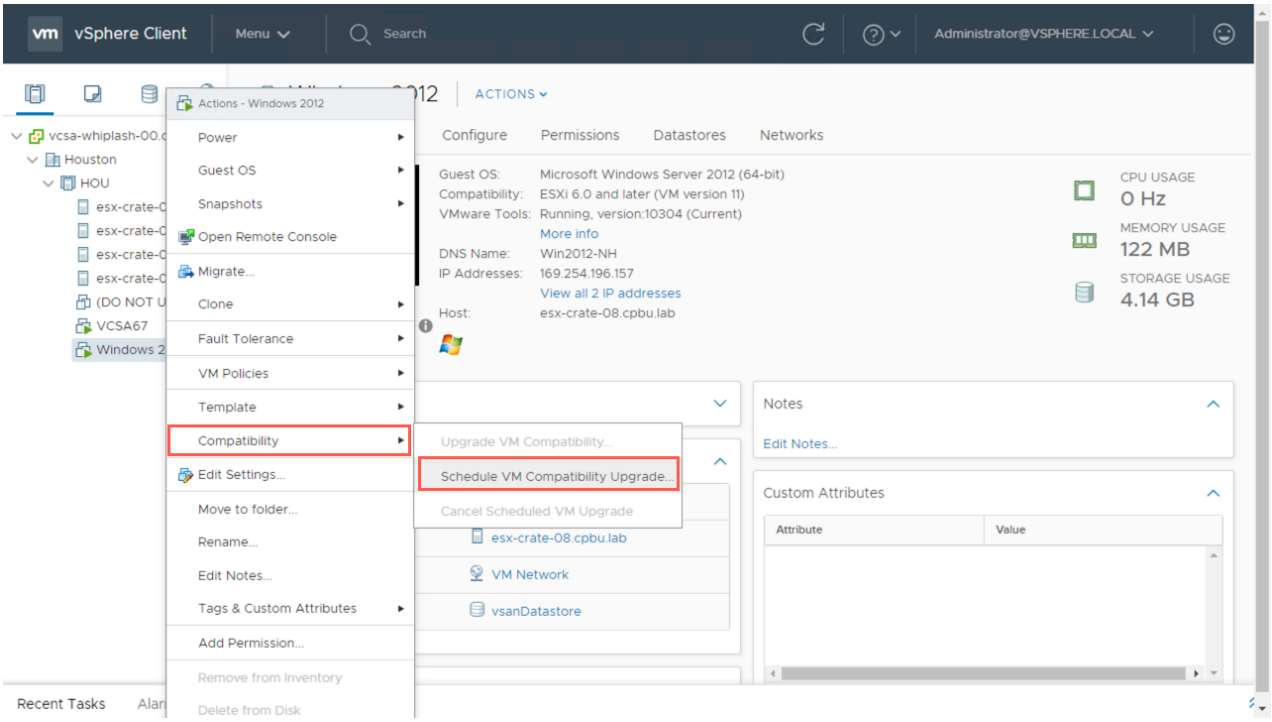
Upgrading Virtual Machine Compatibility

Now we will review Virtual Machine Compatibility. Virtual machine compatibility determines the virtual hardware available to the VM, which corresponds to the physical hardware available on the vSphere host. Upgrading the compatibility level will allow the VM to take advantage of additional hardware features available to the virtual machine.

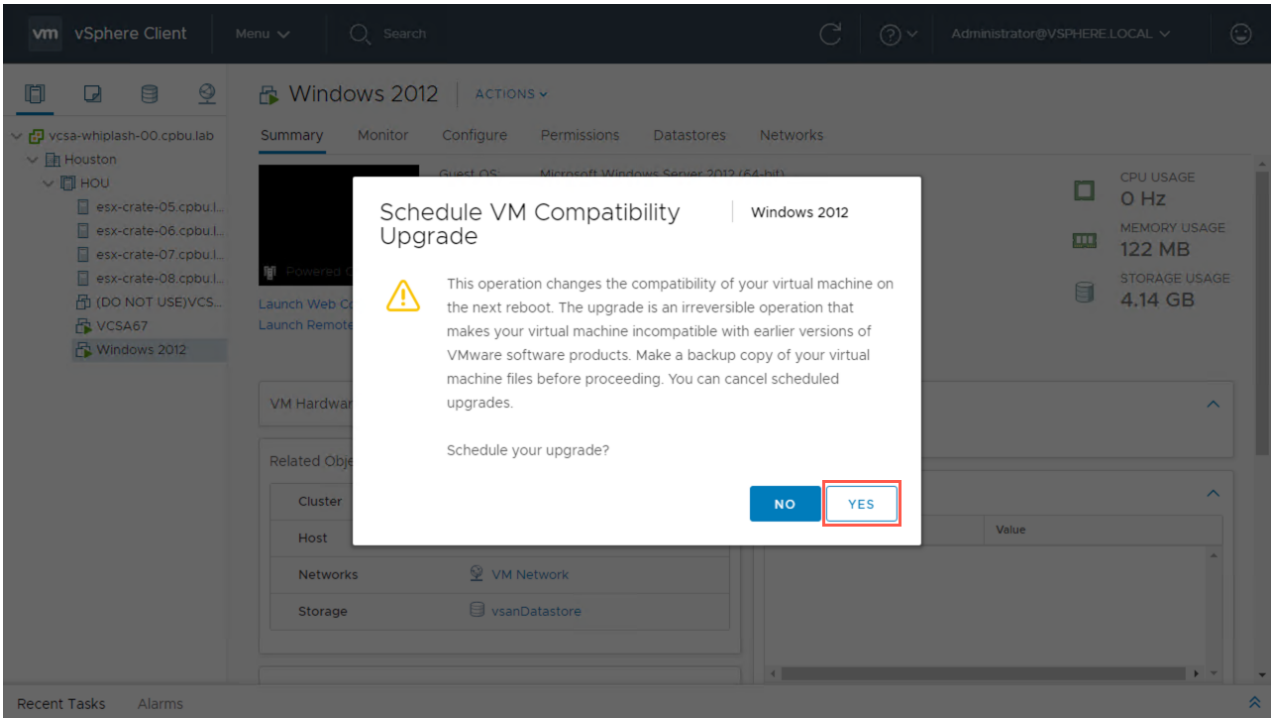
In vSphere 6.7, Virtual Machine Compatibility (Virtual Hardware Version) 14 was introduced. VM Compatibility version 14 includes support for features such as; Per-VM EVC, Virtual TPM 2.0, and Microsoft Virtualization Based Security (VBS). Note that when upgrading VM Compatibility some applications or the OS to may have issues working properly. I suggest only upgrading VM Compatibility if you require a feature that comes with the newer hardware version.

In my lab example I will upgrade VM Compatibility from version 11 (ESXi 6.0 and later) to version 14 (ESXi 6.7 and later) via vSphere Client (HTML5).

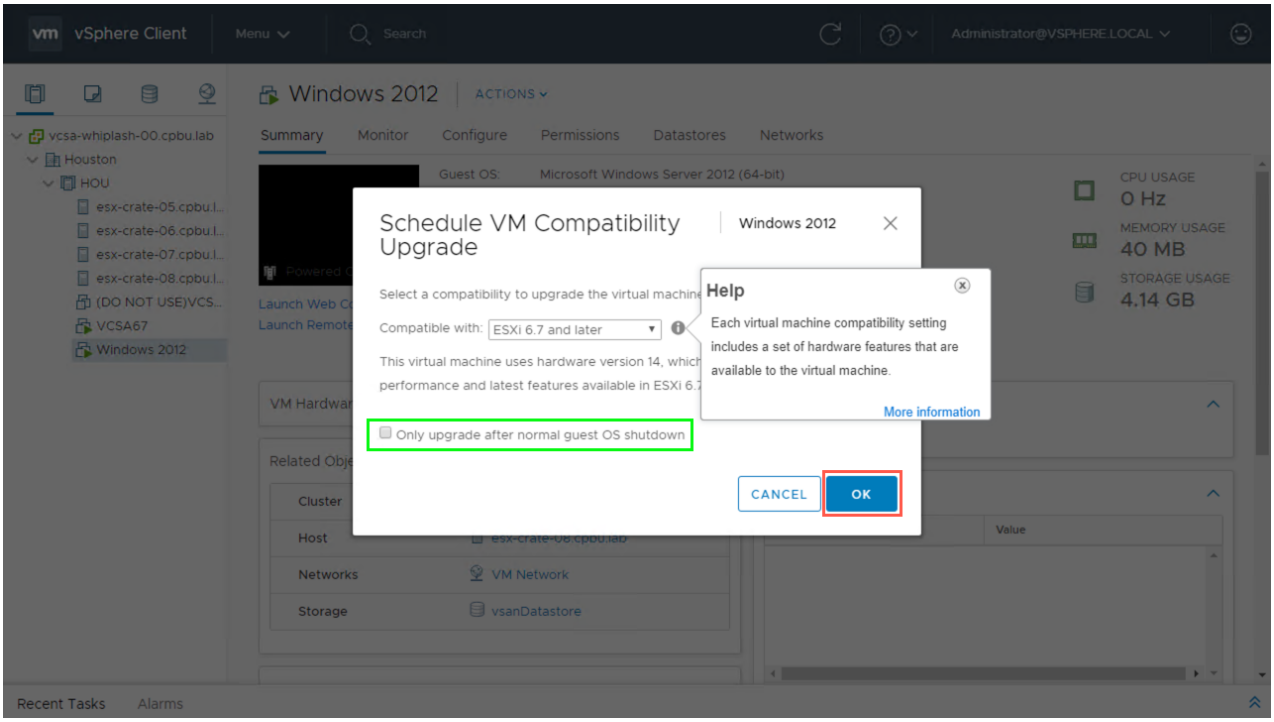
Step 1: Right click on the VM to be upgraded (or use the ***Actions*** menu once the VM is highlighted), choose **Compatibility** and then click **Schedule VM Compatibility Upgrade**.



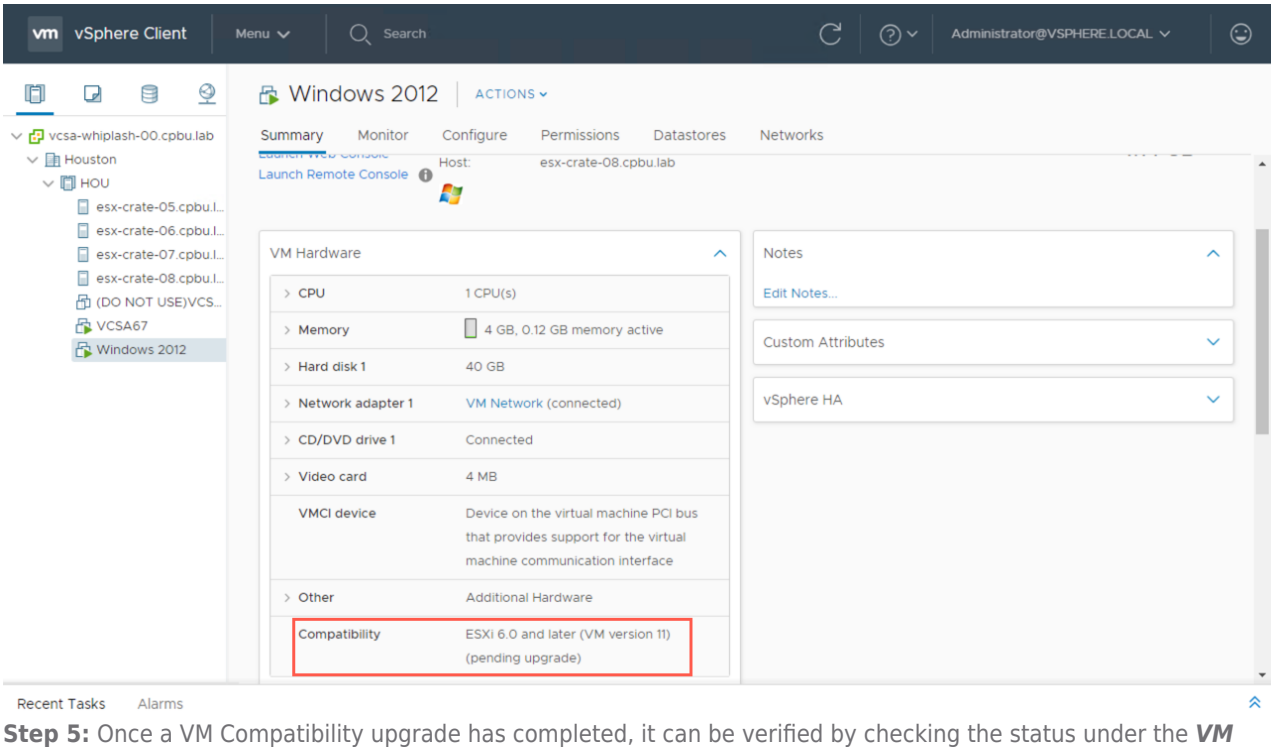
Step 2: Read the next window to understand what a VM Compatibility Upgrade will do. When ready, click **Yes** to continue.



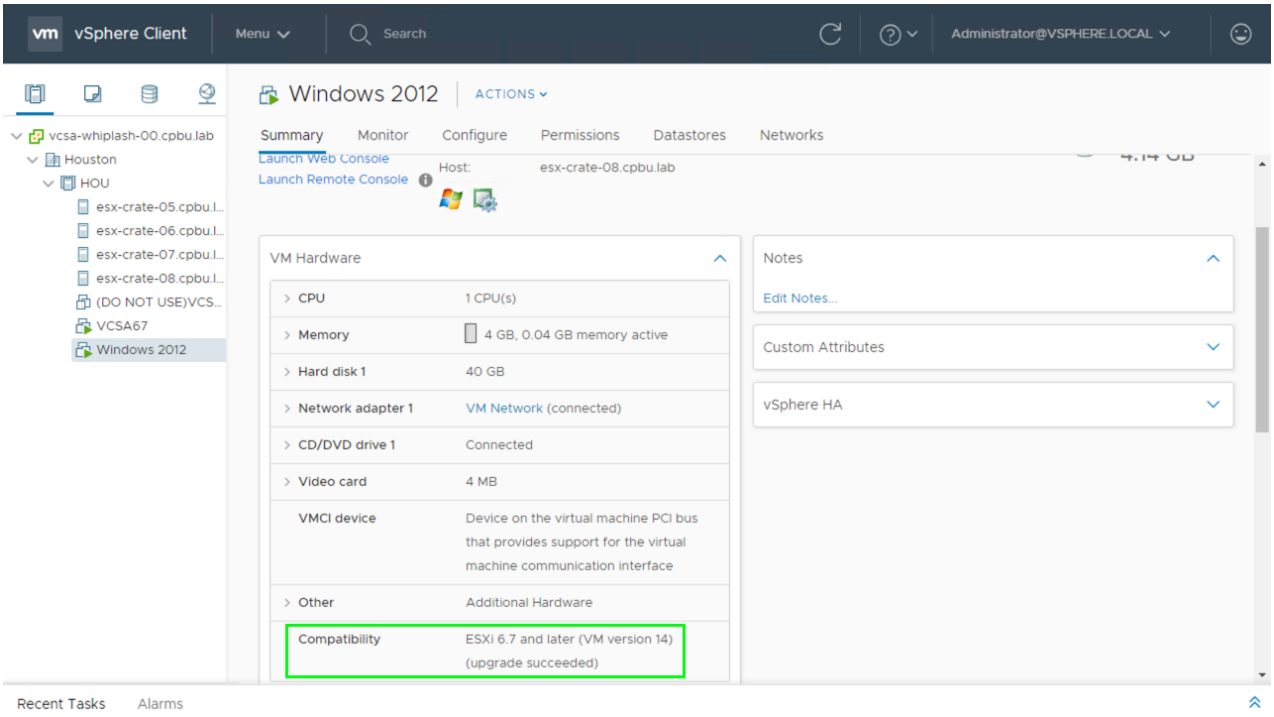
Step 3: Choose the VM Compatibility version desired (example: *ESXi 6.5 and later* or *ESXi 6.7 and later*). You may also choose to only upgrade after a normal OS shutdown versus when an OS crashes then reboots.



Step 4: Next we can review the status of the upgrade. When VM Compatibility is scheduled to be upgraded, you will notice the status of the upgrade is viewable under the ***VM Hardware*** section of the virtual machine.



Step 5: Once a VM Compatibility upgrade has completed, it can be verified by checking the status under the ***VM Hardware*** section of the virtual machine.



Conclusion

As you have witnessed updating VMware Tools or Virtual Machine Compatibility are not complex tasks, but absolutely include steps that will require consideration prior to execution. It is always best to consult VMware Documentation pages for further details on VMware Tools as well as Virtual Machine Compatibility prior to upgrading. Also be sure to review the vSphere Upgrade Guide.

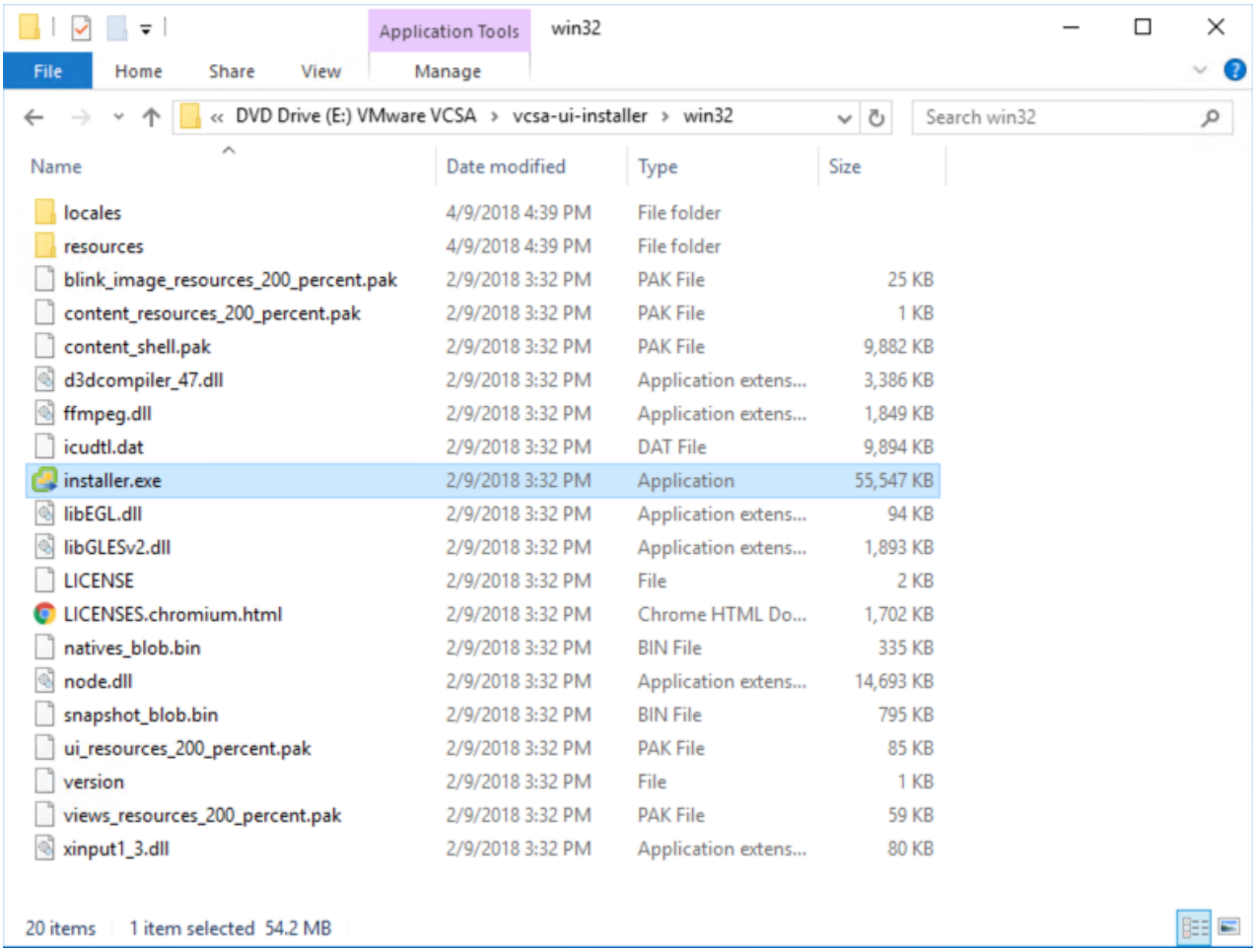
In the next **vSphere Upgrade Series** post, we will focus on upgrading **Upgrading VMFS Storage** in a vSphere 6.7 environment. Please do not hesitate to post questions in the comments section of this blog or reach out to me directly via Twitter @vCenterNerd.

In part 1 of the vSphere Upgrade Series, **Preparing to Upgrade**, we covered getting started with our prerequisites, compatibility, and also prepared the vSphere Update Manager (VUM) server to migrate its data to the VCSA 6.7 during the upgrade. In part 2 we will cover the vCenter Server Upgrade to 6.7. Let's begin.

vCenter Server Upgrade

Now that VUM has passed its Migration Assistant pre-checks, we can move to the vCenter Server Upgrade. I am also assuming here that you have a backup of the vCenter Server prior to upgrading.

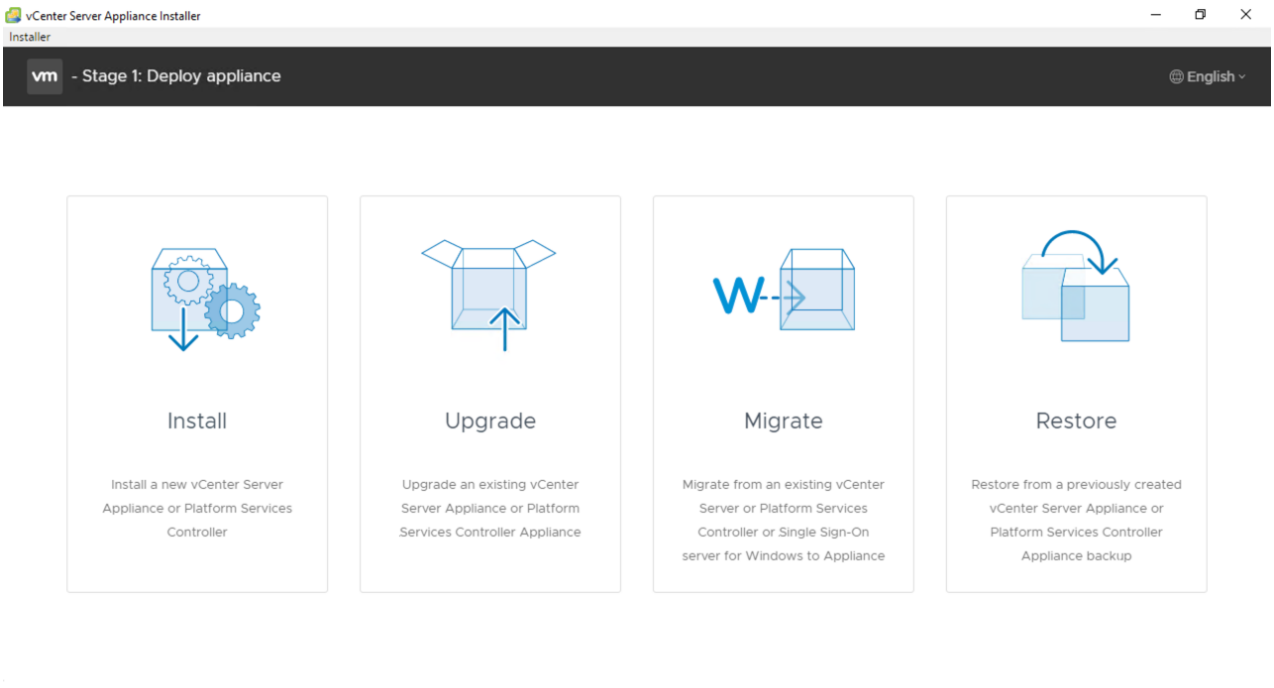
We begin by mounting the VCSA installation ISO to an Admin workstation that is on a routable network to the vCenter Server we will be upgrading. Browse the ISO and open the "**vcsa-ui-installer**" folder then the corresponding folder for your OS. I am running this from a Windows system so I will open the "**win32**" folder.



Next run the application "**installer.exe**" with Administrator Rights. This will launch the vCenter Server Appliance Installer to begin Stage 1.

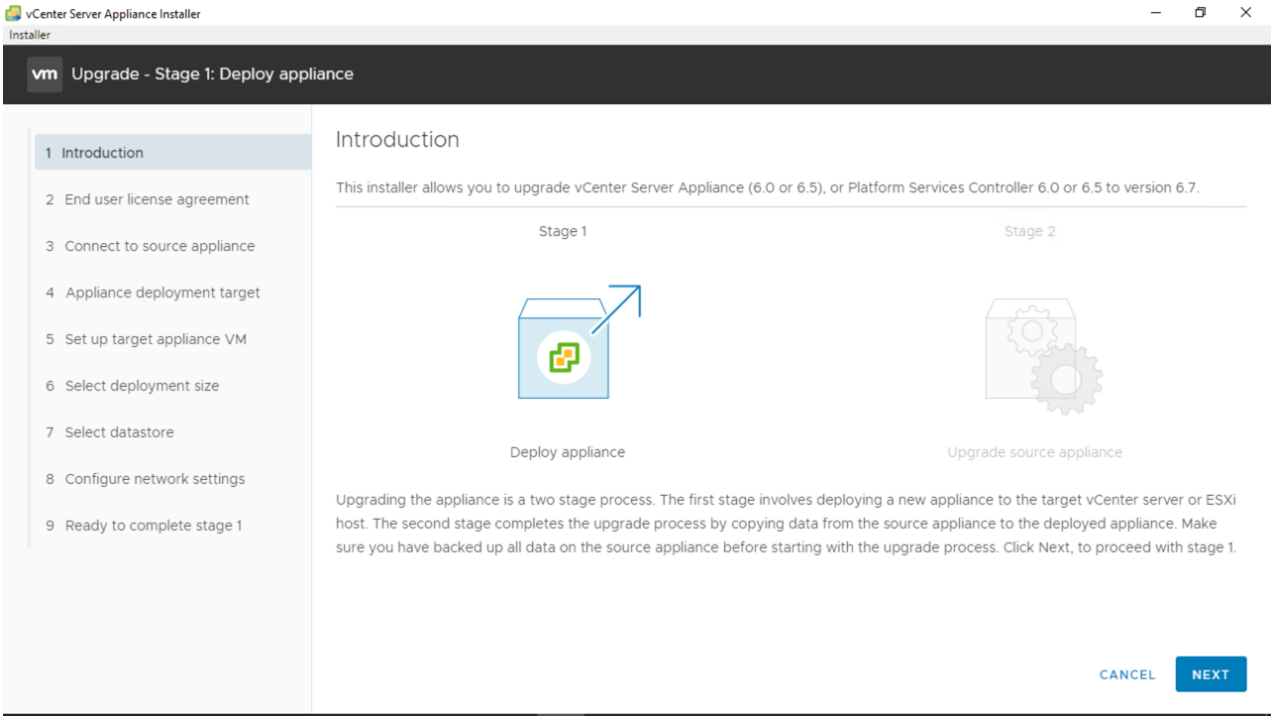
NOTE: The upgrade of the vCenter Server is broken up into two stages, Stage 1 & Stage 2. Stage 1 is the deployment of a new VCSA and Stage 2 is where all of the configuration data and inventory are imported into the newly upgraded vCenter Server.

Begin by clicking **Upgrade**.

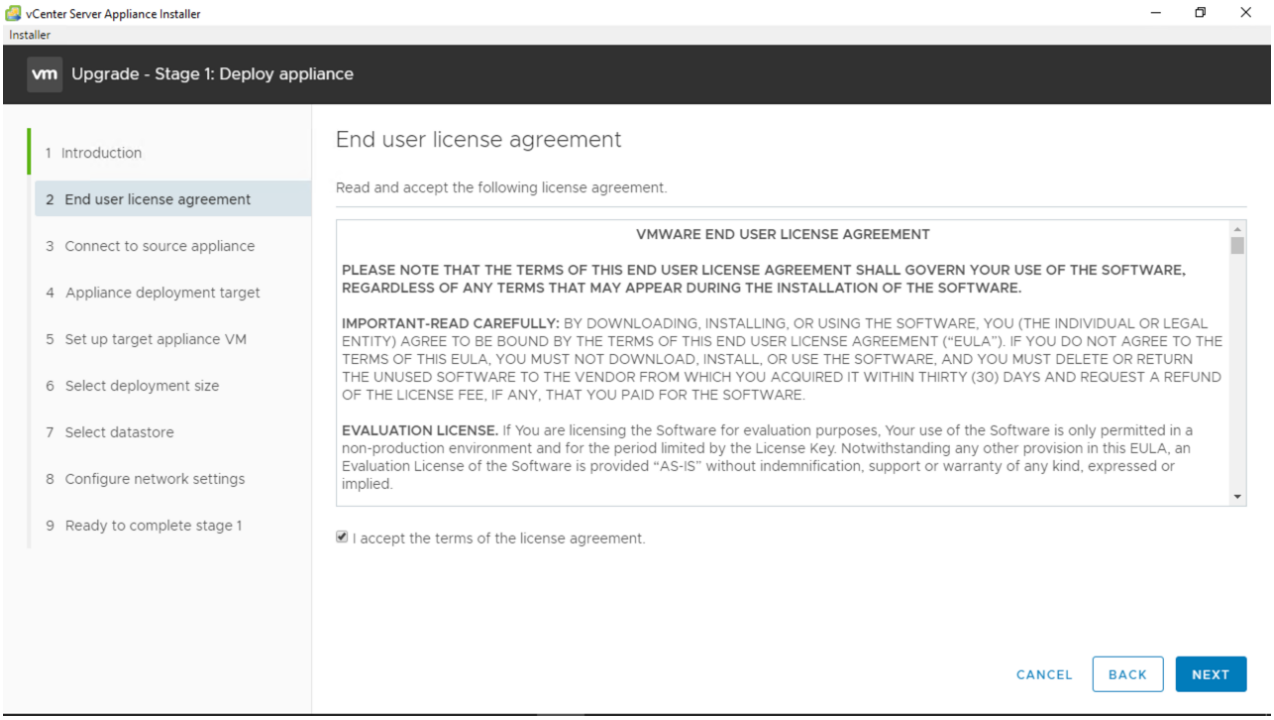


Stage 1

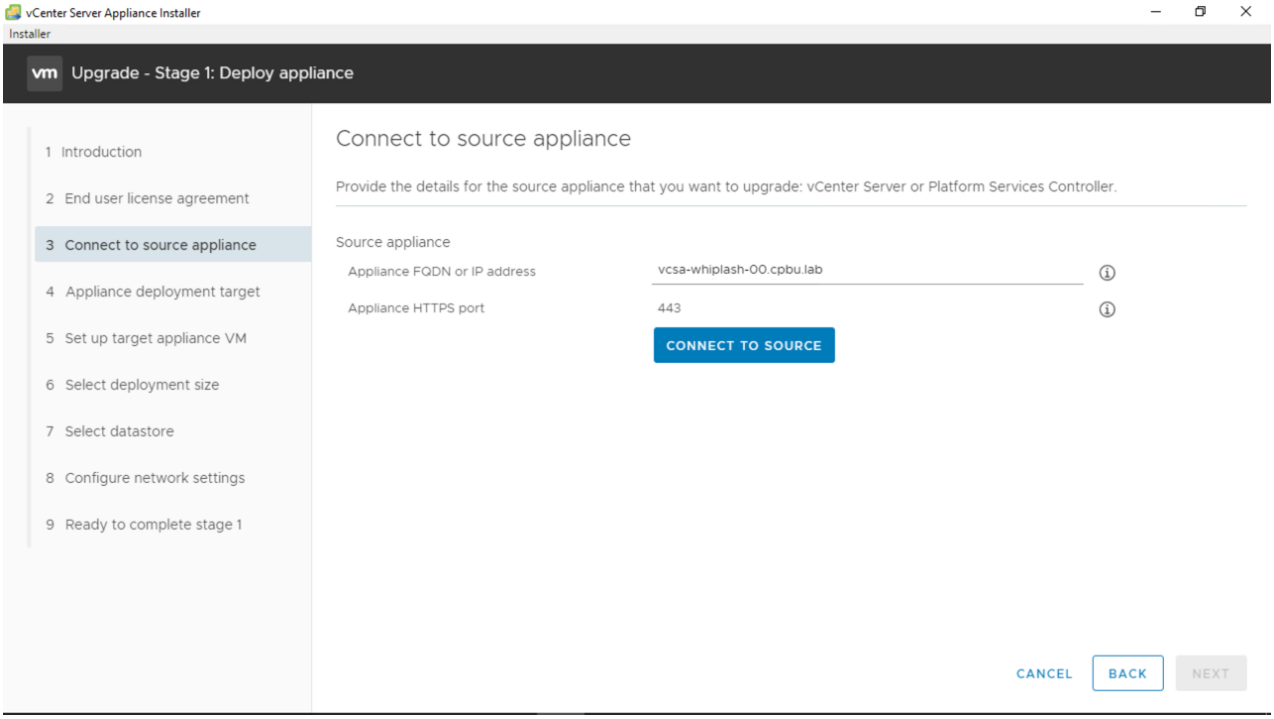
Step 1: This is the introduction and an explanation of the two Stages of the Upgrade. Click **Next** to continue.



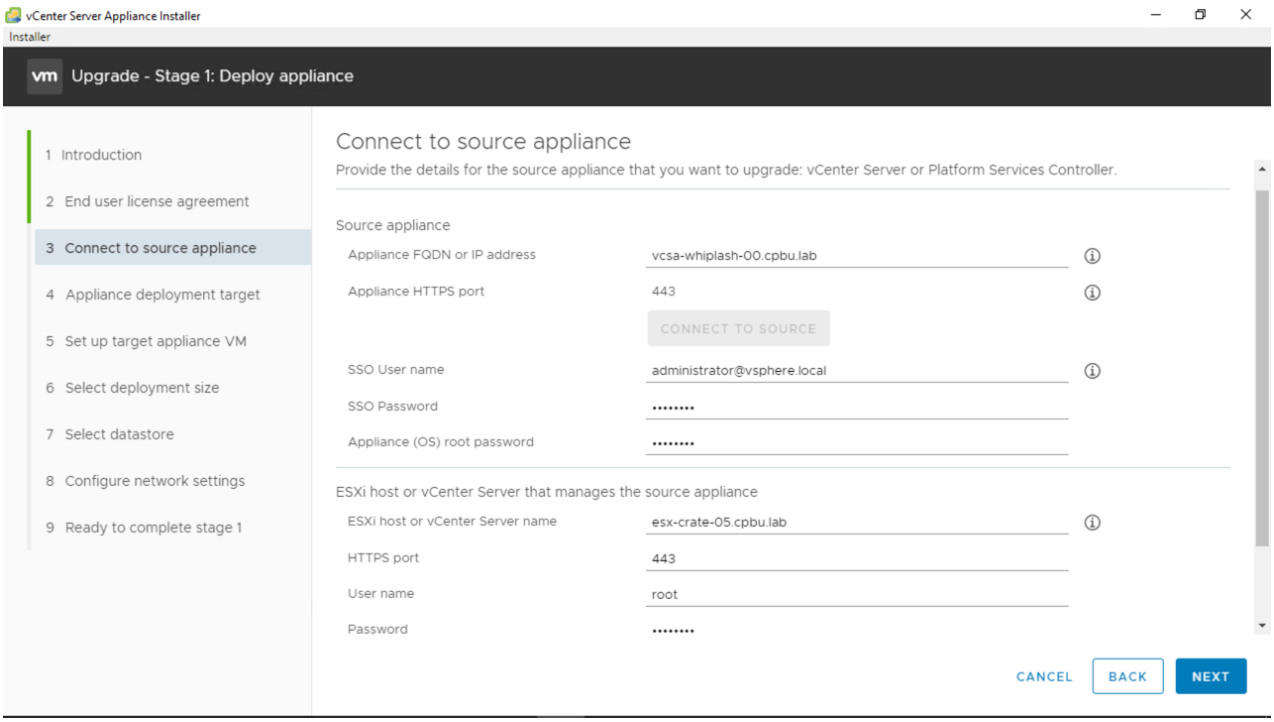
Step 2: Review the EULA, check the box to accept the terms of the license agreement, and then click **Next** to continue.



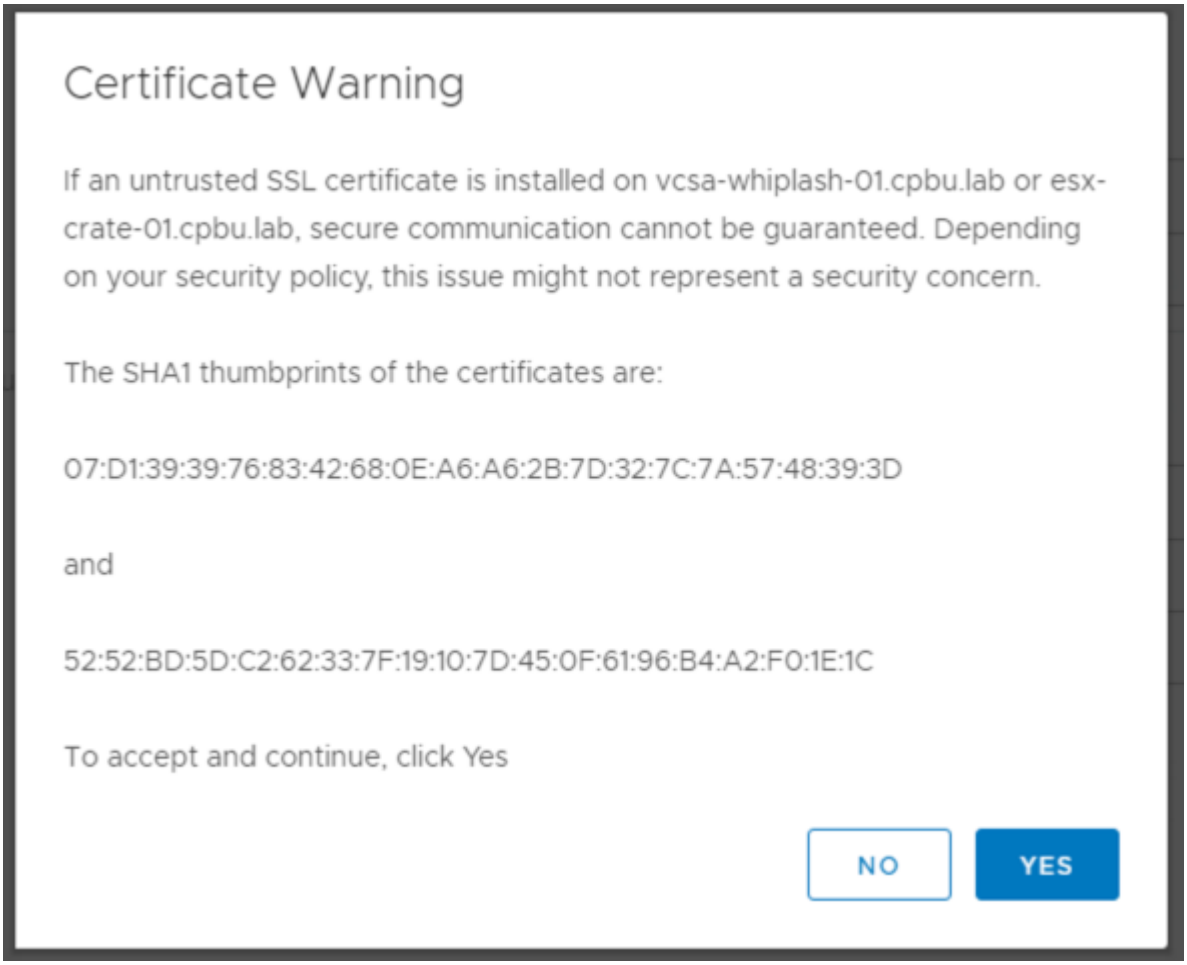
Step 3a: Enter the source vCenter Server that you will be Upgrading by its ***FQDN*** or ***IP address***. Click **Connect To Source** to reveal the additional fields.



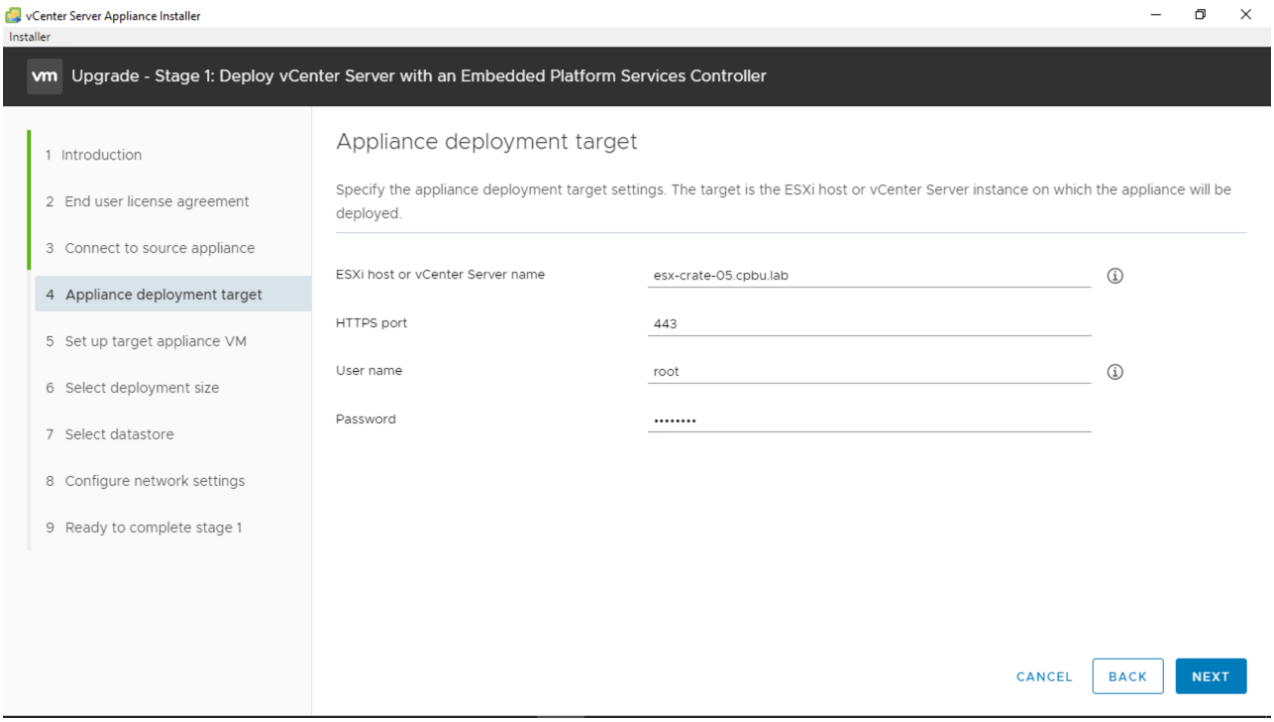
Step 3b: Complete each required field for SSO as well as the information about the ESXi host that manages the source vCenter Server. Then click **Next** to continue.



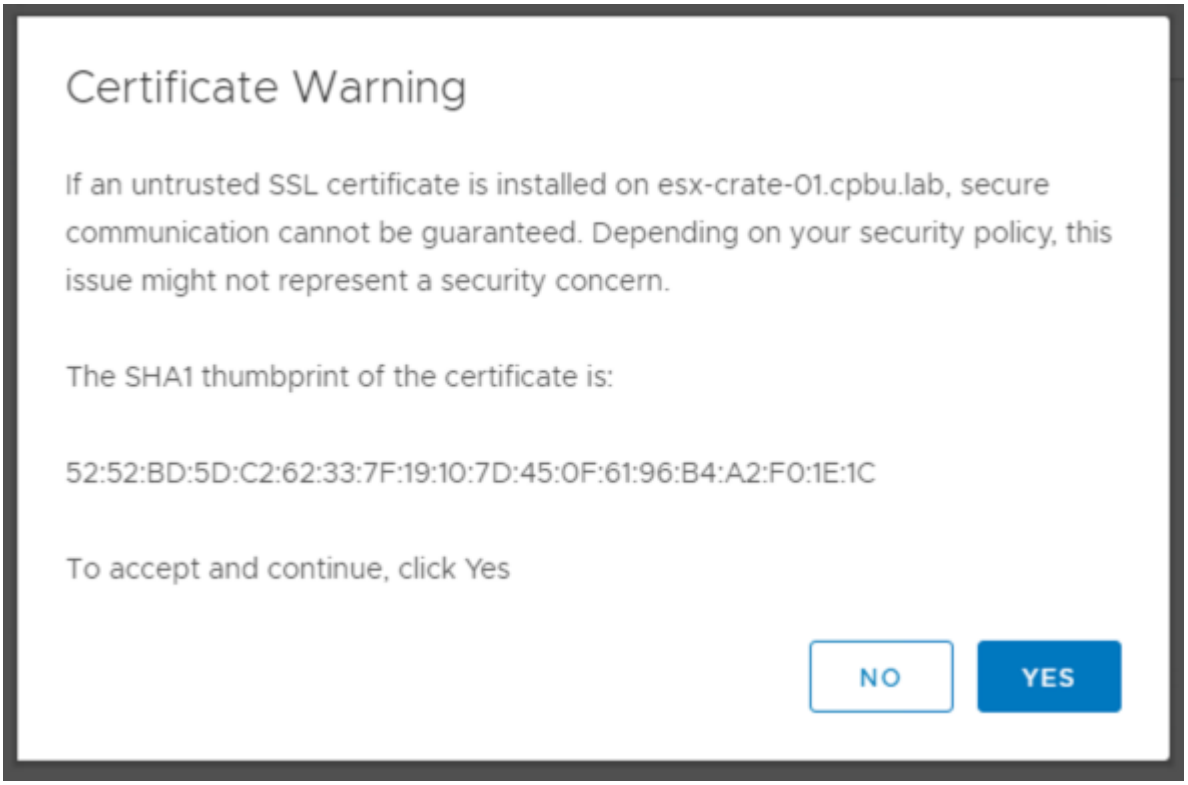
Step 3c: You will prompted to verify the SSL Certificates. Review and click **Yes** to continue.



Step 4a: Specify the target host or vCenter Server to which the new VCSA will be deployed to. Click **Next** to continue.

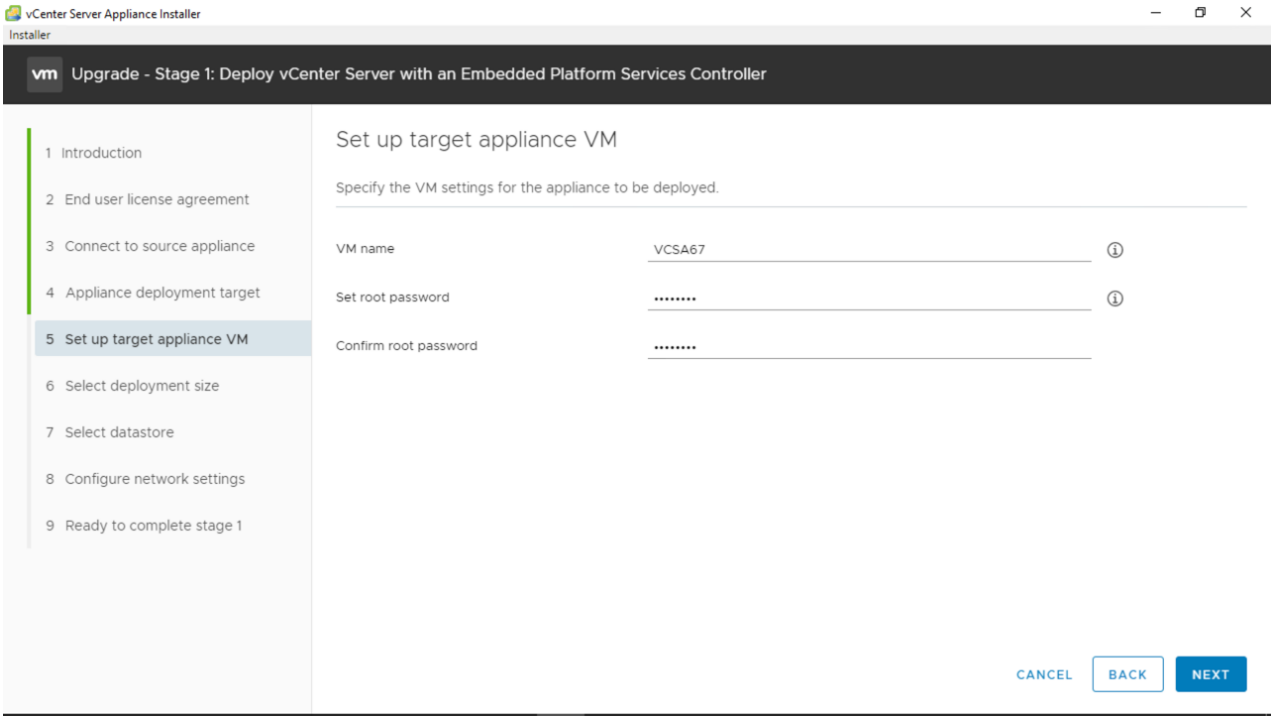


Step 4b: When prompted, review the SSL Certificate and thumbprints then click **Yes** to continue.



Step 5: Specify the Virtual Machine name (this is only the Inventory name) and set the password for the VCSA that will be deployed during the upgrade. Click **Next** to continue.

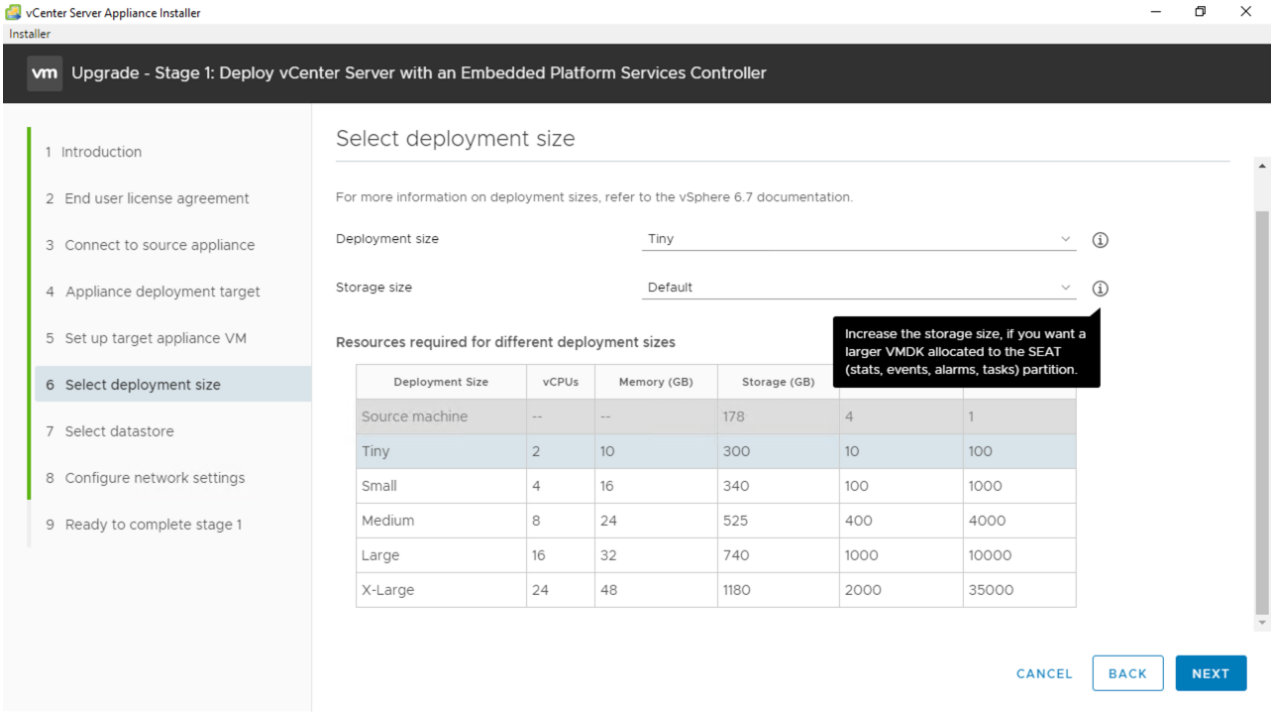
NOTE: During an upgrade of the source VCSA, a new VCSA virtual machine is deployed and configurations are imported to the new vCenter Server Appliance. The source VCSA will be powered off and should be either removed from inventory, or have its network adapter DEACTIVATED after the upgrade completes.



Step 6: Select the deployment size for the vCenter Server. If more storage is needed than the default sizing, choose the "**Storage Size**" dropdown for more choices. Storage size changes will be reflected in the table below the selection dropdown in the **Storage (GB)** column.

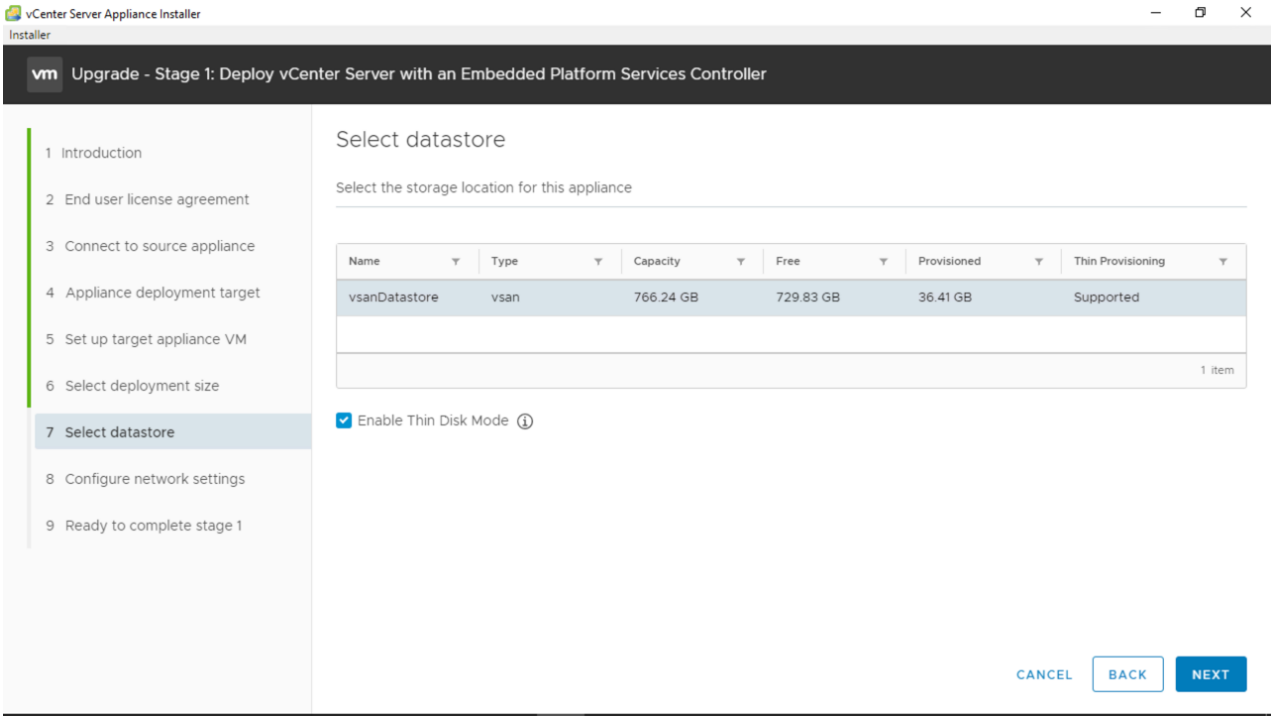
It may be necessary to edit the storage size from **Default** to *Large* or *X-Large* if importing the optional Historical & Performance Data (see image for more details).

NOTE: If you plan to use vCenter High Availability (VCHA) after you upgrade your vCenter Server, the smallest deployment size supported for VCHA is Small.
Click **Next** to continue.

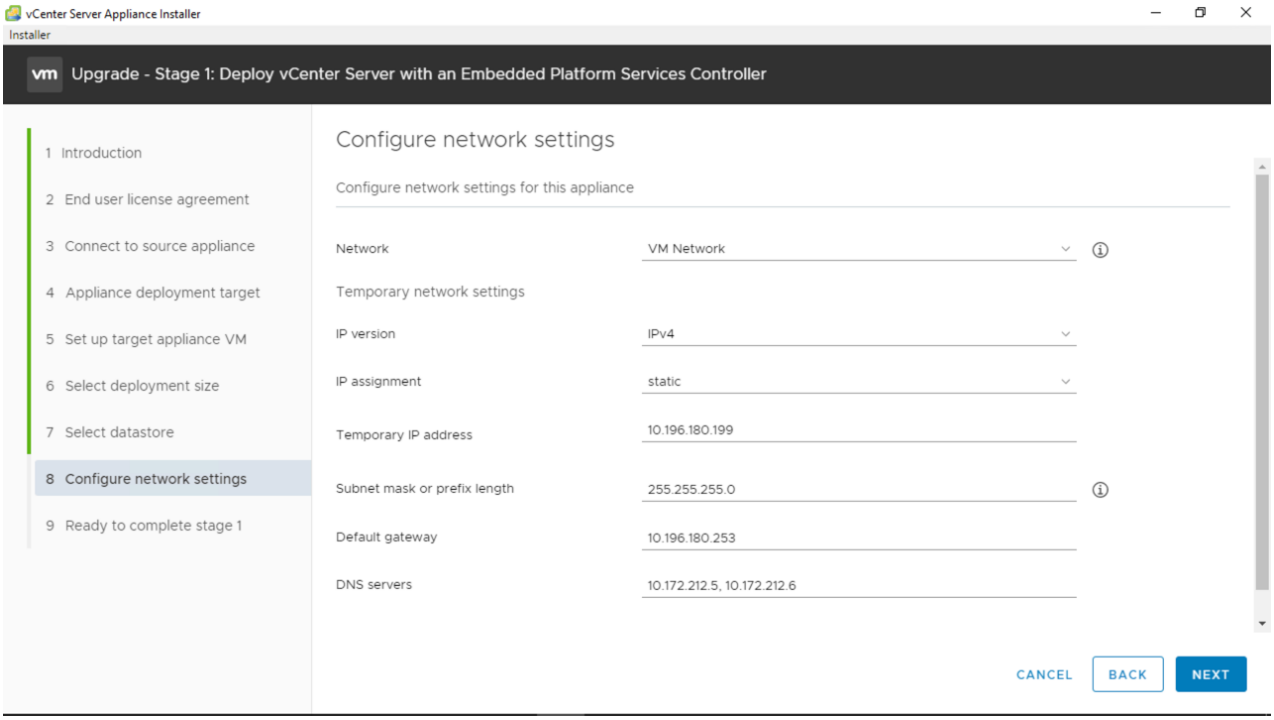


Step 7: Select the datastore location for the vCenter Server Appliance. The option to also Enable Thin Disk Mode is available, if you require it.

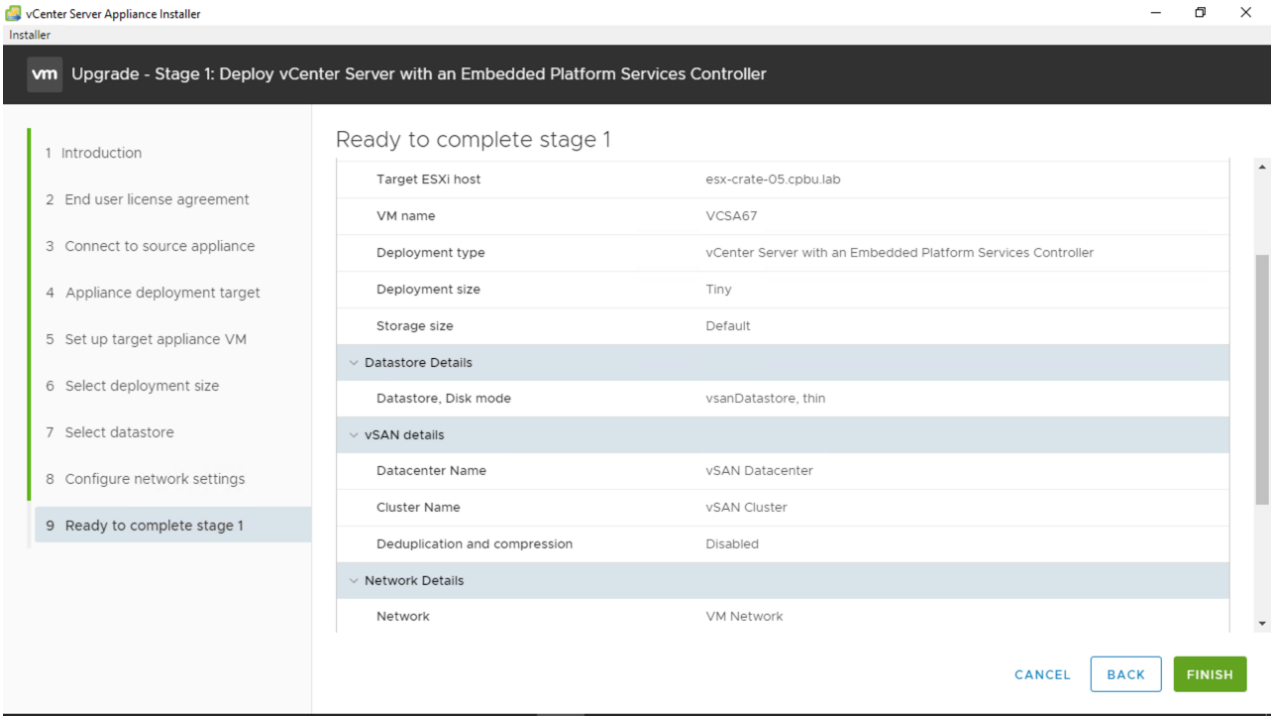
Click **Next** to continue.



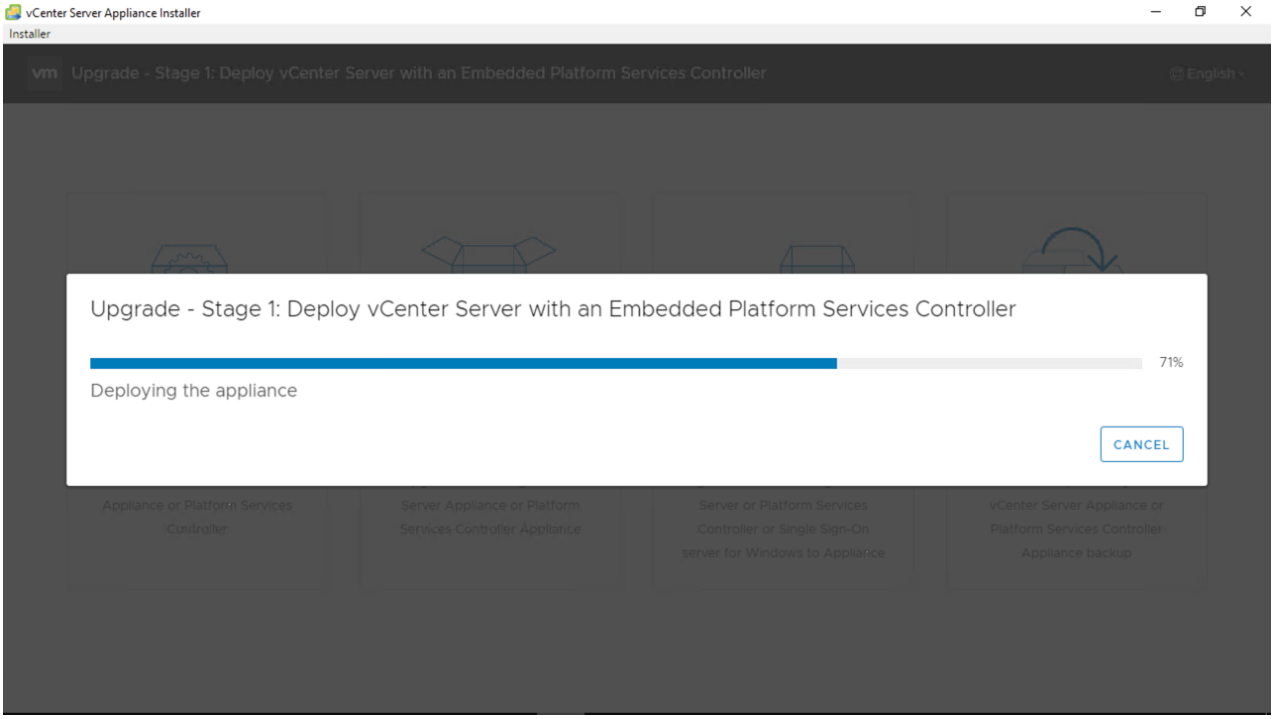
Step 8: Configure the temporary network settings that are required to deploy the new VCSA. Be sure to add at least 1 DNS server. Once all required fields are completed, click **Next** to continue.



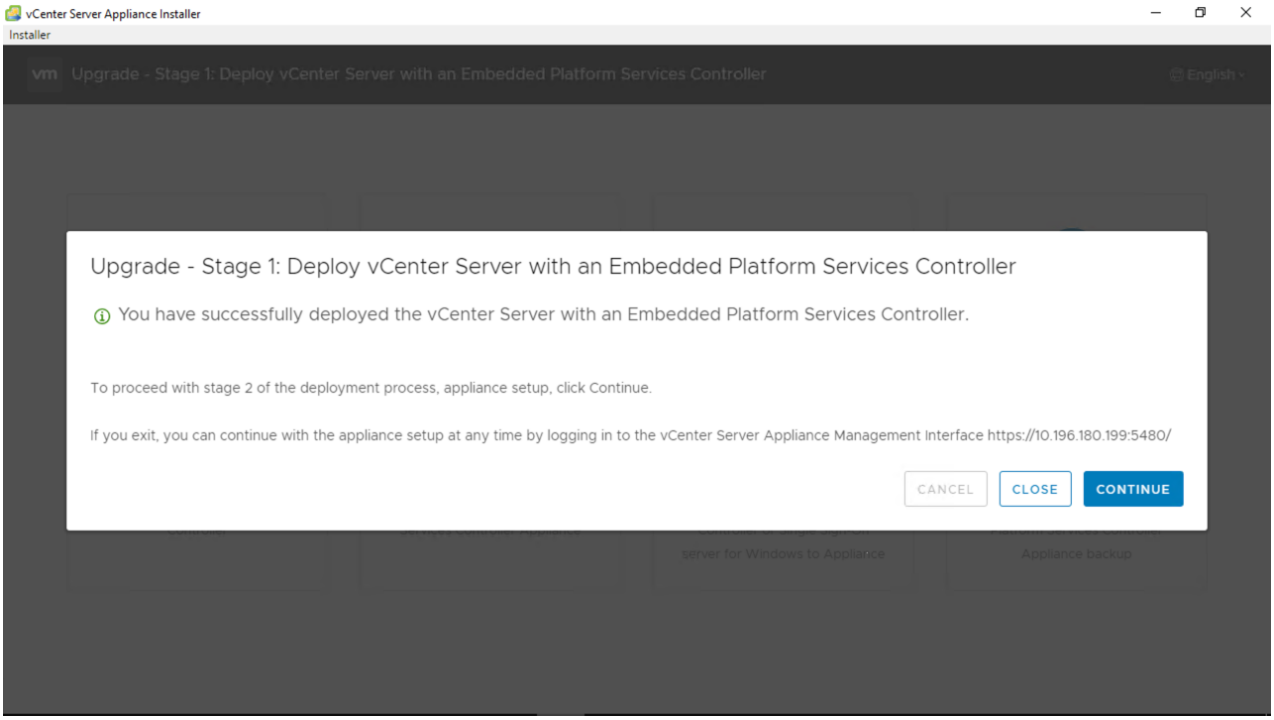
Step 9: Review all settings of Stage 1 prior to Upgrade. Once verified, click **Finish** to continue and kick off Stage 1 of the vCenter Server Upgrade.



Stage 1 of deploying the new vCenter Server Appliance is now underway.



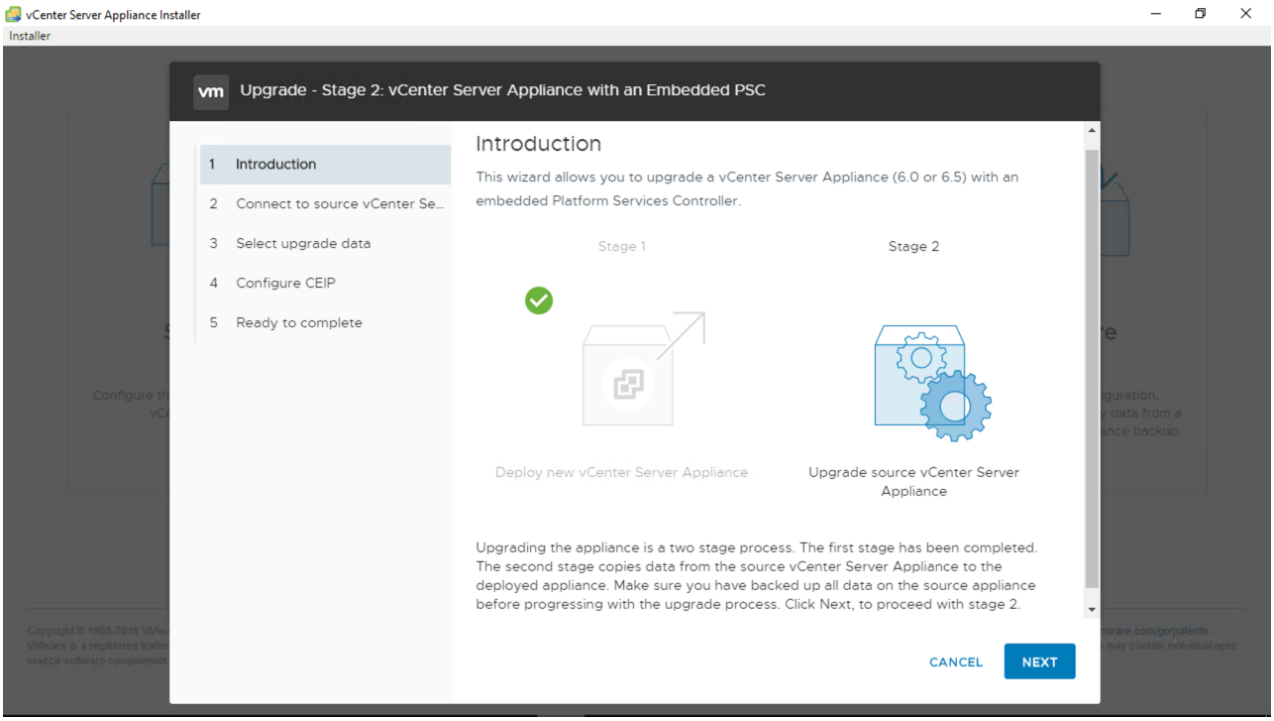
Once the VCSA is deployed and all RPM installs are completed in Stage 1, you can click **Continue** to move on to Stage 2.



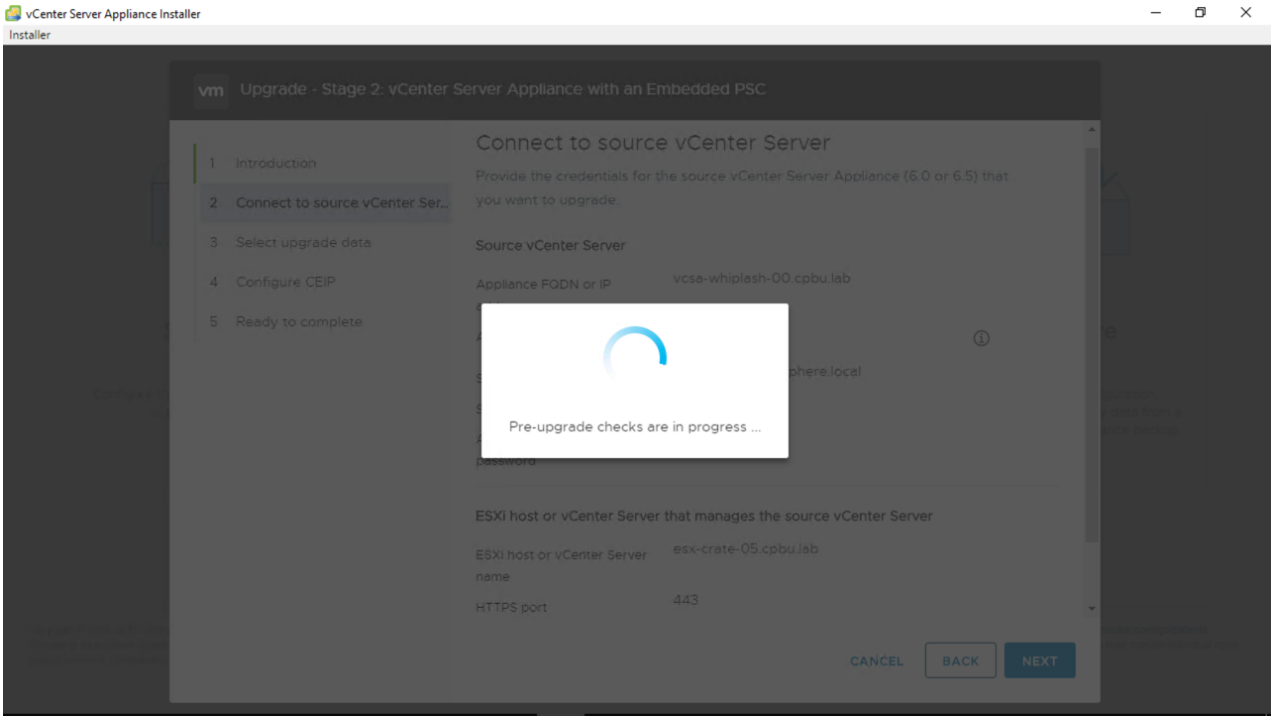
Stage 2

Stage 2 of the upgrade is when the data from the source vCenter Server as well as vSphere Update Manager (VUM) is imported into the newly deployed VCSA.

Step 1: Review the details of the Stage 2 process and then Click **Next** to continue.

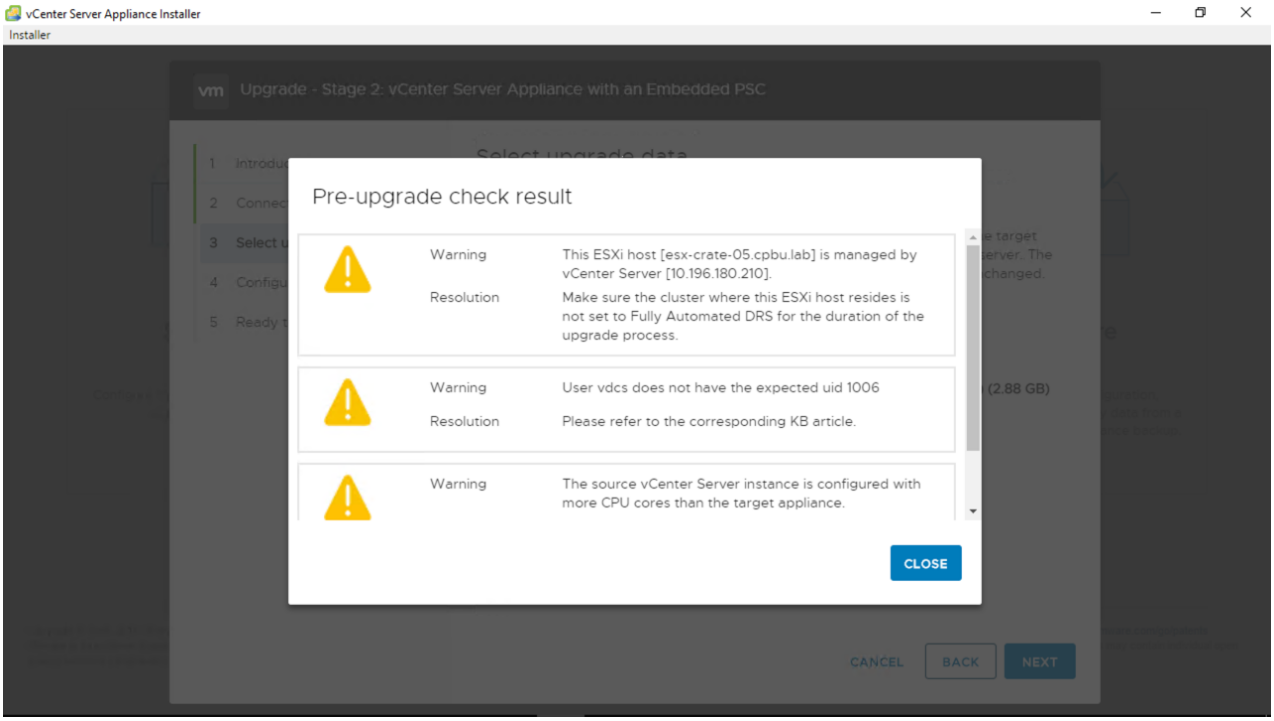


This will kick off a series of pre-checks on the source vCenter Server.

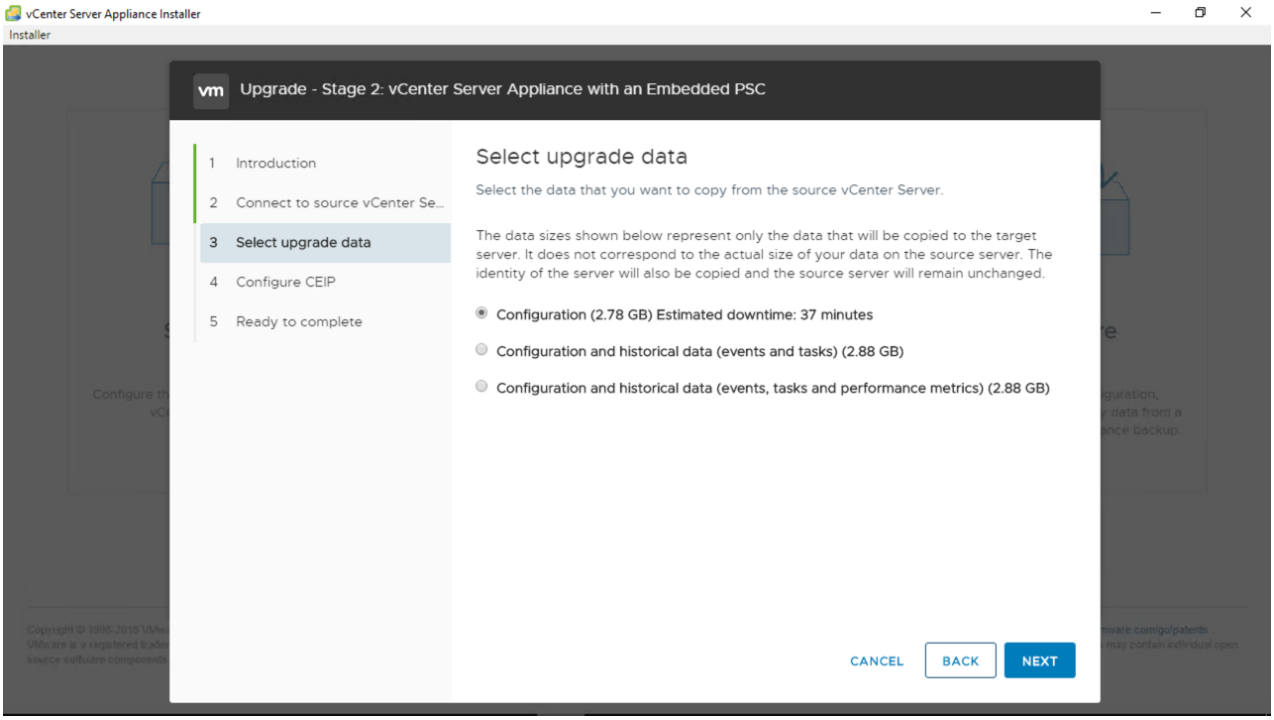


Step 2: Once pre-checks have completed, results will be shown on screen. Review any warnings given, as well as the resolutions to these warnings. In my upgrade scenario I had a few warnings, one reminding me to change DRS which we did before we began, and others that I validated were ok to proceed in this situation.

Review and click **Close** to continue.



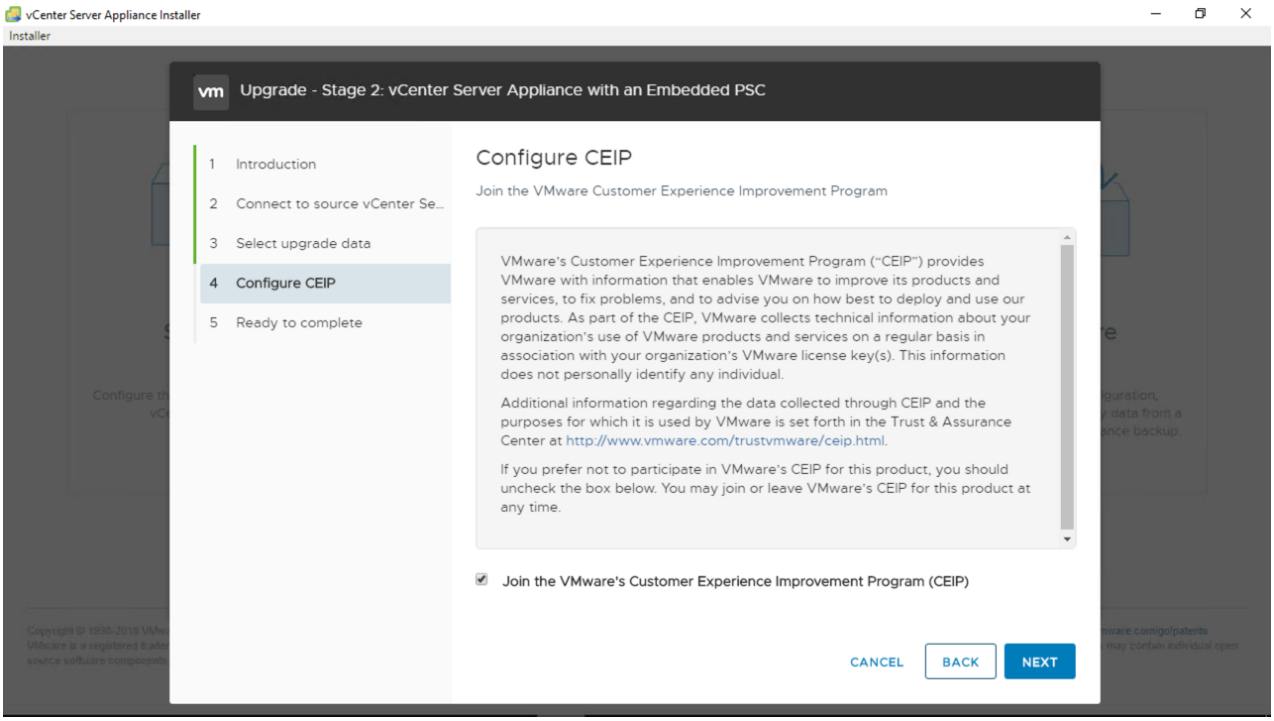
Step 3: Select the data that you will require to be imported. The Inventory & Configuration data is moved by default, any historical data (events, tasks, performance, etc) is optional to import. This is offered to shorten the upgrade and migration of data into the new VCSA. Make the required choice, and then click **Next** to continue.



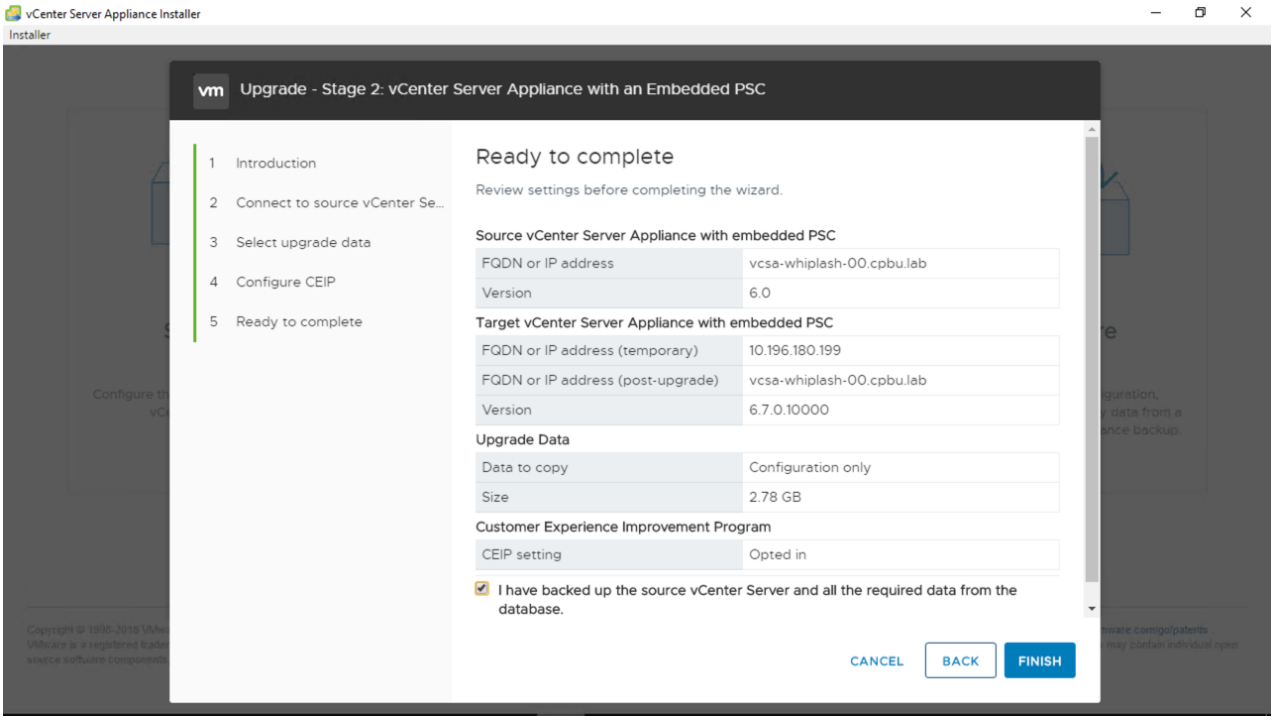
Step 4: Join the VMware Customer Experience Improvement Program or CEIP. Joining this program is optional but when you do join, it helps VMware to improve our products and services, fix problems and advise you on how best to deploy and use our products. CEIP is also required to enable the vSAN health check services.

To understand this better, please review additional information regarding the CEIP and its purpose.

Once a choice has been made, click **Next** to continue.

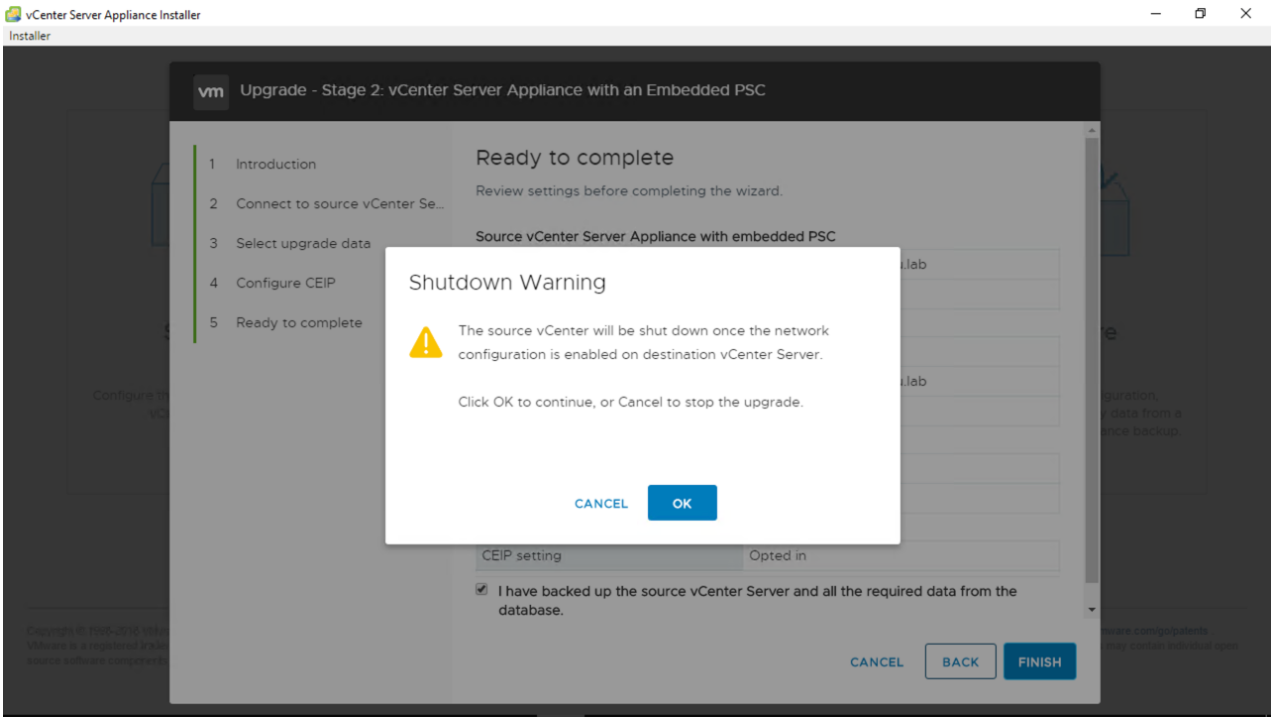


Step 5: Review all setting choices here and once complete, click **Finish** to continue.

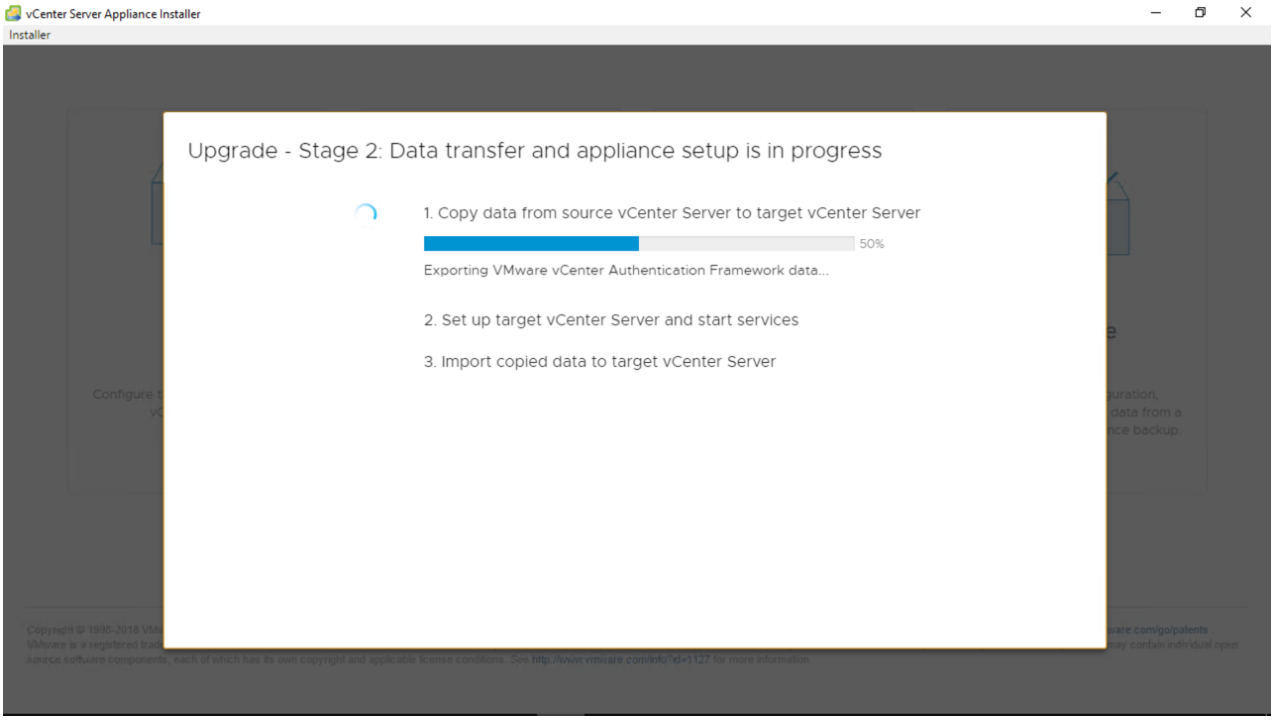


The prompt here is a reminder that the source vCenter Server will be powered down once all network configuration is enabled on the destination VCSA.

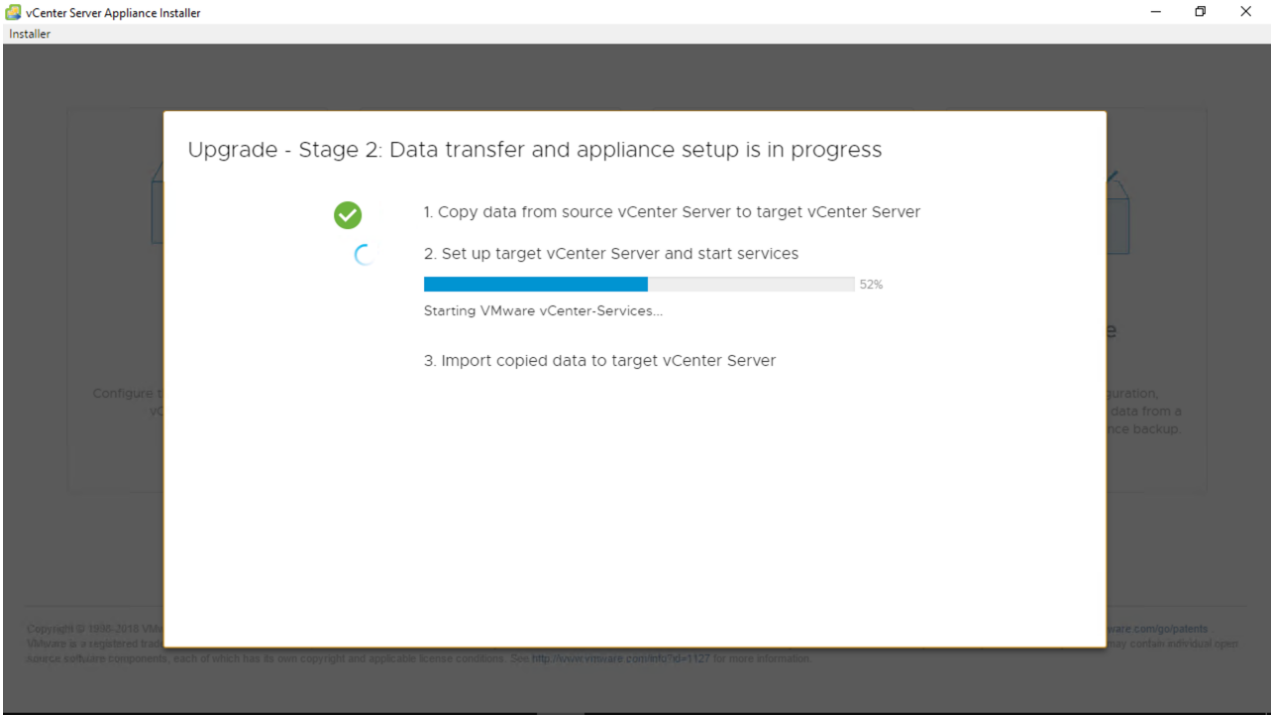
Click **OK** to continue.



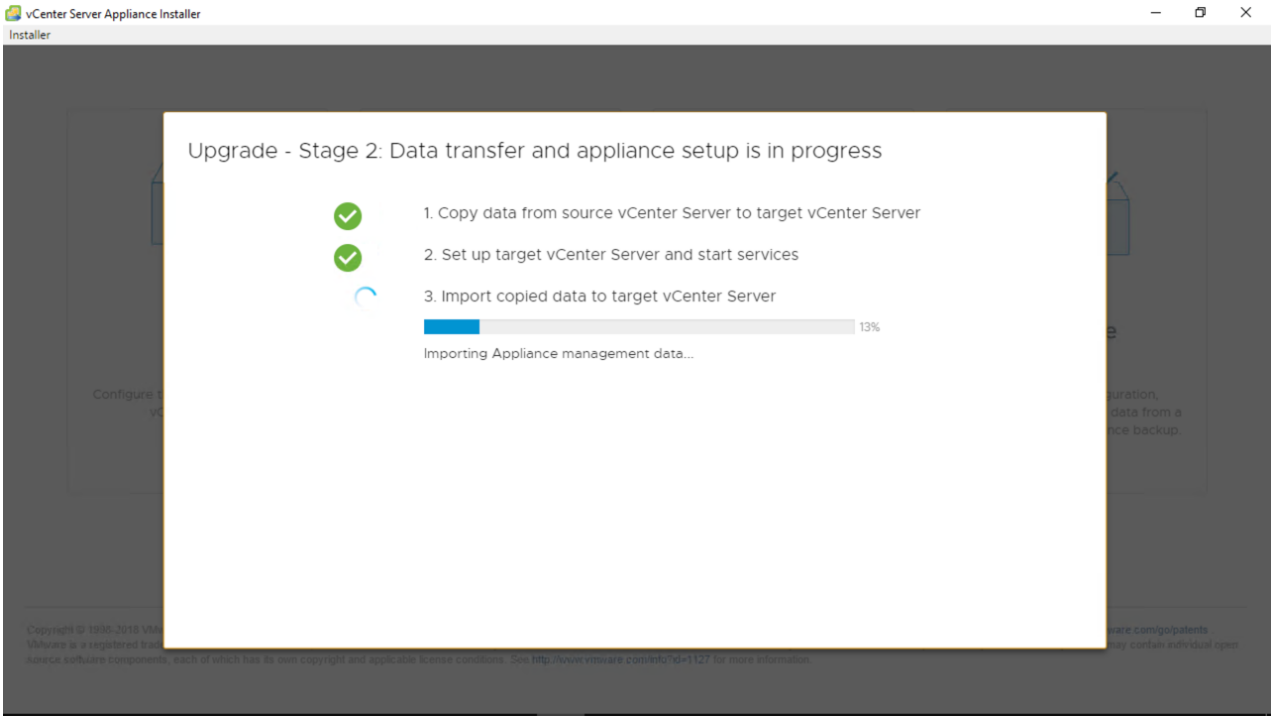
Stage 2 begins. Data is exported from the source vCenter Server and prepared for import.



Next, vCenter Server services are started on the destination VCSA.

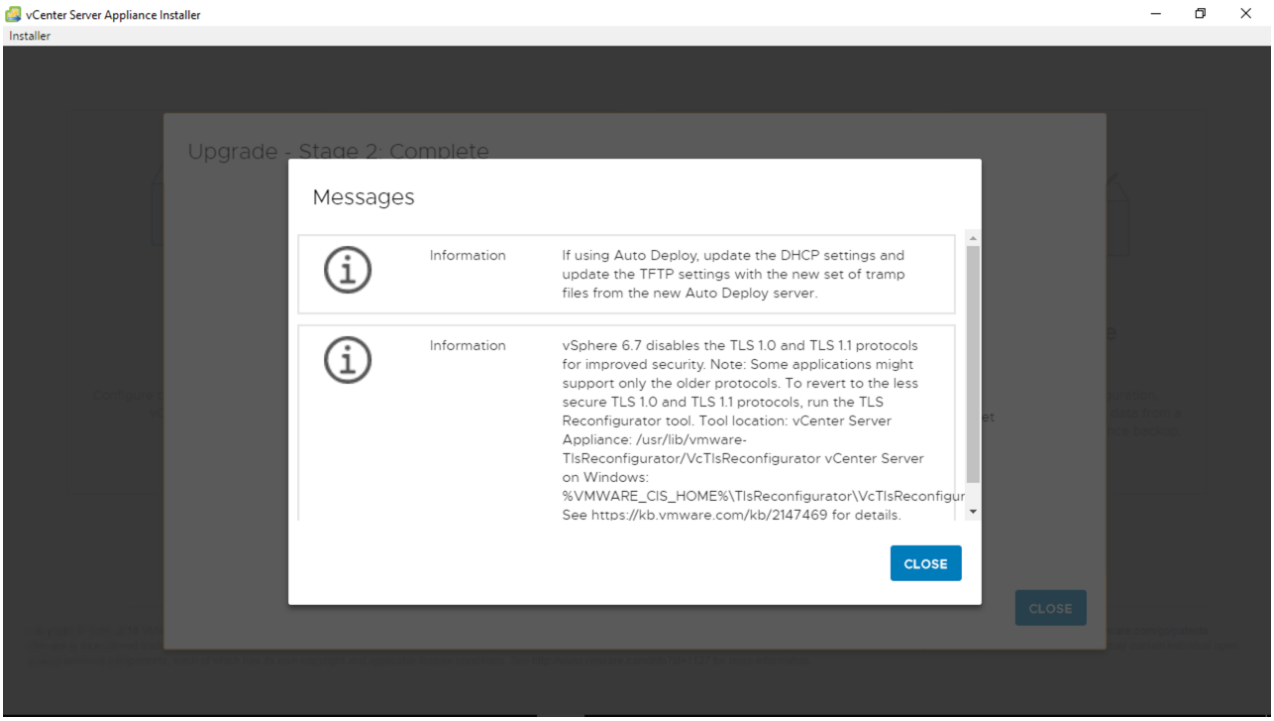


Last, the copied data from the source vCenter Server is imported to the destination VCSA.



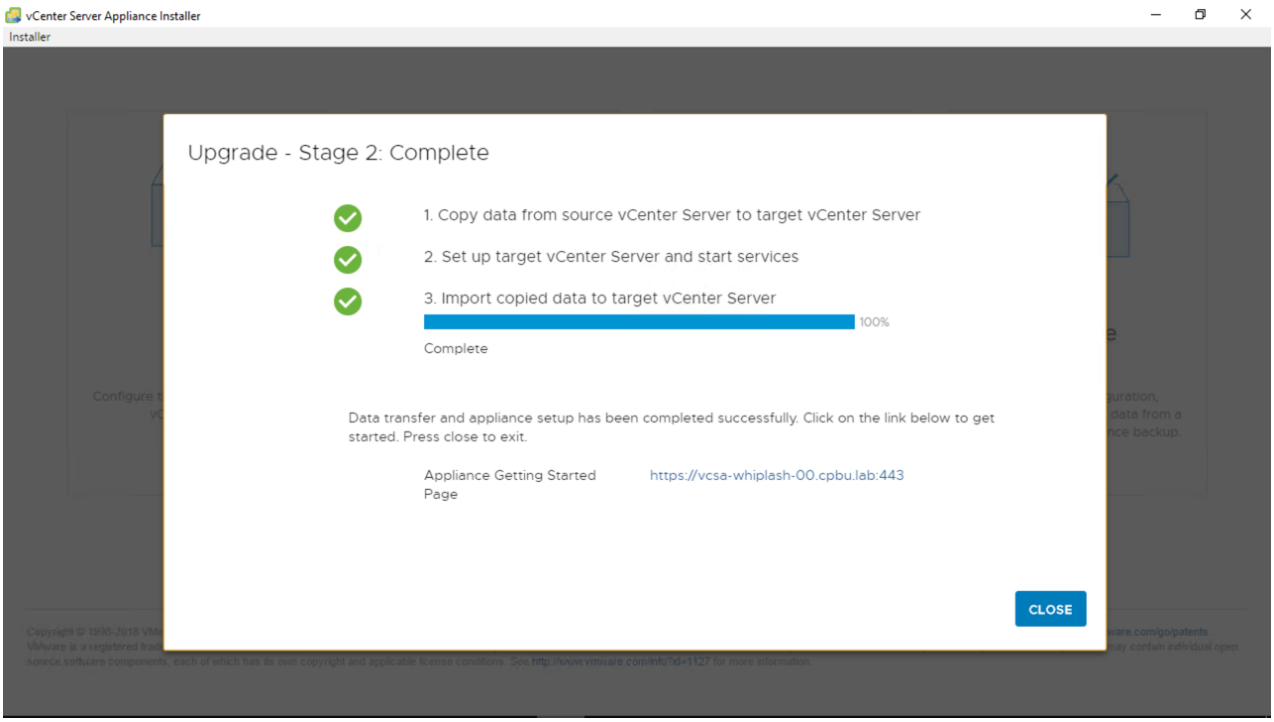
When all data has been imported to the destination VCSA, the process is complete. Messages are presented at this step as informational, such as a notice about TLS changes in vSphere 6.7.

Review these notices then click **Close** to continue.



Stage 2 of the vCenter Server Upgrade is now completed.

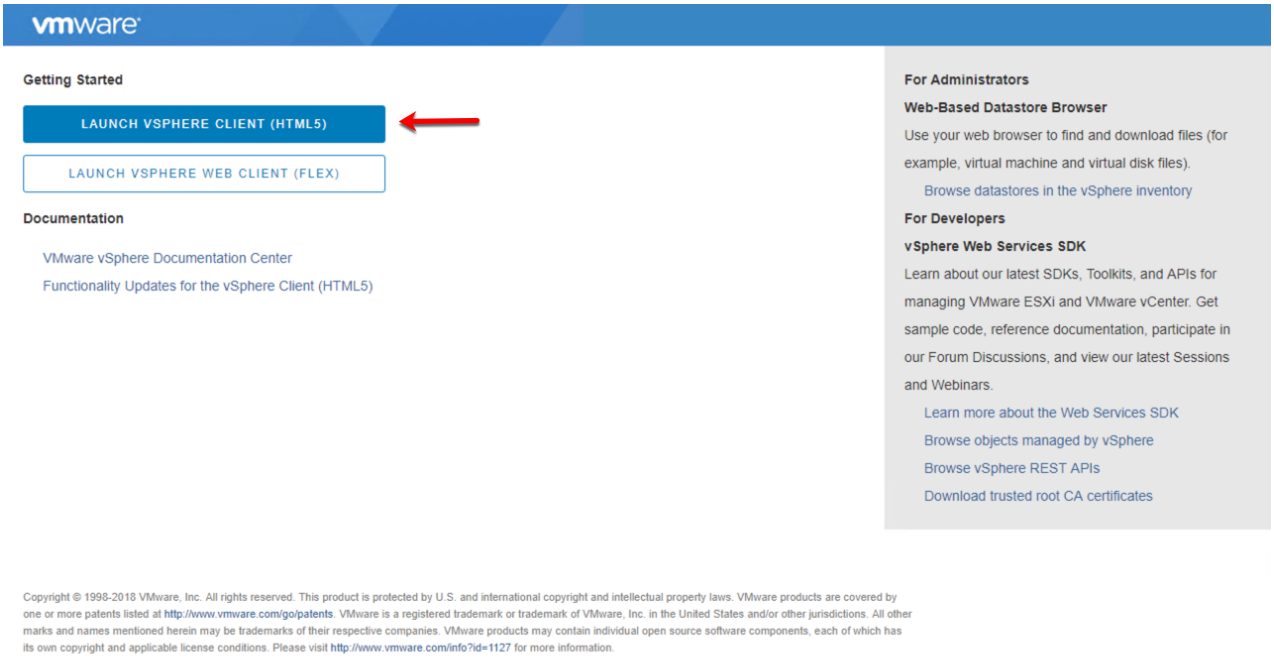
Click **Close** to continue.



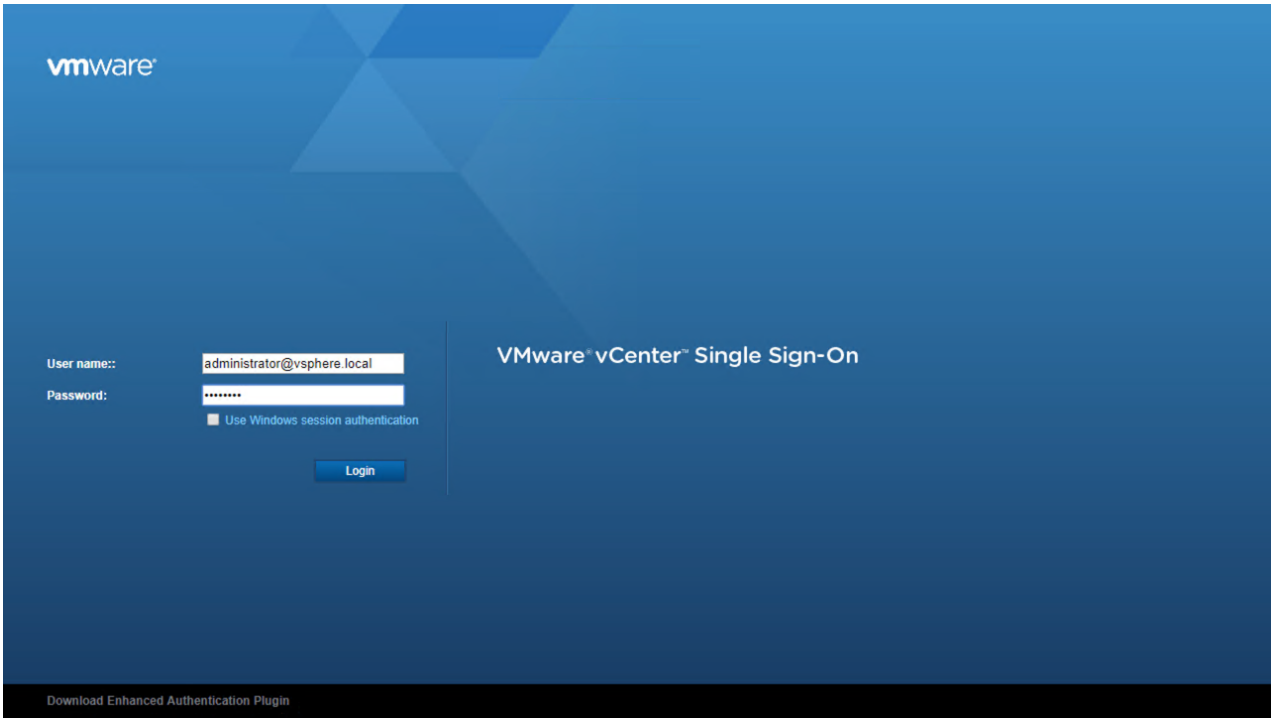
Closing the installer will launch the vCenter Server splash page allowing you to login via the vSphere Client on HTML5.

Since the vSphere Web Client (Flex) is now deprecated, I will use the HTML5 vSphere Client because it is now our default client and it contains 95% of all workflows as compared to vSphere 6.5 Update 1 that contained 90% of workflows. The HTML5 vSphere Client will be fully featured by the Fall of 2018.

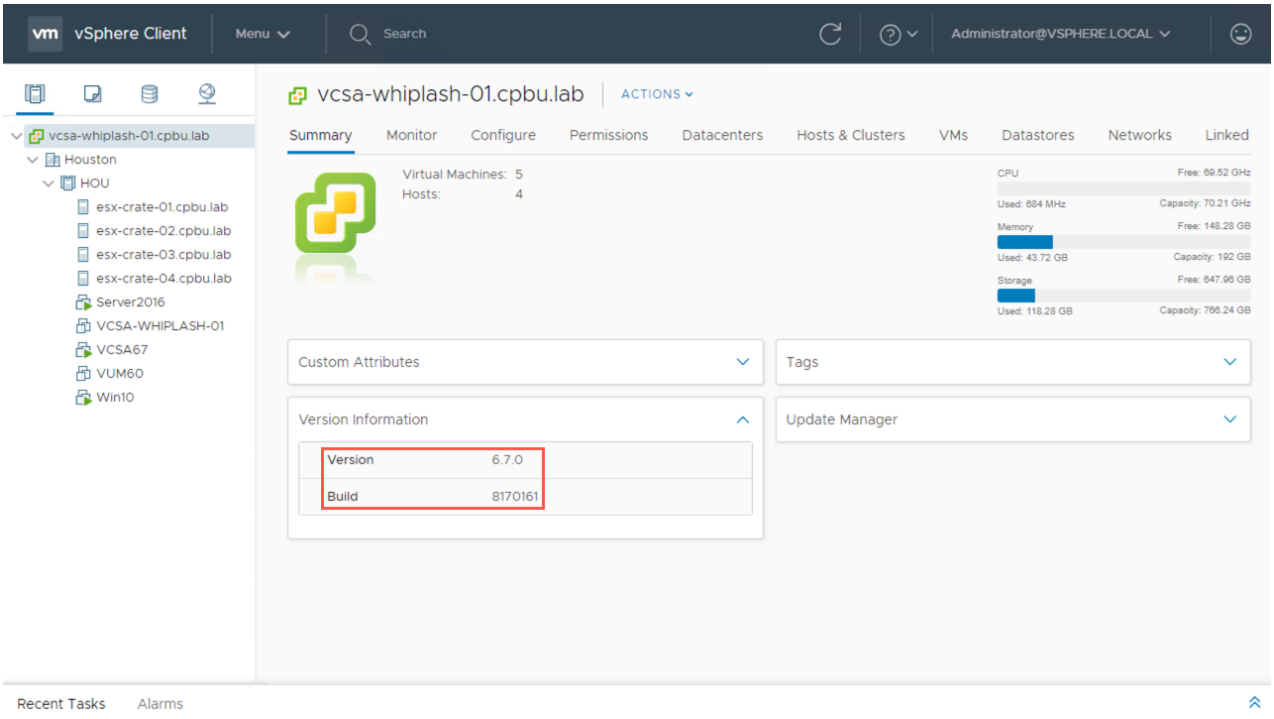
Click the button "**Launch vSphere Client (HTML5)**" to continue.



Enter the administrator credentials (*SSO administrator or other administrator with access to vSphere*) and login to view the vCenter Server and hosts.



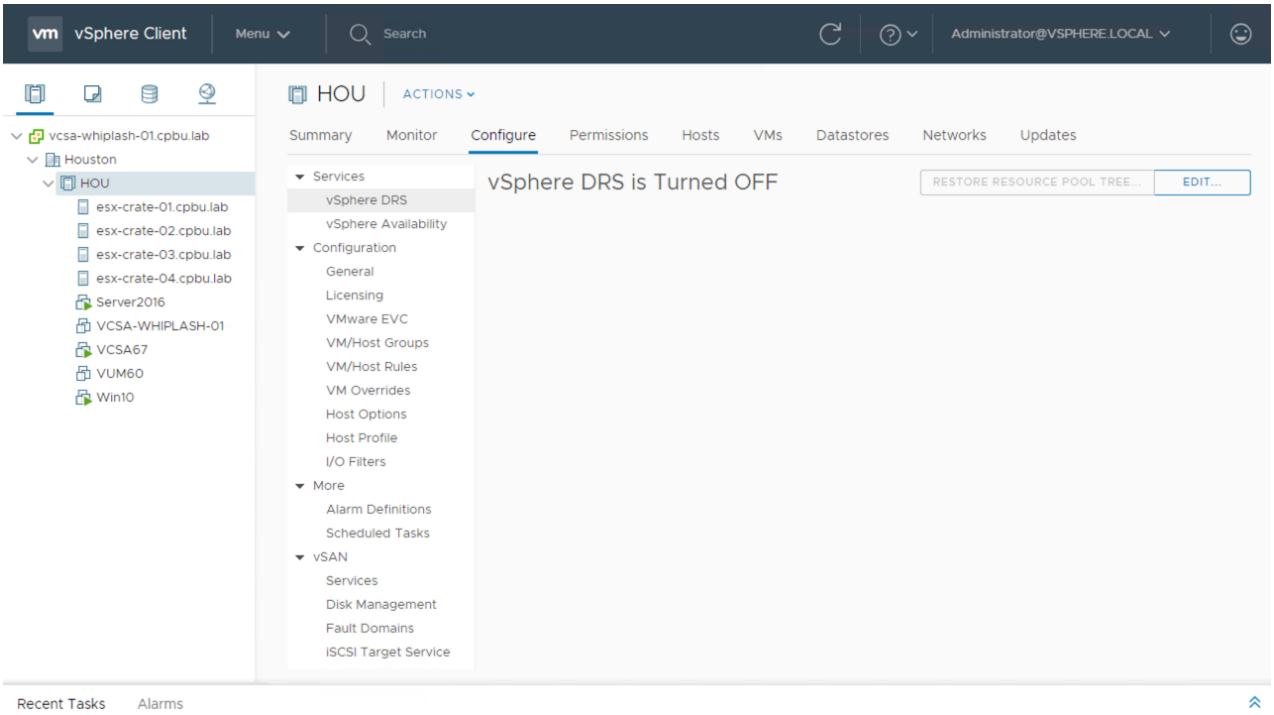
Now we will verify the vCenter Server is now running on version 6.7. We do this by clicking on the inventory name of the vCenter Server and viewing the ***Version Information*** from the **Summary** tab.



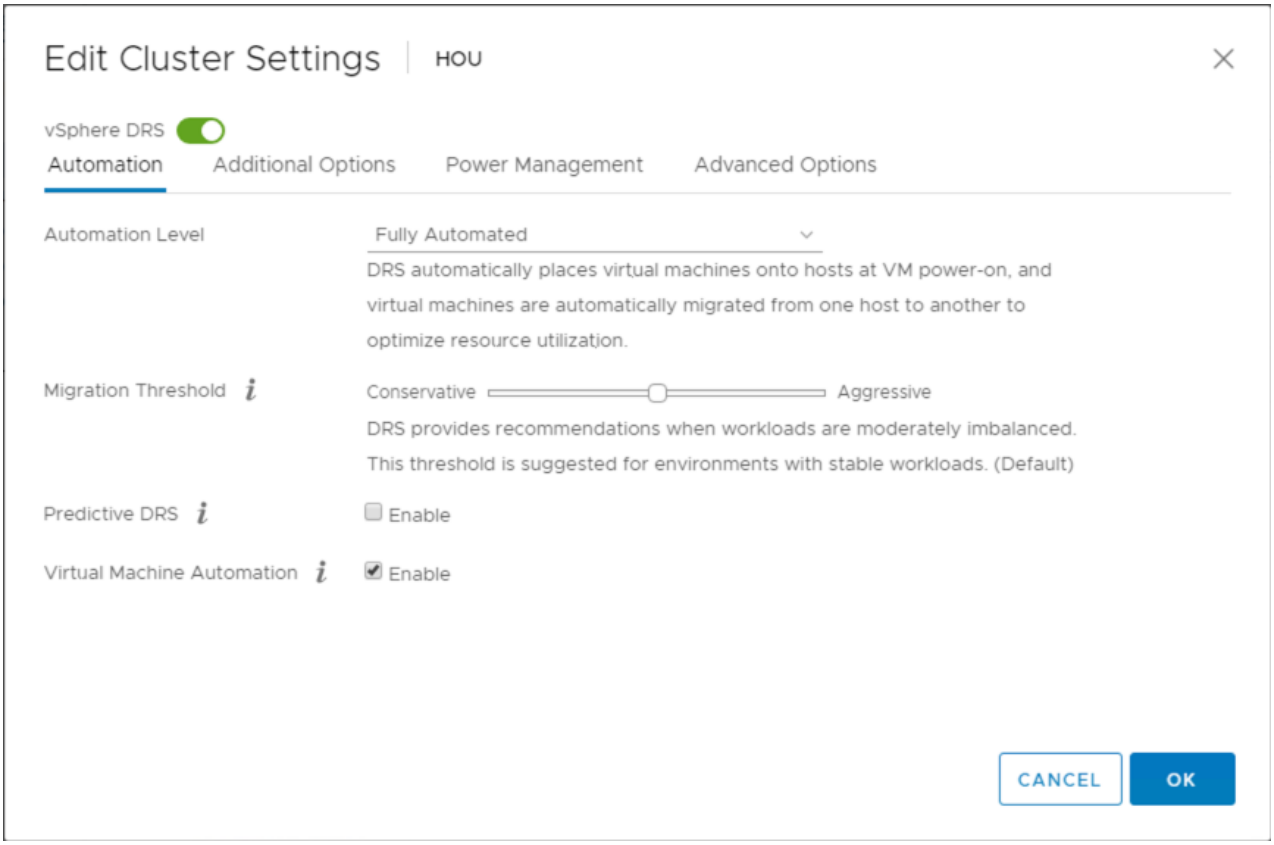
Enable DRS

Since we disabled DRS during our preparing to upgrade post, it should be enabled once again. We begin this task by highlighting the cluster and then clicking on the **Configure** tab to view vSphere DRS settings. If you had set DRS to ***Manual*** versus disabling, now is the time to change that also.

Next click **Edit** to continue.



In the ***Edit Cluster Settings*** window, click the slider switch to enable **vSphere DRS** then click **OK** to save. DRS is now enabled on the cluster.



NOTE: You can also quickly perform these actions via PowerCLI with the 'Set-Cluster' cmdlet.

Enable DRS:

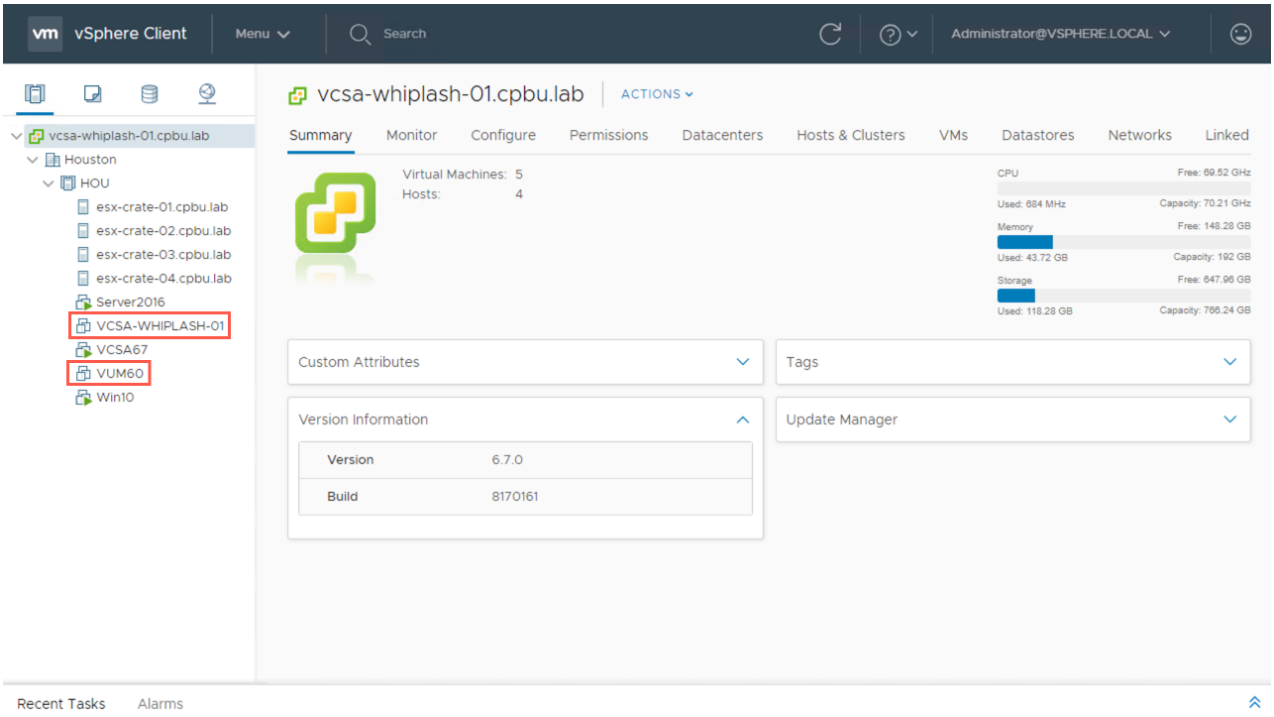
Set DRS automation level to Fully Automated:

Final Steps

After the vCenter Server Upgrade is completed, it is important to call out that power off operations only happen automatically for the vCenter Server, the VUM server must be powered off manually.

As a best practice, I suggest removing the Network Card from the old vCenter Server as well as renaming the VM to differentiate it within inventory and mitigate any accidental power on operations. Notice the new VCSA (VCSA67) virtual machine inventory name that was given. Note that renaming the VM does not change the FQDN of the VCSA, this is just the inventory name of the VM.

Lastly, be sure to power off the VUM server since it is no longer required to be running.



Rollback

Did your maintenance window close or maybe you encountered another issue during an Upgrade? Not to worry, rollback is quite simple. In this vSphere environment that we just upgraded we have no external PSCs, only the vCenter Server Appliance to worry about. If we did have an external PSC, we would first power off the newly deployed PSC, restore the PSC instance from backup, and if it was joined to an Active Directory domain, re-join it to the domain.

In our case without an external PSC we would, power off the newly deployed vCenter Server Appliance 6.7, bring the old vCenter Server Appliance 6.0 instance online (If already powered off, simply power it on. If not powered off, a restart is required), if it was joined to an Active Directory domain, it may need to be joined again (if your vCenter Server was Windows, be sure to have a local account on the server and do not rely on any cached credentials). Lastly we wait for all vCenter Server services to start and log in to the vSphere Web Client to verify the vSphere inventory.

Please review **KB2146453** for more guidance on rollbacks.

Conclusion

Remember; **Focus, Plan, Execute!**

We have successfully executed our vCenter Server Upgrade! We started by reviewing our prerequisites & compatibility, gathering our data, and then upgrading our vCenter Server and vSphere Update Manager (VUM) from vSphere 6.0 Update3 to 6.7. Our VUM was migrated to the 6.7 VCSA also during this upgrade scenario. In the next **vSphere Upgrade Series** post, we will move to upgrading our vSphere 6.0 ESXi hosts to vSphere 6.7 by utilizing vSphere Update Manager to remediate.

Please do not hesitate to post questions in the comments section of this blog or reach out to me directly via Twitter @vCenterNerd.

Article: https://blogs.vmware.com/vsphere/2018/09/vsphere-upgrade-series-part-4-upgrading-vmware-tools-and-vm-compatibility.html

Date: 2018-10-29