Kelverion

# Integration Pack for VMware vSphere

## *For Microsoft System Center Orchestrator*

For System Center 2016 and 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion_Integration_Pack_for_VMware_vSphere_4.0**

For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion_IP_VMware_vSphere_x64_4.0**

## User Guide

Version 4.0

# Kelverion Integration Pack for VMware vSphere

Published: January 2023

## Feedback

Send suggestions and comments about this document to support@kelverion.com

# Contents

# Kelverion Integration Pack for VMware vSphere

The **Integration Pack for VMware vSphere** an add-on for System Center Orchestrator that enables integration with VMware vSphere vCenter.

## System Requirements

The Integration Pack for VMware vSphere requires the following software to be installed and configured prior to implementing the integration. For more information about installing and configuring Orchestrator and VMware vSphere, refer to the respective product documentation.

### *Kelverion_Integration_Pack_for_VMware_vSphere (32-bit)*

- Microsoft System Center Orchestrator 2016, 2019
- Microsoft .NET Framework 4.7.2
- VMware vSphere vCenter 6.7, 7.0, 8.0

### *Kelverion_IP_VMware_vSphere_x64 (64-bit)*

- Microsoft System Center Orchestrator 2022
- Microsoft .NET Framework 4.7.2
- VMware vSphere vCenter 6.7, 7.0, 8.0

## System Requirements for Run PowerCLI Script

The integration pack can also be used for running PowerCLI scripts in your vSphere environment. The following software must be installed and configured:

- Windows PowerShell 5.1
- Windows Remote Management (WinRM)
- VMware PowerCLI 12.0.0, 13.0.0 (64-bit)

## Registering and Deploying the Integration Pack

After you download the integration pack, you register the integration pack file with the Orchestrator management server, and then deploy it to Runbook Servers and computers that have the Runbook Designer installed.

***IMPORTANT:*** Ensure that you are deploying the correct version of the Integration Pack.

- For System Center 2016 and 2019, you must use the 32-bit version of the integration pack, which has the name **Kelverion_Integration_Pack_for_VMware_vSphere**
- For System Center 2022 and later, you must use the 64-bit version of the integration pack, which has the name **Kelverion_IP_VMware_vSphere_x64**

### *To register the integration pack:*

1. On the management server, copy the **.OIP** file for the integration pack to a local hard drive or network share.
2. Confirm that the file is not set to **Read Only** to prevent unregistering the integration pack at a later date.
3. Start the **Deployment Manager**.
4. In the navigation pane of the Deployment Manager, expand **Orchestrator Management Server**, right-click **Integration Packs** to select **Register IP with the Orchestrator Management Server**. The **Integration Pack Registration Wizard** opens.
5. Click **Next**.
6. In the **Select Integration Packs or Hotfixes** dialog box, click **Add**.
7. Locate the **.OIP** file that you copied locally from step 1, click **Open** and then click **Next**.
8. In the **Completing the Integration Pack Wizard** dialog box, click **Finish**.
9. On the **End User Agreement** dialog box, read the Kelverion License Terms**,** and then click **Accept**.
10. The **Log Entries** pane displays a confirmation message when the integration pack is successfully registered.

***To deploy the integration pack:***

1. In the navigation pane of the **Deployment Manager**, right-click **Integration Packs**, click **Deploy IP to Runbook Server** or **Runbook Designer**.
2. Select the integration pack that you want to deploy, and then click **Next**.
3. Enter the name of the runbook server or computers with the Runbook Designer installed, on which you want to deploy the integration pack, click **Add**, and then click **Next**.
4. Continue to add additional runbook servers and computers running the Runbook Designer, on which you want to deploy the integration pack. Click **Next**.
5. In the **Installation Options** dialog box, configure the following settings.
6. To choose a time to deploy the integration pack, select the **Schedule installation** check box, and then select the time and date from the **Perform installation** list.
7. Click one of the following:
    a. **Stop all running runbooks before installing the integration pack** to stop all running runbooks before deploying the integration pack.
    b. **Install the Integration Packs without stopping the running Runbooks** to install the integration pack without stopping any running runbooks.
8. Click **Next**.

9. In the **Completing Integration Pack Deployment Wizard** dialog box, Click **Finish**.
10. When the integration pack is deployed, the **Log Entries** pane displays a confirmation message.

# Licensing the Integration Pack

After you register and deploy the integration pack you must provide a valid Kelverion license before running any runbooks that contain activities from the integration pack.

### *To deploy the integration pack license file to System Center Orchestrator 2019 or earlier:*

1. Copy the .KAL license file to %PROGRAMFILES(X86)%\Kelverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

### *To deploy the integration pack license file to System Center Orchestrator 2022 or later:*

1. Copy the .KAL license file to %PROGRAMFILES%\Kelverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

# Connecting to a vSphere Server

An Orchestrator configuration establishes a set of configuration properties which the integration pack uses to connect to one or more VMware vSphere servers. If your environment consists of a single vSphere server, use the Server URL, User Name and Password properties to specify connection information for that server. If your environment consists of multiple servers, you can use the additional configuration sets (1 to 9) to specify up to nine additional server configurations.

### *To set up a VMware vSphere configuration:*

1. In the Client, click the **Options** menu, and select *KA VMware vSphere*. The **KA VMware vSphere** dialog box appears.
2. On the **Configurations** tab, click **Add** to begin the configuration setup. The **Add Configuration** dialog box appears.
3. In the **Name** box, enter a name for the configuration. This could be the name of the vSphere server or a descriptive name to distinguish the type of configuration.
4. Click the ellipsis button (…) next to the **Type** box and select *VMware vSphere Configuration.*
5. In the **Configuration Name** box, enter a name to denote the server configuration comprised of the **Server URL**, **User Name** and **Password** properties. When using multiple configurations, the configuration name can be used in an activity to specify which server configuration that activity should be using to connect to a vSphere server.

   ***Note:*** Configuration Name is optional when using a single server configuration.

   ***Note:*** Configuration Name is required when using multiple server configurations, if you want this configuration to appear in the list of configuration options available for an activity.
6. In the **Server URL** box, enter the URL to your vCenter server. For example: https://myVCenter.myDomain.com or https://192.168.1.33.
7. In the **User Name** and **Password** boxes, enter the credentials used to administer your vSphere environment specified by **Server URL**. Example administrator@vsphere.local.

8. (Optional) In the **Configuration Name (1 to 9)** box, enter a name to denote the alternate server configuration comprised of the **Server URL (1 to 9)**, **User Name (1 to 9)** and **Password (1 to 9)** properties, respectively. Configuration Name (1 to 9) property is used when using multiple server configurations. The configuration name is used in an activity to specify which server configuration that activity should be using to connect to a vSphere server.
9. (Optional) In the **Server URL (1 to 9)** box, enter the URL for an alternate vCenter server. The Server URL (1 to 9) property is used when using multiple server configurations.
10. (Optional) In the **User Name (1 to 9)** and **Password (1 to 9)** boxes, enter the credentials used to administer your alternate vSphere environment specified by **Server URL (1 to 9)**. The User Name and Password (1 to 9) properties are used when using multiple server configurations.
11. (Optional) In the **Skip Certificate Validation** box, specify if you want the IP to perform server certificate validation or not. This applies only when connecting to the server over HTTPS. When set to True, the IP will not perform certificate validation. This is typically used in secure environments, when working with trusted servers and self-signed certificates. When set to False, the IP will validate the server certificate. The server must be configured with a valid certificate signed by a valid certificate authority and the server name in the **Server URL** must be listed on the certificate.
12. Click **OK** to close the configuration dialog box, and then click **Finish**.

## Connecting to a vSphere Server With PowerCLI

The integration pack can also be used for running PowerCLI scripts in your vSphere environment. In order to do so, in addition to vSphere connection information, you must also provide connection information for the machine where 64-bit PowerCLI is installed.

***To set up a VMware vSphere PowerCLI configuration:***

1. In the Client, click the **Options** menu, and select *KA VMware vSphere*. The **KA VMware vSphere** dialog box appears.
2. On the **Configurations** tab, click **Add** to begin the configuration setup. The **Add Configuration** dialog box appears.
3. In the **Name** box, enter a name for the configuration. This could be the name of the remote machine running PowerCLI, or a descriptive name to distinguish the type of configuration.
4. Click the ellipsis button (…) next to the **Type** box and select *VMware vSphere Script Configuration.*
5. In the **Configuration Name** box, enter a name to denote the server configuration comprised of the **vSphere Server**, **vSphere User Name** and **vSphere Password** properties. When using multiple configurations, the configuration name can be used in an activity to specify which server configuration that activity should be using to connect to a vSphere server.

   ***Note:*** Configuration Name is optional when using a single server configuration.

   ***Note:*** Configuration Name is required when using multiple server configurations, if you want this configuration to appear in the list of configuration options available for an activity.
6. In the **vSphere Server** box, enter the FQDN or IP address of your vSphere server. The integration pack will connect to the specified vSphere server, before running your script in the Run vSphere PowerCLI Script activity. For example: myVCenter.myDomain.com or 192.168.1.33.
7. In the **vSphere User Name** and **vSphere Password** boxes, enter the credentials used to administer your vSphere environment, for example administrator@vsphere.local. The integration pack will use these credentials to connect to the specified vSphere server, before running your script in the Run vSphere PowerCLI Script activity.
8. (Optional) In the **Configuration Name (1 to 9)** box, enter a name to denote the alternate server configuration comprised of the **vSphere Server (1 to 9)**, **vSphere User Name (1 to 9)** and **vSphere Password (1 to 9)** properties, respectively. Configuration Name (1 to 9) property is used when using multiple server configurations. The configuration name is used in an activity to specify which server configuration that activity should be using to connect to a vSphere server.
9. (Optional) In the **vSphere Server (1 to 9)** box, enter the FQDN or IP address for an alternate vSphere environment. The vSphere Server (1 to 9) property is used when using multiple server configurations.

10. (Optional) In the **vSphere User Name (1 to 9)** and **vSphere Password (1 to 9)** boxes, enter the credentials used to administer your alternate vSphere server specified by **vSphere Server (1 to 9)**. The vSphere User Name and vSphere Password (1 to 9) properties are used when using multiple server configurations.
11. (Optional) In the **vSphere Protocol** box, enter the protocol to be used when connecting to the vSphere server. Available values are **http** and **https.** HTTPS is used by default.
12. (Optional) In the **vSphere Port** box, enter the port to be used when connecting to the vSphere server. Port 443 is used by default.
13. In the **PowerCLI Server** box, enter the FQDN or the IP address of the machine where PowerCLI is installed.
14. In the **PowerCLI Server Port** box, enter the port used to communicate with the PowerCLI machine. Typical values are 5985 for HTTP and 5986 for HTTPS. Use the WinRM port for your environment.
15. In the **PowerCLI User** box, enter the user for connecting to the PowerCLI machine, in the form *DOMAIN\user.*
16. In the **PowerCLI Password** box enter the password for the PowerCLI User.
17. The **PowerCLI Use SSL** box specifies whether the IP uses HTTPS or HTTP to communicate with the PowerCLI machine. Select **True** to use SSL over HTTPS. Select **False** to use HTTP. Note that configuring this also requires the **PowerCLI Server Port** to be configured appropriately.
18. The **PowerCLI Skip CA Check** box specifies whether the client does not validate that the server certificate is signed by a trusted certificate authority (CA) when connecting over HTTPS (**PowerCLI Use SSL** box is **True**). Select **True** when the remote server is trusted, such as when it is part of a network that is physically secured and isolated, or when the remote computer is listed as a trusted host in a WinRM configuration.
19. The **PowerCLI Skip CN Check** box specifies whether the certificate common name (CN) of the server does not need to match the hostname of the server, when connecting over HTTPS (**PowerCLI Use SSL** box is **True**). Select **True** for trusted servers.
20. The **PowerCLI Skip Revocation Check** box specifies whether the revocation status of the server certificate is validated or not.
21. In the Configuration File Path box, enter the location of the integration pack configuration file. By default, this is the same location where the integration pack assemblies are installed.

    ***On System Center Orchestrator 2019 or earlier:***
    <Program Files (x86)>\Common Files\Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\A514130C-9FA9-44B3-897A-BCD5A0E965CC\Kelverion.IntegrationPack.Vmware.Vsphere.Configuration.json

    ***On System Center Orchestrator 2022 or later:***

<Program Files>\Common Files\Microsoft System Center 2012\Orchestrator\
Extensions\Support\Integration Toolkit\A514130C-9FA9-44B3-897A-BCD5A0E965CC\
Kelverion.IntegrationPack.Vmware.Vsphere.Configuration.json

22. Click **OK** to close the configuration dialog box, and then click **Finish**.

# Connecting to a vRealize Operations Manager Server with PowerCLI

The integration pack can also be used for running PowerCLI scripts in your vRealize Operations Manager (vROps) environment. In order to do so, in addition to vROps connection information, you must also provide connection information for the machine where 64-bit PowerCLI is installed.

***Important:*** The Run vROps PowerCLI Script activity is only supported as a professional services engagement. For details on how to configure and use this activity please contact Kelverion Professional Services.

***To set up a VMware vROps PowerCLI configuration:***

1. In the Client, click the **Options** menu, and select *KA VMware vSphere*. The **KA VMware vSphere** dialog box appears.
2. On the **Configurations** tab, click **Add** to begin the configuration setup. The **Add Configuration** dialog box appears.
3. In the **Name** box, enter a name for the configuration. This could be the name of the remote machine running PowerCLI, or a descriptive name to distinguish the type of configuration.
4. Click the ellipsis button (…) next to the **Type** box and select *VMware vROps Script Configuration.*
5. In the **Configuration Name** box, enter a name to denote the server configuration comprised of the **vROps Server**, **vROps User Name**, **vROps Password** and **vROps Authentication Source** properties. When using multiple configurations, the configuration name can be used in an activity to specify which server configuration that activity should be using to connect to a vROps server.

   ***Note:*** Configuration Name is optional when using a single server configuration.

   ***Note:*** Configuration Name is required when using multiple server configurations, if you want this configuration to appear in the list of configuration options available for an activity.
6. In the **vROps Server** box, enter the FQDN or IP address of your vRealize Operations Manager server. The integration pack will connect to the specified vROps server, before running your script in the Run vROps PowerCLI Script activity.
7. In the **vROps User Name** and **vROps Password** boxes, enter the credentials used to administer your vROps environment. The integration pack will use these credentials to

connect to the specified vROps server, before running your script in the Run vROps PowerCLI Script activity.

8. (Optional) In the **vROps Authentication Source** box, enter the authentication source type for the connection. If not specified, LocalUser is used. You can specify ActiveDirectory, OpenLDAP, or VC as authentication source type values.
9. (Optional) In the **Configuration Name (1 to 9)** box, enter a name to denote the alternate server configuration comprised of the **vROps Server (1 to 9)**, **vROps User Name (1 to 9)**, **vROps Password (1 to 9)** and **vROps Authentication Source (1 to 9)** properties, respectively. Configuration Name (1 to 9) property is used when using multiple server configurations. The configuration name is used in an activity to specify which server configuration that activity should be using to connect to a vROps server.
10. (Optional) In the **vROps Server (1 to 9)** box, enter the FQDN or IP address of your alternate vRealize Operations Manager server. The vROps Server (1 to 9) property is used when using multiple server configurations.
11. (Optional) In the **vROps User Name (1 to 9)** and **vROps Password (1 to 9)** boxes, enter the credentials used to administer your alternate vROps environment. The vROps User Name and vROps Password (1 to 9) properties are used when using multiple server configurations.
12. (Optional) In the **vROps Authentication Source (1 to 9)** box, enter the authentication source type for the alternate connection. If not specified, LocalUser is used. You can specify ActiveDirectory, OpenLDAP, or VC as authentication source type values. The vROps Authentication Source (1 to 9) property is used when using multiple server configurations.
13. (Optional) In the **vROps Port** box, enter the port to be used when connecting to the specified vROps server. Port 443 is used by default.
14. In the **PowerCLI Server** box, enter the FQDN or the IP address of the machine where PowerCLI is installed.
15. In the **PowerCLI Server Port** box, enter the port used to communicate with the PowerCLI machine. Typical values are 5985 for HTTP and 5986 for HTTPS.
16. In the **PowerCLI User** box, enter the user for connecting to the PowerCLI machine, in the form *DOMAIN\user.*
17. In the **PowerCLI Password** box, enter the password for the PowerCLI User.
18. The **Use SSL** box specifies whether the IP uses HTTPS or HTTP to communicate with the PowerCLI machine. Select **True** to use SSL over HTTPS. Select **False** to use HTTP. Note that configuring this also requires the **PowerCLI Server Port** to be configured appropriately.
19. The **PowerCLI Skip CA Check** box specifies whether the client does not validate that the server certificate is signed by a trusted certificate authority (CA) when connecting over HTTPS (**Use SSL** box is **True**). Select **True** when the remote server is trusted, such as when it

is part of a network that is physically secured and isolated, or when the remote computer is listed as a trusted host in a WinRM configuration.

20. The **PowerCLI Skip CN Check** box specifies whether the certificate common name (CN) of the server does not need to match the hostname of the server, when connecting over HTTPS (**Use SSL** box is **True**). Select **True** for trusted servers.
21. The **PowerCLI Skip Revocation Check** box specifies whether the revocation status of the server certificate is validated or not.
22. In the Configuration File Path box, enter the location of the integration pack configuration file. By default, this is the same location where the integration pack assemblies are installed.

    ***On System Center Orchestrator 2019 or earlier:***
    <Program Files (x86)>\Common Files\Microsoft System Center 2012\Orchestrator\
    Extensions\Support\Integration Toolkit\A514130C-9FA9-44B3-897A-BCD5A0E965CC\
    Kelverion.IntegrationPack.Vmware.Vsphere.Configuration.json

    ***On System Center Orchestrator 2022 or later:***
    <Program Files>\Common Files\Microsoft System Center 2012\Orchestrator\
    Extensions\Support\Integration Toolkit\A514130C-9FA9-44B3-897A-BCD5A0E965CC\
    Kelverion.IntegrationPack.Vmware.Vsphere.Configuration.json

23. Add additional connections if applicable.
24. Click **OK** to close the configuration dialog box, and then click **Finish**.

# WinRM Configuration

The integration pack requires WinRM to be installed and configured in order to connect to the machine where PowerCLI is running. WinRM has to be installed and configured on:

- The Orchestrator Runbook Service machine (WinRM client)
- The PowerCLI machine (WinRM server)

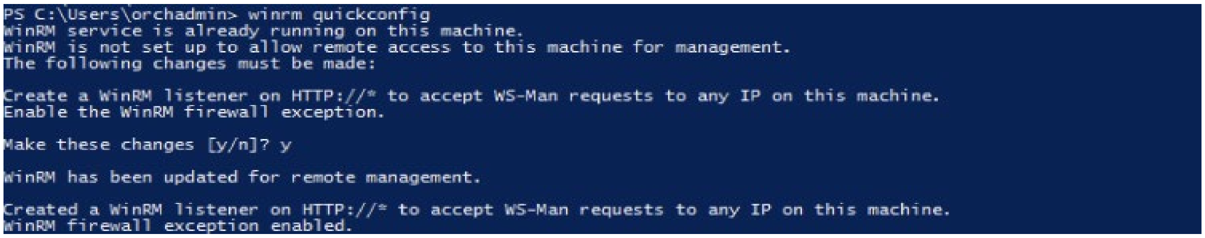
If the Orchestrator Runbook Service and PowerCLI are running on the same machine, both WinRM client and WinRM server settings must be configured the same machine.

***Steps for configuring WinRM Client and Server settings:***

1. PowerShell Script Execution Policy must be set to *Remote Signed* on the WinRM client and server machines. Start 64-bit Windows PowerShell and run the command:

   ```
   Set-ExecutionPolicy RemoteSigned
   ```

2. Configure a WinRM HTTP listener on the PowerCLI machine (WinRM server) by opening a PowerShell console (Run as Administrator) and running the *winrm quickconfig*.

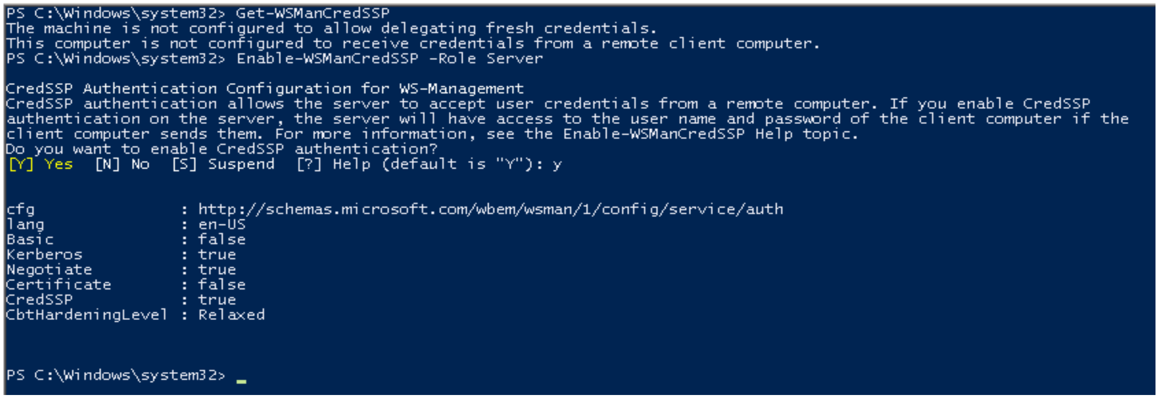


```
PS C:\Users\orchadmin> winrm quickconfig
WinRM service is already running on this machine.
WinRM is not set up to allow remote access to this machine for management.
The following changes must be made:

Create a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this machine.
Enable the WinRM firewall exception.

Make these changes [y/n]? y

WinRM has been updated for remote management.

Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this machine.
WinRM firewall exception enabled.
```

Similarly, if you want to configure a HTTPS listener, use *winrm quickconfig –transport:https*. Note that HTTPS requires a valid SSL certificate to be configured on the server machine.

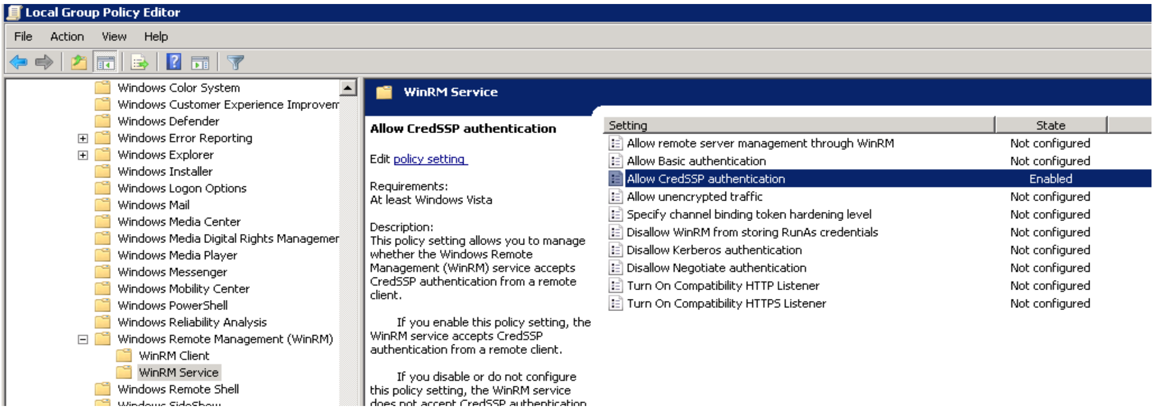
3. On the PowerCLI machine (WinRM server) and Orchestrator Runbook Service machine (WinRM client), make sure that CredSSP authentication is enabled, by running the following PowerShell command:

```
Enable-WSManCredSSP –Role Server
```



```
PS C:\Windows\system32> Get-WSManCredSSP
The machine is not configured to allow delegating fresh credentials.
This computer is not configured to receive credentials from a remote client computer.
PS C:\Windows\system32> Enable-WSManCredSSP -Role Server

CredSSP Authentication Configuration for WS-Management
CredSSP authentication allows the server to accept user credentials from a remote computer. If you enable CredSSP
authentication on the server, the server will have access to the user name and password of the client computer if the
client computer sends them. For more information, see the Enable-WSManCredSSP Help topic.
Do you want to enable CredSSP authentication?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): y


cfg               : http://schemas.microsoft.com/wbem/wsman/1/config/service/auth
lang              : en-US
Basic             : false
Kerberos          : true
Negotiate         : true
Certificate       : false
CredSSP           : true
CbtHardeningLevel : Relaxed


PS C:\Windows\system32> _
```

Or by using the group policy configuration utility gpedit.msc. Navigate to *Computer Configuration -> Administrative Templates -> Windows Components -> Windows Remote Management (WinRM) -> WinRM Service ->Allow CredSSP authentication* and make sure it this is enabled.



**Allow CredSSP authentication**

Edit policy setting

Requirements:
At least Windows Vista

Description:
This policy setting allows you to manage whether the Windows Remote Management (WinRM) service accepts CredSSP authentication from a remote client.

If you enable this policy setting, the WinRM service accepts CredSSP authentication from a remote client.

If you disable or do not configure this policy setting, the WinRM service does not accept CredSSP authentication

| Setting | State |
|---|---|
| Allow remote server management through WinRM | Not configured |
| Allow Basic authentication | Not configured |
| Allow CredSSP authentication | Enabled |
| Allow unencrypted traffic | Not configured |
| Specify channel binding token hardening level | Not configured |
| Disallow WinRM from storing RunAs credentials | Not configured |
| Disallow Kerberos authentication | Not configured |
| Disallow Negotiate authentication | Not configured |
| Turn On Compatibility HTTP Listener | Not configured |
| Turn On Compatibility HTTPS Listener | Not configured |

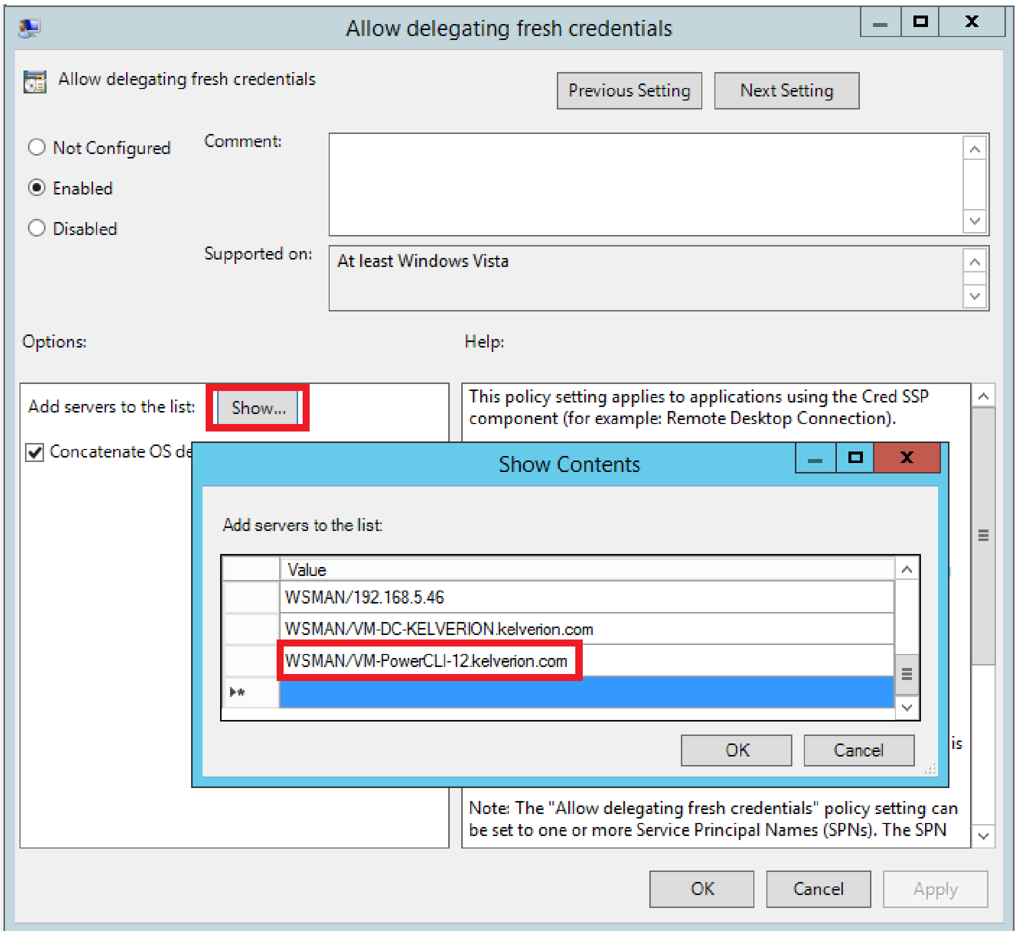
4. On the Orchestrator Runbook Service machine (WinRM client) allow delegation of credentials to PowerCLI machine (WinRM server) by running the following PowerShell command

   ```
   Enable-WSMANCredSSP -Role Client -delegatecomputer <WinRM Server
   FQDN> -force
   ```

   or

   by running the group policy configuration utility gpedit.msc. Navigate to *Computer Configuration -> Administrative Templates -> System -> Credentials Delegation -> Allow Delegating Fresh Credentials*. Make sure this is enabled. Add the PowerCLI machine to the list of servers. This must be in the form *wsman/<PowerCLI machine FQDN>*.

***Important:*** If your Orchestrator machine and the PowerCLI machine are not part of the same domain, you may have to configure *Allow Delegating Fresh Credentials with NTLM-only server authentication* in a similar fashion.



○ Not Configured
● Enabled
○ Disabled

Supported on: At least Windows Vista

Add servers to the list:

☑ Concatenate OS de

This policy setting applies to applications using the Cred SSP component (for example: Remote Desktop Connection).

Note: The "Allow delegating fresh credentials" policy setting can be set to one or more Service Principal Names (SPNs). The SPN

Add servers to the list:

| | Value |
|---|---|
| | WSMAN/192.168.5.46 |
| | WSMAN/VM-DC-KELVERION.kelverion.com |
| | WSMAN/VM-PowerCLI-12.kelverion.com |
| ▶* | |

## Managed Object Reference (MOR)

Managed Object Reference (MOR or MoRef) is the primary identifier used by the vSphere IP to refer to Managed Objects, which are server-side vSphere objects such as Datacenter, Virtual Machine, etc., which can perform operations invoked by the IP. The IP makes extensive use of MORs, with every activity either requiring them as inputs or publishing as outputs.

The general convention used throughout the IP to label MOR properties is the following format:

***<managed object type> (MOR)***

For example, “VM (MOR)”, “Datastore (MOR)”, “Folder (MOR)”, etc.

The MOR itself, represented as a string value, has the following format:

***<managed object type>-<instance value>***

For example, “VirtualMachine-vm-57” is the MOR for a VM, where “VirtualMachine” is the managed object type and “vm-57” is the instance value. Some other examples of MORs: “Datastore-datastore-32”, “Folder-group-v62”. You can refer here for a complete list of managed object types in the vSphere API.

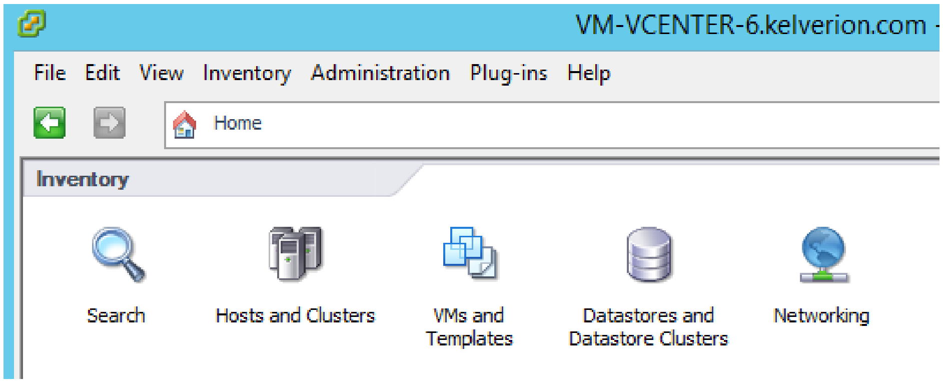
***Note:*** MORs are only guaranteed to be unique within the same vCenter environment. The IP supports multiple individual vSphere vCenter configurations - you can configure and interact individually with multiple vCenter environments from your Orchestrator runbooks. However, the IP does not support interactions across vCenter environments – you cannot refer to objects in one vCenter environment using MORs obtained from a different vCenter environment.

In addition to Managed Objects, the vSphere IP also uses Data Objects to send and retrieve data from the server-side managed objects. Data objects are identified in other ways other than managed object references and are often transitory, meaning that they do not live beyond the context of IP activity implementation.

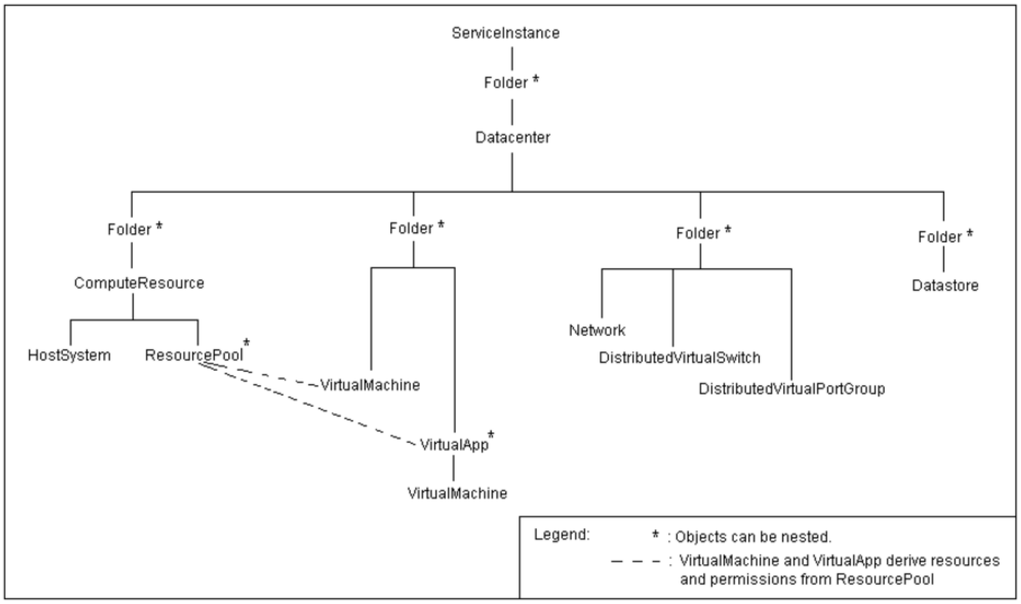
## vSphere Inventory Structure

The vSphere API organizes managed object entities (Datacenter, HostSystem, VirtualMachine, etc.) in an inventory hierarchy.

Here are the main inventory groups as displayed in the VMware vSphere Client:



The underlying vSphere API vCenter Inventory Hierarchy is more complex, however the similarities are apparent:



At top level in the hierarchy are Datacenter folders which can contain datacenters and other datacenter folders. Each datacenter contains the following sub-hierarchies:

- Compute Resource Hierarchy
- Virtual Machine Hierarchy
- Network Hierarchy
- Datastore Hierarchy

The **Compute Resource Hierarchy** consists of Compute Resource folders which can contain Compute Resources, Cluster Compute Resources and other Compute Resource folders.

Each **Compute Resource** contains:

- One Host System
- Resource Pool Hierarchy – one or more resource pools which in turn can contain child resource pools.

Each **Cluster Compute Resource** contains:

- One or more Host System
- Resource Pool Hierarchy – one or more resource pools which in turn can contain child resource pools.

The **Virtual Machine Hierarchy** consists of Virtual Machine folders which can contain Virtual Machines, Virtual Apps and other Virtual machine folders. Virtual Apps can contain other Virtual Apps and also Virtual Machines.

The **Network Hierarchy** consists of Network folders which can contain Networks, Distributed Virtual Switches and Distributed Virtual Port Groups.

The **Datastore Hierarchy** consists of Datastore folders which can contain Datastores.

Looking at the vSphere API inventory, there are a number of things worth noting in relation to the vSphere integration pack:

- A non-template virtual machine must always be associated with a resource pool.
- A compute resource (or cluster compute resource) is considered to be the “owner” of any resource pool in the resource pool hierarchy it contains.
- Each vSphere managed object has a “parent” property:
    - The parent of a virtual machine can be a folder or a virtual app.
    - The parent of a host can be a compute resource or cluster compute resource.
    - The parent of a resource pool can be a resource pool or a compute resource or a cluster compute resource.
    - The parent of a datastore can be a folder.
    - The parent of a network can be a folder.
    - The parent of a folder can be another folder or a datacenter.
- For each datacenter the server automatically creates the following hidden folders:
    - A “vm” folder for virtual machines, templates and virtual apps
    - A “host” folder for Compute Resources and Cluster Compute Resources
    - A “network” folder
    - A “datastore” folder

  These default folders are transparent in the vSphere Client and Web UI. For example, in the UI it looks like you can create a VM directly under a datacenter when in fact that VM is placed in the “vm” folder under that datacenter.

# PowerCLI Script Activity Configuration

The integration pack also provides activities for running VMware PowerCLI scripts. When using PowerCLI script activities, you can optionally use the provided JSON configuration file to define custom parameters and outputs for your script.

- Custom script parameters will appear as input properties in PowerCLI script activities, which can be used to pass data to your script.
- Custom outputs can be used to select and publish relevant script outputs to the Orchestrator data bus and reduce the need to process larger amounts of published data.

The integration pack installs the *Kelverion.IntegrationPack.Vmware.Vsphere.Configuration.json* configuration file in the same location where the integration pack binaries are installed:

> *C:\Program Files (x86)\Common Files\Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\A514130C-9FA9-44B3-897A-BCD5A0E965CC*

The installed configuration file contains sample script configurations which can be used in the PowerCLI script activities. You can create additional configurations or edit the existing configurations. Make sure to back up the configuration file before making changes.

***Important:*** We strongly recommend users locate their custom JSON configuration file(s) in a location outside of the IP folder such as a network share that is visible from all Runbook Server and Runbook Designer machines. The example JSON configuration file inside the IP folder will be replaced each time you perform an IP upgrade, so we recommend you change the file name and location.

## ScriptConfig

The **PowerCLI/ScriptConfig** element contains multiple script configurations that can be used in multiple activities. When you want to run a new PowerCLI script, you can add a new ScriptConfig section and configure activity inputs and published data specific to your script inputs and outputs.

```
"PowerCLI": {
   "ScriptConfig": [
   {
       "Name": "Get Host Sample",
    "Parameters": [...],
    "Outputs": [...],
   },
   ... ]}
```

You can specify the following properties in the ScriptConfig element:

- **Name** – specifies a unique name for this script configuration.
- **Parameters** – Defines zero or more **Parameter** elements. The activity will provide design time input properties for each defined parameter. The activity input runtime value will be available in the PowerCLI script as a PowerShell parameter.

- **Outputs** – Defines zero or more **Output** elements. At runtime, the activity will publish the PowerCLI script outputs on the Orchestrator data bus.

## Parameter

Defines a script parameter. For each defined script parameter, the activity will provide an activity input property.

```
"Parameters": [
{
    "Name": "NameFilter",
    "InputName": "VM Name Filter",
    "DataType": "System.String"
},
... ]
```

You can specify the following parameter properties:

- **Name** – specifies the script parameter name. For example, when specifying the name **NameFilter** in the JSON configuration file, the PowerShell $NameFilter parameter will be available to be used by your script at runtime.
- **InputName** – specifies the activity input name, corresponding to this parameter.
- **DataType** – specifies the parameter data type. Supported values are:
    - System.String
    - System.Security.SecureString
    - System.Boolean
    - System.DateTime
    - System.Int32
    - System.Int64
    - System.Single
    - System.Double
    - System.Decimal
- **IsPassword** – specifies whether the string input value will be obfuscated in the Orchestrator Runbook Designer. Supported values are **true** and **false**.

***Note:*** When defining parameters, avoid input names which have already been defined in the activity configuration options, for example vSphere Server, vSphere Username, vSphere Password, etc., as this can result in configuration options values being overwritten by activity values at runtime.

## Output

Defines a PowerCLI script output. The integration pack will expect to find a PowerShell property with this name in the list of PowerShell objects returned by the script. Output values will be published at runtime on the Orchestrator data bus.

```
"Outputs": [
{
    "Name": "Name",
    "PublishName": "VM Name",
    "DataType": "System.String"
},{
   "Name": "PowerState",
  "PublishName": "Power State",
  "DataType": "System.String"
},

... ]
```

You can specify the following output properties:

- **Name** – specifies the output name. This name must match the PowerShell output name.
- **PublishName** - specifies the Orchestrator published data name. The integration pack will publish an activity output with this name on the Orchestrator data bus.
- **DataType** – specifies the output data type. Supported values are:
    - System.String
    - System.Boolean
    - System.DateTime
    - System.Int32
    - System.Int64
    - System.Single
    - System.Double
    - System.Decimal

# Integration Pack Activities

This integration pack adds the KA VMware vSphere category to the **Activities** pane in the Orchestrator Runbook Designer.

## vSphere Activities

The following activities can be used to integrate with your vSphere vCenter environment:

- Add VM Device
- Create VM
- Create VM From Template
- Delete VM
- Get Custom Attribute
- Get vSphere Object
- Move VM
- Restart VM
- Set Custom Attribute
- Start VM
- Stop VM
- Suspend VM
- Update VM

## Run PowerCLI Script Activities

The integration pack also provides the following activities for running VMware PowerCLI scripts.

- Run vSphere Script
- Run vROps Script

## Common Configuration Instructions for All Activities

The following configuration instructions apply to all activities in this integration pack. Links to this section are included in the configuration instructions for each activity.

### Activity Properties

Each activity has a set of required or optional properties that define the configuration of that activity. This includes how it connects to other activity or how the activity performs its actions. You can view or modify activity properties in the Orchestrator Client.

***To configure the properties for an activity:***

1. Double-click the activity. Alternatively, you can right-click the activity, and then click **Properties**.
2. To save your configuration entries, click **Finish**.

In the activity properties dialog box, several tabs along the left side provide access to general and specific settings for the activity. Although the number of available tabs for activity properties differs from activity to activity, all activities will have a **General** tab, a **Properties** tab and/or **Filters** tab, and a **Run Behavior** tab. Some activities may have additional tabs.

## General Tab

This tab contains the **Name** and **Description** properties for the activity. By default, the **Name** of the activity is the same as its activity type, and the **Description** is blank. You can modify these properties to create more descriptive names or provide detailed descriptions of the actions of the activity.

## Properties/Filters Tab

These tabs contain properties that are specific to the activity.

All activities in this integration pack have the **Configuration Name** property at the top of the **Properties** tab. This property is used to specify the connection to a VMware vSphere server.

***To configure the Configuration Name property:***

1. Click the ellipsis **(…)** button next to the **Name** field, and then select the applicable connection name. Connections displayed in the list have been previously configured as described in Configuring the Integration Pack.

### *Filter Behavior*

The Monitor and Get activities use filters to determine the values that will invoke a runbook or retrieve activities. Property values of potential candidates are compared to the values of the filters to determine if they meet the criteria. When matching against values, you select one of the available methods of comparison. An option is provided to either match or not match the filter using each method. For example, the "Does not" version of a method causes alerts that do not match the filter to trigger the runbook. The Monitor activity will only trigger if all filters match and the Get activity will only return Alarms that match all filters.

- **Equals**: the field of the record exactly matches the text or number specified in the filter.
- **Does not equal**: the field of the record does not exactly match the text or number specified in the filter.
- **Is less than**: the field of the record is less than the number specified in the filter.
- **Is less than or equal to**: the field of the record is less than or equal to the number specified in the filter.
- **Is greater than**: the field of the record is greater than the number specified in the filter.
- **Is greater than or equal to**: the field of the record is greater than or equal to the number specified in the filter.
- **Contains**: the field of the record contains the text specified in the filter.
- **Does not contain**: the field of the record does not contain the text specified in the filter.
- **Starts with**: the field of the record starts with the text specified in the filter.

- **Ends with**: the field of the record ends with the text specified in the filter.

### *Server-Side Filtering vs Client-Side Filtering*

Some of the Get activities in the IP implement server-side filtering, which in general is preferred to client-side filtering because it reduces the amount of data transferred between the server and the IP. However, server-side filtering depends on the capabilities of the API and often not all filter operators listed above are supported when retrieving data from the server. For the vSphere IP, activities which support server-side filtering only provide the Equals filter operator.

Some Get activities provide Client-side filtering, in cases when server-side filtering is not available, or, when data is published in a format which cannot be filtered on the server. Client-side filters operate on un-filtered data which retrieved from the server and their main advantage is that the IP has more control on supporting filter operators. The IP activities which implement Client-side filtering provide all the filter operators listed above. Also, these activities retrieve relatively small amounts of data which should not overload the server or the IP system.

## Run Behavior Tab

This tab contains the properties that determine how the activity handles multi-value published data and what notifications will be sent if the activity fails or runs for an excessive period of time.

### *Multi-Value Published Data Behavior*

The Get activities retrieve information from another activity or outside source and can return one or more values in the published data. For example, when you use the Get Collection Member activity, the data output from that activity might be a list of computers that belong to the specified collection.

By default, the data from the Get activity will be passed on as multiple individual outputs. This invokes the next activity as many times as there are items in the output. Alternatively, you can provide a single output for the activity by enabling the **Flatten** option. When you enable this option, you also choose a formatting option:

- **Separate with line breaks**. Each item is on a new line. This format is useful for creating human-readable text files for the output.
- **Separate with _** . Each item is separated by one or more characters of your choice.
- **Use CSV format**. All items are in CSV (comma-separated value) format. This format is useful for importing data into spreadsheets or other applications.

The activity will produce a new set of data every time it runs. The **Flatten** feature does not flatten data across multiple instances of the same activity.

### *Event Notifications*

Some activities are expected to take a limited amount of time to complete. If they do not complete within that time they may be stalled or there may be another issue preventing them from completing. You can define the number of seconds to wait for completion of the action. After this period a platform event will be sent, and the issue will be reported. You can also choose whether to generate a platform event if the activity returns a failure.

### *To be notified when the activity takes longer than a specified time to run or fails to run:*

1. In the **Event Notifications** box, enter the **number of seconds** of run time before a notification is generated.
2. Select **Report if activity fails to run** to generate run failure notifications.

For more information about Orchestrator events, see the “Event Notifications " topics in the Runbook Properties (https://technet.microsoft.com/en-us/library/hh489610.aspx#Event Notifications).

## Published Data

Published data is the foundation of a working runbook. It is the data produced as a result of the actions of an activity. This data is published to an internal data bus that is unique for each runbook. Subsequent activities in the runbook can subscribe to this data and use it in their configuration. Link conditions also use this information to add decision-making capabilities to runbooks.

An activity can only subscribe to data from the activities that are linked before it in the runbook. You can use published data to automatically populate the property values needed by activities.

### *To use published data:*

1. Right-click the property value box, click **Subscribe**, and then click **Published Data**.
2. Click the **Activity** drop-down box and select the activity from which you want to obtain the data.
3. To view additional data elements common to all runbooks, select **Show Common Published Data**.
4. Click the published data element that you want to use, and then click **OK**.

For a list of the data elements published by each activity, see the Published Data tables in the activity topic.

# Add VM Device

The **Add VM Device** activity can used in a runbook to add new devices to an existing VM. The **Add VM Device** property specifies the type of device to be added.

## *Required Properties*

You must first configure the **Add VM Device** property with the type of device you want to add. Additional properties will be provided after you make this selection.

| | |
|---|---|
| **Add VM Device** | Specifies the type of device to be added. Options include:<br>• CD/DVD Drive<br>• Disk Drive<br>• NIC |

## *Required Properties (CD/DVD Drive)*

You must configure the following properties when adding a **CD/DVD Drive** to the VM.

| | |
|---|---|
| **Add VM Device** | Specifies the type of device to be added |
| **VM (MOR)** | Managed object reference of the VM being updated. |
| **CD/DVD Drive Type** | Drive type:<br>• **Client Device** - connect the device to a physical CD/DVD device on the system from which you access the vSphere Client.<br>• **Host Device** - connect the device to a physical CD/DVD device on the host.<br>• **Datastore ISO File** - connect the device to an ISO file that is stored on a datastore accessible to the host. |
| **CD/DVD Drive Host Device Path** | Location of host device. This property is available only when *CD/DVD Drive Type = "Host Device"* |
| **CD/DVD Drive ISO File Path** | Location of the datastore ISO file for the device. This property is available only when *CD/DVD Drive Type = "Datastore ISO File"* |

## *Required Properties (Disk Drive)*

You must also configure the following properties when adding a **Disk Drive** to the VM.

| | |
|---|---|
| **Add VM Device** | Specifies the type of device to be added |
| **VM (MOR)** | Managed object reference of the VM being updated. |
| **Disk Size (MB)** | The size of the new disk drive, in MB. |
| **Disk Provisioning** | Disk provisioning for the new VM disk. |

## Required Properties (NIC)

You must also configure the following properties when adding a **NIC** to the VM.

| | |
|---|---|
| **Add VM Device** | Specifies the type of device to be added |
| **VM (MOR)** | Managed object reference of the VM being updated. |
| **Network (MOR)** | Managed object reference of the network or distributed virtual port group to be assigned to the new NIC. |
| **NIC Type** | Network interface controller card type to be added. This should be one of the NICs available for the Compute Resource (or Cluster Compute Resource) owner of the Resource Pool of the VM, and it should be supported by the operating system of the VM.<br>To obtain the NIC Type you can use the Get vSphere Object activity with *Object Type* = "NIC Option".<br>For more details on choosing a NIC for your VM see VMware KB1001805. |

## Optional Properties

You can use the following optional properties.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |

## Optional Properties (CD/DVD Drive)

You can use the following properties, as necessary, when adding a **CD/DVD Drive** to the VM.

| | |
|---|---|
| **CD/DVD Drive ISO Datastore (MOR)** | Managed object reference of the datastore where the ISO file is stored. This property is available only when *CD/DVD Drive Type = "Datastore ISO File"* |
| **Connect at Power On** | Specifies if CD/DVD Drive should be connected when the VM powers on. |

## Optional Properties (NIC)

You can use the following properties, as necessary, when adding a **NIC** to the VM.

| | |
|---|---|
| **Connect at Power On** | Specifies if NIC should be connected when the VM powers on. |

# Create VM

The **Create VM** activity can be used in a runbook to configure and create a new VM.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **Datastore (MOR)** | Managed object reference of the datastore where the new VM will be stored. The specified datastore must be accessible from the Compute Resource (or Cluster Compute Resource) owner of the Resource Pool specified by the *Resource Pool (MOR)* property. To obtain the Datastore MOR you can use the Get vSphere Object activity with *Object Type = "Datastore".* |
| **Disk Provisioning** | Disk provisioning for the new VM disk. |
| **Disk Size (MB)** | Amount of disk space in MB assigned to the new VM. |
| **Folder (MOR)** | Managed object reference of the folder where the new VM should be created. To obtain the Folder MOR you can use the Get vSphere Object activity with *Object Type* = "Folder". |
| **Guest OS ID** | Guest operating system identifier for the new VM. This should be one of the guest operating systems available for the Compute Resource (or Cluster Compute Resource) owner of the Resource Pool specified by the *Resource Pool (MOR)* property. To obtain the Guest OS ID you can use the Get vSphere Object activity with *Object Type* = "Guest OS". |
| **Memory (MB)** | Amount of memory in MB assigned to the new VM. |
| **NIC Type** | Network interface controller card type to be used in the new VM. This should be one of the NICs available for the Compute Resource (or Cluster Compute Resource) owner of the Resource Pool specified by the *Resource Pool (MOR)* property and it should be supported by the operating system specified by *Guest OS ID*.<br>To obtain the NIC Type you can use the Get vSphere Object activity with *Object Type* = "NIC Option".<br>For more details on choosing a NIC for your VM see VMware KB1001805. |
| **Network (MOR)** | Managed object reference of the network or distributed virtual port group the new VM should be connected to. The network should be accessible from the Compute Resource (or Cluster Compute Resource) owner of the Resource Pool specified by the *Resource Pool (MOR)* property.<br>To obtain the NIC Type you can use the Get vSphere Object activity with *Object Type* = "Network".<br>For more details on choosing a NIC for your VM see VMware KB1001805. |

| | |
|---|---|
| **Number of Cores per Socket** | Number of CPU cores per socket for the new VM. |
| **Resource Pool (MOR)** | The resource pool to which the virtual machine will be attached. To obtain the Resource Pool MOR you can use the Get vSphere Object activity with *Object Type* = “Resource Pool”.<br>Specifying the resource pool also indirectly specifies the Compute Resource for the new VM, and thus, the Host associated with the Compute Resource in the case of a stand-alone host or, the Host Cluster when the compute resource is a Cluster Compute Resource.<br>For more details refer to vSphere Inventory Structure. |
| **VM Name** | Display name of the virtual machine. ***Note:*** *As a best practice it is recommended to avoid spaces and special characters such as %, &, *, $, #, @, !, \, /, :, *, ?, ", <, >, \|, ;, ', (, ), etc.*<br>For more details see VMware KB2046088. |
| **Total Number of Cores** | Total number of virtual CPU cores, across all sockets, for the new VM. |

## Optional Properties

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Host (MOR)** | Managed object of the host where the VM should be created.<br>To obtain the Host MOR you can use the Get vSphere Object activity with *Object Type* = “Host”.<br>This property is optional when creating a new VM under a stand-alone host since the host is indirectly specified by the *Resource Pool (MOR)* property. If specified in this case, the host must belong to the Compute Resource owner of the Resource Pool specified by the *Resource Pool (MOR)* property.<br>This property is optional when creating a new VM under a DRS enabled Cluster, since the host is assigned automatically as one of the hosts belonging to the Cluster Compute Resource indirectly specified by the *Resource Pool (MOR)* property. If specified in this case, the host must belong to the Cluster Compute Resource owner of the Resource Pool specified by the *Resource Pool (MOR)* property.<br>This property is required when creating a VM under to a non-DRS Cluster. The specified host must belong to the Cluster Compute Resource owner of the Resource Pool specified by the *Resource Pool (MOR)* property. The datastore, guest OS, NIC Type and network specified by the *Datastore (MOR), Guest OS ID, NIC Type and Network (MOR)* properties, respectively, must be accessible from the host. |

## *Published Data*

The activity publishes the following activity-specific data items.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the newly created VM.<br>To obtain more properties for this VM you can use the Get vSphere Object activity with *Object Type* = “VM”.<br>To obtain the VM datacenter or more information about the VM parent folder you can use the Get vSphere Object activity with *Object Type* = “Folder”. |

# Create VM From Template

The **Create VM From Template** activity can be used in a runbook to create a new VM based on an existing VM template.

## _Required Properties_

You must configure the following properties:

| | |
|---|---|
| **Datastore (MOR)** | Managed object reference of the datastore where the new VM will be stored. The specified datastore must be accessible from the Compute Resource owner of the Resource Pool specified by the _Resource Pool (MOR)_ property. To obtain the Datastore MOR you can use the Get vSphere Object activity with _Object Type_ = "Datastore". |
| **Folder (MOR)** | Managed object reference of the folder where the new VM should be created. To obtain the Folder MOR you can use the Get vSphere Object activity with _Object Type_ = "Folder". |
| **Resource Pool (MOR)** | The resource pool to which the virtual machine will be attached. To obtain the Resource Pool MOR you can use the Get vSphere Object activity with _Object Type_ = "Resource Pool".<br>Specifying the resource pool also indirectly specifies the Compute Resource for the new VM, and thus, the Host associated with the Compute Resource in the case of a stand-alone host or, the Host Cluster when the compute resource is a Cluster Compute Resource.<br>For more details refer to vSphere Inventory Structure. |
| **VM Name** | Display name of the virtual machine. **_Note:_** _As a best practice it is recommended to avoid spaces and special characters such as %, &, *, $, #, @, !, \, /, :, *, ?, ", <, >, \|, ;, ', (, ), etc._<br>For more details see VMware KB2046088. |
| **VM Template (MOR)** | The source template VM on which the new VM will be based. |

## _Optional Properties_

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Host (MOR)** | Managed object of the host where the VM should be created.<br>To obtain the Resource Pool MOR you can use the Get vSphere Object activity with _Object Type_ = "Host". |

| | |
|---|---|
| | This property is **optional** when creating a new VM under a stand-alone host since the host is indirectly specified by the *Resource Pool (MOR)* property. If specified in this case, the host must belong to the Compute Resource owner of the Resource Pool specified by the *Resource Pool (MOR)* property.<br>This property is **optional** when creating a new VM under a DRS enabled Cluster, since the host is assigned automatically as one of the hosts belonging to the Cluster Compute Resource indirectly specified by the *Resource Pool (MOR)* property. If specified in this case, the host must belong to the Cluster Compute Resource owner of the Resource Pool specified by the *Resource Pool (MOR)* property.<br>This property is **required** when creating a VM under to a non-DRS Cluster. The specified host must belong to the Cluster Compute Resource owner of the Resource Pool specified by the *Resource Pool (MOR)* property. The datastore, specified by the *Datastore (MOR)* property, must be accessible from the host. |
| **Customization Spec Name** | The name of a Customization Spec which is to be applied to the new VM. You can use the vSphere *Customization Specifications Manager* (under Home/Management) to configure VM customization specs.<br>***Note:*** VMware Tools must be installed on the template source VM in order for a customization spec to be applied.<br>***Note:*** A customization spec is applied on a new VM after the VM is started the first time. |
| **Disk Provisioning** | Specifies disk provisioning for the VM disk(s). If *Disk Drive Key* is not specified, the specified *Disk Provisioning* will be applied to all VM disks. |
| **Disk Drive Key** | Identifies the VM disk(s) for which *Disk Provisioning* is to be applied. If not specified, *Disk Provisioning* will be applied to all VM disks. To specify multiple disks, use a comma (,) separated list of drive keys.<br>To obtain the device key you can use the Get vSphere Object activity with *Object Type = "VM Device"*. When not specified, the activity updates the first disk drive device it finds, if any. |

## *Published Data*

The activity publishes the following activity-specific data items.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the newly created VM.<br>To obtain more properties for this VM you can use the Get vSphere Object activity with *Object Type* = "VM".<br>To obtain the VM datacenter or more information about the VM parent folder you can use the Get vSphere Object activity with *Object Type* = "Folder". |

# Delete VM

The **Delete VM** activity can be used in a runbook to permanently delete a VM from datastore or to remove a VM from inventory.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **VM (MOR)** | Identifies the VM to be deleted. |
| **Delete Type** | Specifies whether the VM will be permanently removed from disk or if it will be un-registered from inventory. Un-registered VMs can be later be re-registered and become part of inventory again. For details, please consult VMware documentation. ***Note:*** A re-registered VM will have a different Managed Object Reference (MOR) than before it was un-registered. |

## *Optional Properties*

You can use the following optional properties.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |

## *Published Data*

This activity does not publish any activity-specific data items.

# Get Custom Attribute

The **Get Custom Attribute** activity can be used in a runbook to retrieve custom attributes that have been set for a VM or host.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **VM or Host (MOR)** | Managed object reference of the VM or host for which the custom attributes are to be retrieved. |

## *Optional Properties*

You can use the following optional properties.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |

## *Filters*

This activity provides the following filters that you can combine to selectively filter which attributes to retrieve.

| | |
|---|---|
| **Attribute Key** | Filter by Attribute Key. |
| **Attribute Name** | Filter by Attribute Name. |
| **Attribute Value** | Filter by Attribute Value. |
| **Object Type** | Filter by Object Type |

## *Published Data*

This activity publishes the following activity-specific data items.

| | |
|---|---|
| **Attribute Key** | Specifies the unique attribute key. |
| **Attribute Name** | Specifies the attribute name. |
| **Attribute Value** | Specifies the attribute value. |
| **Count** | The number of custom attribute records returned by the activity. |
| **Object Type** | Specifies the vSphere object type associated with this attribute. This value is null for global custom attributes. |

# Get vSphere Object

The **Get vSphere Object** activity can used in a runbook to retrieve information about various objects in vSphere. The **Object Type** property specifies the type of object retrieved by the activity.

## *Required Properties*

You must configure the **Object Type** property with the type of object you want to retrieve. After you make your selection additional properties, filters and published data will be made available.

| | |
|---|---|
| **Object Type** | Specifies the object type to be retrieved. Options include:<br>• Cluster Compute Resource<br>• Compute Resource<br>• Datacenter<br>• Datastore<br>• Folder<br>• Guest OS<br>• Host<br>• Network<br>• NIC Option<br>• Resource Pool<br>• VM<br>• VM Device<br>• VM Guest Info<br>• VM Status |

## *Required Properties (Guest OS)*

You must configure the following properties when retrieving **Guest OS** objects.

| | |
|---|---|
| **Compute Resource (MOR)** | Specifies the compute resource (or cluster compute resource) for which available guest operating systems should be retrieved. When specifying a cluster compute resource, the returned list is an aggregate list for all the hosts under the cluster compute resource. |

## *Required Properties (NIC)*

You must configure the following properties when retrieving **NIC** objects.

| | |
|---|---|
| **Compute Resource (MOR)** | Managed object reference of a compute resource (or cluster compute resource) for which NIC options should be retrieved. When specifying a cluster compute resource, the returned list is an aggregate list for all the hosts under the cluster compute resource. |
| **Guest OS ID** | Guest operating system for which NIC options should be retrieved. |

## *Required Properties (VM Device)*

You must configure the following properties when retrieving **VM Device** objects.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the virtual machine for which devices should be retrieved. |

## *Required Properties (VM Guest Info)*

You must configure the following properties when retrieving **VM Guest Info** objects.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the virtual machine for which guest information should be retrieved. |

## *Required Properties (VM Status)*

You must configure the following properties when retrieving **VM Status** objects.

| | |
|---|---|
| **VM (MOR)** | Identifies the VM for which status information is to be retrieved. |

## *Optional Properties*

You can use the following optional properties.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |

## *Optional Properties (Cluster Compute Resource)*

You can use the following properties, as necessary, when retrieving **Cluster Compute Resource** objects.

| | |
|---|---|
| **Cluster Compute Resource (MOR)** | Managed object reference for a specific cluster compute resource to be retrieved. |

## *Optional Properties (Compute Resource)*

You can use the following properties, as necessary, when retrieving **Compute Resource** objects.

| | |
|---|---|
| **Compute Resource (MOR)** | Managed object reference for a specific compute resource to be retrieved. |

## *Optional Properties (Datacenter)*

You can use the following properties, as necessary, when retrieving **Datacenter** objects.

| | |
|---|---|
| **Datacenter (MOR)** | Managed object reference for a specific datacenter to be retrieved. |

## *Optional Properties (Datastore)*

You can use the following properties, as necessary, when retrieving **Datastore** objects.

| | |
|---|---|
| **Compute Resource (MOR)** | Managed object reference of a Compute Resource (or Cluster Compute Resource) for which accessible datastores should be retrieved. When |

| | |
|---|---|
| | specifying a cluster compute resource, the returned list is an aggregate list for all the hosts under the cluster compute resource. |
| **Datastore (MOR)** | Managed object reference for a specific Datastore to be retrieved. |

## *Optional Properties (Host)*

You can use the following properties, as necessary, when retrieving **Host** objects.

| | |
|---|---|
| **Host (MOR)** | Managed object reference of a specific host to be retrieved. |

## *Optional Properties (Network)*

You can use the following properties, as necessary, when retrieving **Network** objects.

| | |
|---|---|
| **Compute Resource (MOR)** | Managed object reference of a compute resource (or cluster compute resource) for which available networks should be retrieved. When specifying a cluster compute resource, the returned list is an aggregate list for all the hosts under the cluster compute resource. |
| **Network (MOR)** | Managed object reference of a specific network to be retrieved. |

## *Optional Properties (Resource Pool)*

You can use the following properties, as necessary, when retrieving **Resource Pool** objects.

| | |
|---|---|
| **Resource Pool (MOR)** | Managed object reference of a specific resource pool to be retrieved. |

## *Optional Properties (VM)*

You can use the following properties, as necessary, when retrieving **VM** objects.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of a specific virtual machine to be retrieved. |

## *Filters (Cluster Compute Resource)*

This activity provides the following filters when retrieving **Cluster Compute Resource** objects.

| | |
|---|---|
| **Datastores (MOR)** | Filter by a datastore (MOR) that is (or is not) part of Datastores (MOR) list. |
| **Effective CPU (MHz)** | Filter by Effective CPU (MHz). |
| **Effective Memory (MB)** | Filter by Effective Memory (MB). |
| **Hosts (MOR)** | Filter by a Host (MOR) that is (or is not) part of Hosts (MOR) list. |
| **Name** | Filter by Name. |
| **Networks (MOR)** | Filter by a network (MOR) that is (or is not) part of Networks (MOR) list. |
| **Number of CPU Cores** | Filter by Number of CPU Cores. |
| **Number of CPU Threads** | Filter by Number of CPU Threads. |

| | |
|---|---|
| **Number of Effective Hosts** | Filter by Number of Effective Hosts. |
| **Number of Hosts** | Filter by Number of Hosts. |
| **Parent (MOR)** | Filter by Parent (MOR). |
| **Resource Pool (MOR)** | Filter by Resource Pool (MOR). |
| **Total CPU (MHz)** | Filter by Total CPU (MHz). |
| **Total Memory (bytes)** | Filter by Total Memory (bytes). |

## *Filters (Compute Resource)*

This activity provides the following filters when retrieving **Compute Resource** objects.

| | |
|---|---|
| **Datastores (MOR)** | Filter by a datastore (MOR) that is (or is not) part of Datastores (MOR) list. |
| **Effective CPU (MHz)** | Filter by Effective CPU (MHz). |
| **Effective Memory (MB)** | Filter by Effective Memory (MB). |
| **Hosts (MOR)** | Filter by a Host (MOR) that is (or is not) part of Hosts (MOR) list. |
| **Name** | Filter by Name. |
| **Networks (MOR)** | Filter by a network (MOR) that is (or is not) part of Networks (MOR) list. |
| **Number of CPU Cores** | Filter by Number of CPU Cores. |
| **Number of CPU Threads** | Filter by Number of CPU Threads. |
| **Number of Effective Hosts** | Filter by Number of Effective Hosts. |
| **Number of Hosts** | Filter by Number of Hosts. |
| **Parent (MOR)** | Filter by Parent (MOR). |
| **Resource Pool (MOR)** | Filter by Resource Pool (MOR). |
| **Total CPU (MHz)** | Filter by Total CPU (MHz). |
| **Total Memory (bytes)** | Filter by Total Memory (bytes). |

## *Filters (Datacenter)*

This activity provides the following filters when retrieving **Datacenter** objects.

| | |
|---|---|
| **Datastore Folder (MOR)** | Filter by Datastore Folder (MOR). |
| **Host Folder (MOR)** | Filter by Host Folder (MOR). |
| **Name** | Filter by Name. |

| | |
|---|---|
| **Network Folder (MOR)** | Filter by Network Folder (MOR). |
| **Parent (MOR)** | Filter by Parent (MOR). |
| **VM Folder (MOR)** | Filter by VM Folder (MOR). |

## *Filters (Datastore)*

This activity provides the following filters when retrieving **Datastore** objects.

| | |
|---|---|
| **Accessible** | Filter by Accessible value. |
| **Maintenance Mode** | Filter by Maintenance Mode. Valid values are:<br>• enteringMaintenance<br>• inMaintenance<br>• normal |
| **Multiple Host Access** | Filter by Multiple Host Access value. |
| **Name** | Filter by Name. |
| **Parent (MOR)** | Filter by Parent (MOR). |
| **Type** | Filter by Type. Valid values are:<br>• CIFS<br>• NFS<br>• NFS41<br>• OTHER<br>• VFFS<br>• VMFS<br>• vsan<br>• VVOL |

## *Filters (Folder)*

This activity provides the following filters when retrieving **Folder** objects.

| | |
|---|---|
| **Child Types** | Filter by Child Types. |
| **Datacenter** | Filter by Datacenter. |
| **Datacenter (MOR)** | Filter by Datacenter (MOR). |
| **Folder (MOR)** | Filter by MOR. |
| **Name** | Filter by Name. |
| **Parent (MOR)** | Filter by Parent (MOR). |
| **Path** | Filter by Path. |

## Filters (Guest OS)

This activity provides the following filters when retrieving **Guest OS** objects.

| | |
|---|---|
| **Family** | Filter by Family. |
| **Full Name** | Filter by Full Name. |
| **ID** | Filter by ID. |
| **Recommended NIC Card** | Filter by Recommended NIC Card value. |
| **Support Level** | Filter by Support Level value. |

## Filters (Host)

This activity provides the following filters when retrieving **Host** objects.

| | |
|---|---|
| **Connection State** | Filter by Connection State |
| **CPU (MHz)** | Filter by CPU (MHz) |
| **In Maintenance Mode** | Filter by In Maintenance Mode value |
| **Memory Size (bytes)** | Filter by Memory Size (bytes) |
| **Name** | Filter by Name |
| **Number of CPU Cores** | Filter by Number of CPU Cores |
| **Number of CPU Threads** | Filter by Number of CPU Threads |
| **Number of CPUs** | Filter by Number of CPUs |
| **Number of NICs** | Filter by Number of NICs |
| **Overall CPU Usage** | Filter by Overall CPU Usage |
| **Overall Memory Usage** | Filter by Overall Memory Usage |
| **Parent MOR** | Filter by Parent MOR |
| **Power State** | Filter by Power State |
| **Uptime** | Filter by Uptime |
| **UUID** | Filter by UUID |

## Filters (Network)

This activity provides the following filters when retrieving **Network** objects.

| | |
|---|---|
| **Accessible** | Filter by Accessible |
| **IP Pool ID** | Filter by IP Pool ID |
| **IP Pool Name** | Filter by IP Pool Name |
| **Name** | Filter by Name |

| | |
|---|---|
| **Parent (MOR)** | Filter by Parent (MOR) |

## *Filters (NIC)*

This activity provides the following filters when retrieving **NIC** objects.

| | |
|---|---|
| **Connectable** | Filter by Connectable |
| **Deprecated** | Filter by Deprecated |
| **MAC Option Default** | Filter by MAC Option Default |
| **MAC Options** | Filter by MAC Options |
| **Recommended** | Filter by Recommended |
| **Type** | Filter by Type |
| **Wake on LAN Supported** | Filter by Wake on LAN Supported |

## *Filters (Resource Pool)*

This activity provides the following filters when retrieving **Resource Pool** objects.

| | |
|---|---|
| **Name** | Filter by Name |
| **Owner (MOR)** | Filter by Owner (MOR) |
| **Parent (MOR)** | Filter by Parent (MOR) |

## *Filters (VM)*

This activity provides the following filters when retrieving **VM** objects.

| | |
|---|---|
| **Guest OS ID** | Filter by Guest OS ID |
| **Instance UUID** | Filter by Instance UUID |
| **Is Template** | Filter by Is Template |
| **Name** | Filter by Name |
| **Parent (MOR)** | Filter by Parent (MOR) |
| **Parent VApp (MOR)** | Filter by Parent VApp (MOR) |
| **Resource Pool (MOR)** | Filter by Resource Pool (MOR) |
| **UUID** | Filter by UUID |
| **Version** | Filter by Version |

## *Filters (VM Device)*

This activity provides the following filters when retrieving **VM Device** objects.

| | |
|---|---|
| **Allow Connect Guest Control** | Filter by Allow Connect Guest Control |
| **Connect Status** | Filter by Connect Status |
| **Connected** | Filter by Connected |
| **Controller Key** | Filter by Controller Key |
| **Key** | Filter by Key |
| **Label** | Filter by Label |
| **Start Connected** | Filter by Start Connected |
| **Summary** | Filter by Summary |
| **Type** | Filter by Type |
| **Unit Number** | Filter by Unit Number |

## *Published Data (Cluster Compute Resource)*

This activity publishes the following activity-specific data items when retrieving **Cluster Compute Resource** objects.

| | |
|---|---|
| **Compute Resource (MOR)** | Managed object reference for the cluster compute resource. |
| **Count** | Number of items retrieved by the activity. |
| **Datastores (MOR)** | Datastores available for this compute resource. This value is a comma separated list of managed object references. |
| **Effective CPU (MHz)** | Effective CPU resources available to run VMs. This is the aggregated effective resource level from all running hosts. Hosts that are in maintenance mode or are unresponsive are not counted. Resources used by the VMware Service Console are not included in the aggregate. This value represents the number of resources available for the root resource pool for running virtual machines. |
| **Effective Memory (MB)** | Effective memory resources (in MB) available to run virtual machines. This is the aggregated effective resource level from all running hosts. Hosts that are in maintenance mode or are unresponsive are not counted. Resources used by the VMware Service Console are not included in the aggregate. This value represents the number of resources available for the root resource pool for running virtual machines. |

| | |
|---|---|
| **Hosts (MOR)** | Hosts that are part of this compute resource. This value is a comma separated list of managed object references. |
| **Name** | Compute resource name. |
| **Networks (MOR)** | Networks available for this compute resource. This value is a comma separated list of managed object references. |
| **Number of CPU Cores** | Number of physical CPU cores. |
| **Number of CPU Threads** | Aggregated number of CPU threads. |
| **Number of Effective Hosts** | Total number of effective hosts. |
| **Number of Hosts** | Total number of hosts. |
| **Parent (MOR)** | Managed object reference of the parent folder for this compute resource. |
| **Resource Pool (MOR)** | Managed object reference of the root resource pool for this compute resource. |
| **Total CPU (MHz)** | Aggregated CPU resources of all hosts, in MHz |
| **Total Memory (bytes)** | Aggregated memory resources of all hosts, in bytes. |

## *Published Data (Compute Resource)*

This activity publishes the following activity-specific data items when retrieving **Compute Resource** objects.

| | |
|---|---|
| **Compute Resource (MOR)** | Managed object reference for the compute resource. |
| **Count** | Number of items retrieved by the activity. |
| **Datastores (MOR)** | Datastores available for this compute resource. This value is a comma separated list of managed object references. |
| **Effective CPU (MHz)** | Effective CPU resources available to run VMs. This is the aggregated effective resource level from all running hosts. Hosts that are in maintenance mode or are unresponsive are not counted. Resources used by the VMware Service Console are not included in the aggregate. This value represents the number of resources available for the root resource pool for running virtual machines. |
| **Effective Memory (MB)** | Effective memory resources (in MB) available to run virtual machines. This is the aggregated effective resource level from all running hosts. Hosts that are in maintenance mode or are unresponsive are not counted. Resources used by the VMware Service Console are not included in the aggregate. This value represents the number of |

| | |
|---|---|
| | resources available for the root resource pool for running virtual machines. |
| **Hosts (MOR)** | Hosts that are part of this compute resource. This value is a comma separated list of managed object references. |
| **Name** | Compute resource name. |
| **Networks (MOR)** | Networks available for this compute resource. This value is a comma separated list of managed object references. |
| **Number of CPU Cores** | Number of physical CPU cores. |
| **Number of CPU Threads** | Aggregated number of CPU threads. |
| **Number of Effective Hosts** | Total number of effective hosts. |
| **Number of Hosts** | Total number of hosts. |
| **Parent (MOR)** | Managed object reference of the parent folder for this compute resource. |
| **Resource Pool (MOR)** | Managed object reference of the root resource pool for this compute resource. |
| **Total CPU (MHz)** | Aggregated CPU resources of all hosts, in MHz |
| **Total Memory (bytes)** | Aggregated memory resources of all hosts, in bytes. |

### *Published Data (Datacenter)*

This activity publishes the following activity-specific data items when retrieving **Datacenter** objects.

| | |
|---|---|
| **Count** | Number of items retrieved by the activity. |
| **Datacenter (MOR)** | Managed object reference for this datacenter. |
| **Datastore Folder (MOR)** | Managed object reference of the main folder containing datastores for this datacenter. |
| **Datastores (MOR)** | Comma separated list of managed object references of datastores available in this datacenter. |
| **Host Folder (MOR)** | Managed object reference of the main folder containing hosts for this datacenter. |
| **Name** | Datacenter name. |
| **Network Folder (MOR)** | Managed object reference of the main folder containing networks for this datacenter. |
| **Networks (MOR)** | Comma separated list of managed object references of networks available in this datacenter. |
| **Parent (MOR)** | Managed object reference of the parent folder for this datacenter. |

| | |
|---|---|
| **VM Folder (MOR)** | Managed object reference of the main folder containing VMs for this datacenter. |

## Published Data (Datastore)

This activity publishes the following activity-specific data items when retrieving **Datastore** objects.

| | |
|---|---|
| **Accessible** | The connectivity status of this datastore. If this is False, some of the other published properties cannot be guaranteed to be accurate. For datastores accessed from multiple hosts, vCenter Server reports Accessible as an aggregated value. |
| **Capacity (bytes)** | Maximum capacity of the datastore, in bytes. This property is guaranteed to be valid only if Accessible is True. |
| **Count** | Number of items retrieved by the activity. |
| **Datastore (MOR)** | Managed object reference of the datastore. |
| **Free Space (bytes)** | Available space of this datastore, in bytes. This property is guaranteed to be valid only if Accessible is True. |
| **Hosts (MOR)** | Comma separated list of managed object references of hosts attached to this datastore. |
| **Maintenance Mode** | The current maintenance mode state of the datastore. |
| **Multiple Host Access** | Specifies if more than one host in the datacenter has been configured with access to the datastore. |
| **Name** | Datastore name. |
| **Parent (MOR)** | Managed object reference of the datastore parent folder. |
| **Type** | Type of file system volume. |
| **Uncommitted (bytes)** | Total additional storage space, in bytes, potentially used by all virtual machines on this datastore. This property is guaranteed to be valid only if Accessible is True. |

## Published Data (Folder)

This activity publishes the following activity-specific data items when retrieving **Folder** objects.

| | |
|---|---|
| **Child Types** | Comma separated list of child types that can exist under this folder. |
| **Count** | Number of items retrieved by the activity. |
| **Datacenter** | Name of datacenter under which this folder is located. This value is empty for datacenter folders. |
| **Datacenter (MOR)** | Managed object reference of datacenter under which this folder is located. This value is empty for datacenter folders. |

| | |
|---|---|
| **Folder (MOR)** | Managed object reference of this folder. |
| **Name** | Folder name. |
| **Parent (MOR)** | Managed object reference of parent folder. |
| **Path** | Folder path. For non-datacenter folders, the path includes the datacenter path under which the folder is located. |

## *Published Data (Guest OS)*

This activity publishes the following activity-specific data items when retrieving **Guest OS** objects.

| | |
|---|---|
| **Count** | Number of items retrieved by the activity. |
| **Family** | Family to which this guest operating system belongs. |
| **ID** | Identifier for the guest operating system. |
| **Recommended NIC Card** | Recommended default NIC card type for this guest. |
| **Support Level** | Support level for this guest operating system.<br>• **deprecated** - Support for this operating system will be terminated in the future. Please migrate to using a different operating system.<br>• **experimental** - This operating system is not supported but may be supported in the future.<br>• **legacy** - This operating system is not fully supported but may have been supported in the past.<br>• **supported** - Fully supported.<br>• **techPreview** - This operating system may not be supported yet, please check VMware compatibility guide.<br>• **terminated** - No longer supported.<br>• **unsupported** |

## *Published Data (Host)*

This activity publishes the following activity-specific data items when retrieving **Host** objects.

| | |
|---|---|
| **Connection State** | Host connection state. |
| **Count** | Number of items retrieved by the activity. |
| **CPU (MHz)** | The speed of the CPU cores. This is an average value if there are multiple speeds. |
| **Datastores (MOR)** | Comma separated list of managed object references of the datastores available on this host. |
| **Host (MOR)** | Managed object reference of the host. |

| | |
|---|---|
| **In Maintenance Mode** | Indicates whether the host is in maintenance mode or not. |
| **Management Server IP Address** | IP address of the vCenter server managing this host, if any. |
| **Memory Size (bytes)** | The physical memory size on the host, in bytes. |
| **Name** | Host name. |
| **Networks (MOR)** | Comma separated list of managed object references of the networks available on this host. |
| **Number of CPU Cores** | Number of physical CPU cores on the host. |
| **Number of CPU Threads** | Number of physical CPU threads on the host. |
| **Number of CPUs** | Number of physical CPU packages on the host. |
| **Number of NICs** | The number of network adapters. |
| **Overall CPU Usage** | Aggregated CPU usage across all cores on the host in MHz's This is only available if the host is connected. |
| **Overall Memory Usage** | Physical memory usage on the host in MB. This is only available if the host is connected. |
| **Parent MOR** | Management object reference of the host parent folder. |
| **Power State** | The host power state. |
| **Uptime** | The system uptime of the host in seconds. |
| **UUID** | The hardware BIOS identification. |

### *Published Data (Network)*

This activity publishes the following activity-specific data items when retrieving **Network** objects.

| | |
|---|---|
| **Accessible** | Indicates if at least one host is configured to provide this network. |
| **Count** | Number of items retrieved by the activity. |
| **Hosts (MOR)** | Comma separated list of managed object references of hosts attached to the network. |
| **IP Pool ID** | Identifier of the associated IP pool. Zero if the network is not associated with an IP pool. |
| **IP Pool Name** | Name of the associated IP pool. Empty if the network is not associated with an IP pool. |
| **Network (MOR)** | Managed object reference of the network. |
| **Name** | Network name. |

| | |
|---|---|
| **Parent (MOR)** | Managed object reference of the network parent folder. |

## Published Data (NIC)

This activity publishes the following activity-specific data items when retrieving **NIC** objects.

| | |
|---|---|
| **Connectable** | Indicates if this NIC option is connectable. |
| **Count** | Number of items returned by the activity. |
| **Deprecated** | Indicates if this NIC option is deprecated. |
| **MAC Option Default** | Default MAC address type. |
| **MAC Options** | Comma separated list of MAC address types. |
| **Recommended** | Indicates if this is the recommended NIC option. |
| **Type** | The type of the NIC. |
| **Wake on LAN Supported** | Indicates if Wake-on-LAN is supported for this NIC option. |

## Published Data (Resource Pool)

This activity publishes the following activity-specific data items when retrieving **Resource Pool** objects.

| | |
|---|---|
| **Count** | Number of items retrieved by the activity. |
| **Name** | Resource pool name. |
| **Owner (MOR)** | Managed object reference of the resource pool owner. |
| **Owner Type** | Resource pool owner type. Can be:<br>• Compute Resource<br>• Cluster Compute Resource |
| **Parent (MOR)** | Managed object reference of the resource pool owner. |
| **Parent Type** | Resource pool parent type. Can be:<br>• Resource Pool<br>• Compute Resource<br>• Cluster Compute Resource |
| **Resource Pool (MOR)** | Managed object reference of the resource pool. |

## Published Data (VM)

This activity publishes the following activity-specific data items when retrieving **VM** objects.

| | |
|---|---|
| **Count** | Number of items retrieved by the activity. |

| | |
|---|---|
| **Datastores (MOR)** | Comma separated list of managed object references of the datastores used by this VM. |
| **Guest OS** | Name of guest OS used by this VM. |
| **Guest OS ID** | ID of the guest OS used by this VM. |
| **Instance UUID** | vCenter-specific 128-bit UUID of a virtual machine, represented as a hexadecimal string. This identifier is used by vCenter to uniquely identify all virtual machine instances, including those that may share the same SMBIOS UUID. |
| **Is Template** | Indicates if this is a VM template. |
| **Memory (MB)** | Amount of memory in MB assigned to this VM. |
| **Name** | VM display name. |
| **Networks (MOR)** | Comma separated list of managed object references of the networks used by this VM. |
| **Number of CPUs** | Number of CPUs on the VM. |
| **Number of NICs** | Number of NIC cards on the VM. |
| **Parent (MOR)** | Managed object reference of the VM parent folder. |
| **Parent VApp (MOR)** | Managed object reference of the VM parent VApp. |
| **Resource Pool (MOR)** | Managed object reference of the resource pool associated with this VM. |
| **UUID** | Virtual machine BIOS identification. |
| **Version** | VM hardware virtualization version. |
| **VM (MOR)** | Managed object reference of the VM. |
| **VMX Path** | Path of the VMX configuration file of the VM. |

## *Published Data (VM Device)*

This activity publishes the following activity-specific data items when retrieving **VM Device** objects.

| | |
|---|---|
| **Allow Connect Guest Control** | Indicates whether the guest OS controls when the device is connected. |
| **Connect Status** | Device connect status.<br>• **ok** - The device is working correctly.<br>• **recoverableError** - The device has reported a recoverable error. For example, attempting to connect to floppy device that is being used by another virtual machine or some other program would result in this status.<br>• **unrecoverableError** - The device cannot be used. For example, attempting to connect to a floppy device that does not exist would result in this status. |

| | |
|---|---|
| | - **untried** - The device status is unknown, or it has not been requested to connect when the VM is powered on. |
| **Connected** | Indicates whether the device is currently connected. Valid only while the virtual machine is running. |
| **Controller Key** | Key of the controller device for this device. |
| **Count** | Number of items retrieved by the activity. |
| **Key** | A unique key that distinguishes this device from other devices in the same virtual machine. Keys are immutable but may be recycled; that is, a key does not change as long as the device is associated with a particular virtual machine. However, once a device is removed, its key may be used when another device is added. |
| **Label** | Device display label. |
| **Start Connected** | Specifies whether or not to connect the device when the virtual machine starts. |
| **Summary** | Device summary description. |
| **Type** | Device type. |
| **Unit Number** | The unit number of this device on its controller. This property is null if the device does not have a controller.<br>Normally, two devices on the same controller may not be assigned the same unit number. If multiple devices could exist on a controller, then unit number has to be specified to configure respective devices. |

## *Published Data (VM Guest Info)*

This activity publishes the following activity-specific data items when retrieving **VM Guest Info** objects.

| | |
|---|---|
| **App Heartbeat Status** | Application heartbeat status. Possible values are:<br>- **appStatusGray** – Heartbeat status is disabled.<br>- **appStatusGreen** – Heartbeat status is OK.<br>- **appStatusRed** – Heartbeat has stopped. |
| **Guest Family** | Guest OS family, if known. |
| **Guest Full Name** | Guest OS full name, if known. |
| **Guest ID** | Guest OS identifier, if known. |
| **Guest Kernel Crashed** | Indicates if guest OS kernel has crashed. |
| **Guest Operations Ready** | Indicates guest operations availability. When True, the VM is ready to process guest operations. |
| **Guest State** | Operation mode of the guest operating system. Possible values are:<br>- **running** – Guest OS is running normally. |

| | |
|---|---|
| | • **shuttingdown** – Guest OS has a pending shutdown command.<br>• **resetting** – Guest OS has a pending reset command.<br>• **standby** – Guest OS has a pending standby command.<br>• **notrunning** – Guest OS is not running.<br>• **unknown** - Guest OS information is not available. |
| **Guest State Change Supported** | Indicates if the VM is ready to process soft power operations. |
| **Host Name** | Host name of the guest OS, if known. |
| **Interactive Guest Operations Ready** | Indicates if the VM is ready to process guest operations as the user interacting with the guest desktop. |
| **IP Address** | Primary IP address assigned to the guest OS, if known. |
| **Tools Running Status** | Current running status of VMware Tools in the guest operating system, if known. Possible values are:<br>• **guestToolsRunning** – Tools running.<br>• **guestToolsNotRunning** – Tools not running.<br>• **guestToolsExecutingScripts** – Tools starting. |
| **Tools Version** | Current versions of VMware Tools, if known. |
| **Tools Version Status** | • **guestToolsBlacklisted** - VMware Tools is installed, but the installed version is known to have a grave bug and should be immediately upgraded.<br>• **guestToolsCurrent** - VMware Tools is installed, and the version is current.<br>• **guestToolsNeedUpgrade** - VMware Tools is installed, but the version is not current.<br>• **guestToolsNotInstalled** - VMware Tools has never been installed.<br>• **guestToolsSupportedNew** - VMware Tools is installed, supported, and newer than the version available on the host.<br>• **guestToolsSupportedOld** - VMware Tools is installed, supported, but a newer version is available.<br>• **guestToolsTooNew** - VMware Tools is installed, and the version is known to be too new to work correctly with this virtual machine.<br>• **guestToolsTooOld** - VMware Tools is installed, but the version is too old.<br>• **guestToolsUnmanaged** - VMware Tools is installed, but it is not managed by VMware. |

## *Published Data (VM Status)*

This activity publishes the following activity-specific data items when retrieving **VM Status** objects.

| | |
|---|---|
| **VM Guest OS State** | Operation mode of the guest operating system. Possible values are:<br>• **running** – Guest OS is running normally.<br>• **shuttingdown** – Guest OS has a pending shutdown command. |

| | |
|---|---|
| | • **resetting** – Guest OS has a pending reset command.<br>• **standby** – Guest OS has a pending standby command.<br>• **notrunning** – Guest OS is not running.<br>• **unknown** – Guest OS information is not available. |
| **VM Power State** | Current power state of the VM. Possible values are:<br>• **poweredOff** – VM is currently powered off.<br>• **poweredOn** – VM is currently powered on.<br>• **suspended** – VM is currently suspended. |
| **VM Tools Running Status** | Current running status of VMware Tools in the guest operating system, if known. Possible values are:<br>• **guestToolsRunning** – Tools running.<br>• **guestToolsNotRunning** – Tools not running.<br>• **guestToolsExecutingScripts** – Tools starting. |

# Move VM

The **Move VM** activity can be used in a runbook to move an existing VM to another folder, resource pool, datastore, host or datacenter.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **VM (MOR)** | Managed object reference identifying the VM that is to be moved. |

## *Optional Properties*

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Datastore (MOR)** | Specifies the datastore where the VM will be moved.<br>If not specified, the current VM datastore will be used.<br>If not specified and the current VM datastore is not accessible, the activity will fail.<br>Must be specified when moving the VM to another datacenter.<br>Must be specified when moving the VM to another host that is not associated with the current VM datastore.<br>To obtain the Datastore MOR you can use the Get vSphere Object activity with *Object Type = "Datastore".* |
| **Resource Pool (MOR)** | Specifies the resource pool to which the VM will be attached.<br>Must be specified when moving the VM to another datacenter.<br>Must be specified when moving the VM to another host that is not associated with the current VM resource pool.<br>To obtain the Resource Pool MOR you can use the Get vSphere Object activity with *Object Type* = "Resource Pool".<br>Specifying the resource pool also indirectly specifies the Compute Resource for the VM, and thus, the Host associated with the Compute Resource in the case of a stand-alone host or, the Host Cluster when the compute resource is a Cluster Compute Resource.<br>For more details refer to vSphere Inventory Structure. |
| **Folder (MOR)** | Specifies the folder where the VM should be moved.<br>When not specified:<br>• If moving within the same datacenter, the VM will keep the same folder. |

| | |
|---|---|
| | • If moving to another datacenter, the root VM folder of the destination datacenter will be used.<br>To obtain the Folder MOR you can use the Get vSphere Object activity with *Object Type* = “Folder”. |
| **Host (MOR)** | Specifies the host where the VM should be moved.<br>If not specified:<br>• If the Resource Pool is not specified, the current VM host will be used.<br>• If the Resource Pool is specified, and it is associated with a stand-alone host, that host will be used.<br>• If the Resource Pool is specified, and it is associated with a DRS enabled cluster, a host selected by DRS will be used.<br>If specified, the host must be associated with the new VM Resource Pool, otherwise the activity will fail.<br>Typically, this property is specified when moving the VM to another host in a cluster, where multiple hosts are associated with the same resource pool. When moving the VM to a stand-alone host, that host is implicitly specified by specifying its associated Resource Pool, thus this property does not have to be specified.<br>To obtain the Host MOR you can use the Get vSphere Object activity with *Object Type* = “Host”. |
| **Disk Provisioning** | Specifies disk provisioning for the VM disk(s). If *Disk Drive Key* is not specified, the specified *Disk Provisioning* will be applied to all VM disks. |
| **Disk Drive Key** | Identifies the VM disk(s) for which *Disk Provisioning* is to be applied. If not specified, *Disk Provisioning* will be applied to all VM disks. To specify multiple disks, use a comma (,) separated list of drive keys.<br>To obtain the device key you can use the Get vSphere Object activity with *Object Type = “VM Device”*. When not specified, the activity updates the first disk drive device it finds, if any. |

## Published Data

This activity does not publish any activity-specific data items.

# Restart VM

The **Restart VM** activity can be used in a runbook to restart a VM.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being restarted. |
| **Reboot Guest OS** | Specifies whether the VM Guest OS will be rebooted (soft restart) or if the VM power will be reset (hard restart). ***Note:*** *The VM must have VMware Tools installed and running in order for the activity to reboot (soft restart) the VM.* |
| **Wait For** | Specifies a VM state that the activity will be waiting for before completing. Possible values are:<br>• **VM Powered On** – activity waits until the VM is in *poweredOn* state.<br>• **VM Guest OS running** – activity waits until VM Guest operating system is in *running* state.<br>• **VM Tools Running** – activity waits until VMware Tools are in *guestToolsRunning* running state.<br>• **Do Not Wait –** activity does not wait.<br>***Note:*** *When the activity is configured to wait for a state, the VM is first Shut Down or Powered Off (dictated by Reboot Guest OS) and then started again.* |

## *Optional Properties*

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Wait Timeout (seconds)** | Specifies the number of seconds before the activity times out, when *Wait For* is configured to wait for a VM state. ***Note:*** *If this property is not specified and* ***Wait For*** *is configured so that the activity waits, the activity will not time out.* |

# Run vSphere Script

The **Run vSphere Script** activity can be used to run a PowerCLI script in your vSphere environment. Before running your script, the activity first connects to the vSphere server specified in your vSphere configuration options.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **Script Configuration** | Specifies the activity script configuration. The activity will provide additional inputs and outputs based on the script configuration selection.<br>The available values for this property will list the script configurations defined in your JSON configuration file. You can select one of the custom configurations, or you can select <NONE> to run a script without custom parameters or outputs. |
| **Script** | Contains the PowerCLI script to be executed.<br>If you are using the JSON configuration file to define custom parameters, activity input runtime values will be available in the script as PowerShell parameters.<br>If you are using the JSON configuration file to define custom outputs, output script values will be published on the Orchestrator data bus. |

## *Optional Properties*

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |

## *Published Data*

This activity does not publish any activity-specific data items.

# Run vROps Script

The **Run vROps Scrip**t activity can be used to run a PowerCLI script in your vRealize Operation Manager environment. Before running your script, the activity first connects to the vSphere server specified in your vROps configuration options.

***Important:*** The Run vROps Script activity is only supported as a professional services engagement. For details on how to configure and use this activity please contact Kelverion Professional Services.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **Script Configuration** | Specifies the activity script configuration. The activity will provide additional inputs and outputs based on the script configuration selection.<br>The available values for this property will list the script configurations defined in your JSON configuration file. You can select one of the custom configurations, or you can select <NONE> to run a script without custom parameters or outputs. |
| **Script** | Contains the PowerCLI script to be executed.<br>If you are using the JSON configuration file to define custom parameters, activity input runtime values will be available in the script as PowerShell parameters.<br>If you are using the JSON configuration file to define custom outputs, output script values will be published on the Orchestrator data bus. |

## *Optional Properties*

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |

## *Published Data*

This activity does not publish any activity-specific data items.

# Set Custom Attribute

The **Set Custom Attribute** activity can be used in a runbook to set a custom attribute value for a VM or host. If the specified attribute does not exist, the IP will create it.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **VM or Host (MOR)** | Managed object reference of the VM or host for which the custom attribute value is to be set. |
| **Attribute Name** | Custom attribute name. By default, if the attribute does not exist, the IP will create it. You can use the *Don't Create Attribute* optional property to change this behavior. |
| **Attribute Value** | Custom attribute value to set. |

## *Optional Properties*

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Don't Create Attribute** | Specifies that the IP should not create the custom attribute, when it does not exist. **When this property is set to *True and* the specified custom attribute does not exist, the activity will return an error.** |
| **Is Global Attribute** | Specifies if the attribute that is to be set/created is a global attribute or not. Note that *VM or Host (MOR)* still has to be specified when setting global attributes. |

## *Published Data*

This activity does not publish any activity-specific data items.

# Start VM

The **Start VM** activity can be used in a runbook to start a stopped or suspended VM.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being started. |
| **Wait For** | Specifies a VM state that the activity will be waiting for before completing. Possible values are:<br>• **VM Powered On** – activity waits until the VM is in *poweredOn* state.<br>• **VM Guest OS running** – activity waits until VM Guest operating system is in *running* state.<br>• **VM Tools Running** – activity waits until VM tools are in *guestToolsRunning* running state.<br>• **Do Not Wait –** activity does not wait. |

## *Optional Properties*

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Datacenter ID (MOR)** | Identifies the datacenter in which the VM should be started. If this is not specified, the activity will use the first datacenter it finds in the list of datacenters available on the vCenter server. |
| **Override Automation Level** | Override the DRS automation level. Possible values are:<br>• **fullyAutomated** - Specifies that vCenter should automate both the migration of virtual machines and their placement with a host at power on.<br>• **manual** - Specifies that vCenter should generate recommendations for virtual machine migration and for placement with a host but should not implement the recommendations automatically.<br>• **partiallyAutomated** - Specifies that vCenter should generate recommendations for virtual machine migration and for placement with a host but should automatically implement only the placement at power on. |
| **Reserve Resources** | Reserve resources for the powering-on VMs throughout the power-on session. When this option is set to true, the server will return at most one recommended host per manual VM, and the VM's reservations are held on the recommended host until the VM is actually powered on (either by applying the recommendation or by a power-on request on the VM), or until |

| | |
|---|---|
| | the recommendation is cancelled, or until the recommendation expires. The expiration time is currently set to 10 minutes. This option does not have an effect on automatic VMs since their recommendations are executed immediately. This option is effective on DRS clusters only. |
| **Wait Timeout (seconds)** | Specifies the number of seconds before the activity times out when configured to wait for a VM state. ***Note:*** If this property is not specified and *Wait For* is configured so that the activity waits, the activity will not time out. |

## *Published Data*

This activity does not publish any activity-specific data items.

# Stop VM

The **Stop VM** activity can be used in a runbook to stop a running or suspended VM.

## *Required Properties*

You must configure the following properties:

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being stopped. |
| **Shutdown Guest OS** | Specifies whether the VM Guest OS will be shut down (soft stop) or if the VM will be powered off (hard stop). ***Note:*** *The VM must have VMware Tools installed and running in order for the activity to shut down (soft stop) the VM.* |
| **Wait for VM Powered Off** | Specifies whether the activity will be waiting for the VM to reach the *poweredOff* state before completing. |

## *Optional Properties*

You can use the following properties, as necessary, to control how the activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Wait Timeout (seconds)** | Specifies the number of seconds before the activity times out when configured to wait. ***Note:*** If this property is not specified and the activity is configured to wait, the activity will not time out. |

## *Published Data*

This activity does not publish any activity-specific data items.

# Suspend VM

The **Stop VM** activity can be used in a runbook to suspend a running VM. The following tables list the properties for this activity.

## *Required Properties*

You must configure the following properties.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being suspended. |

## *Optional Properties*

You can use the following optional properties.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |

## *Published Data*

This activity does not publish any activity-specific data items.

# Update VM

The **Update VM** activity can used in a runbook to update properties or device properties for an existing VM. The *Update* property specifies which area of the VM will be updated. Depending on the selected **Update** value, the activity will present different inputs.

## *Required Properties*

You must first configure the **Update** property by selecting what VM settings you want to update.

| | |
|---|---|
| **Update** | Specifies which part of the VM will be updated. Options include:<br>• CD/DVD Drive<br>• CPU<br>• Disk Drive<br>• Memory<br>• NIC<br>• VM Properties |

## *Required Properties (CD/DVD Drive)*

You must configure the following properties when updating **CD/DVD Drive** settings.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being updated. |
| **CD/DVD Drive Type** | Drive type:<br>• **Client Device** - connect the device to a physical CD/DVD device on the system from which you access the vSphere Client.<br>• **Host Device** - connect the device to a physical CD/DVD device on the host.<br>• **Datastore ISO File** - connect the device to an ISO file that is stored on a datastore accessible to the host. |
| **CD/DVD Drive Host Device Path** | Location of host device. This property is available only when *CD/DVD Drive Type = "Host Device"* |
| **CD/DVD Drive ISO File Path** | Location of the datastore ISO file for the device. This property is available only when *CD/DVD Drive Type = "Datastore ISO File"* |

## *Required Properties (CPU)*

You must configure the following properties when updating **CPU** settings.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being updated. |

### *Required Properties (Disk Drive)*

You must configure the following properties when updating **Disk Drive** settings.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being updated. |

### *Required Properties (Memory)*

You must configure the following properties when updating **Memory** settings.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being updated. |
| **Memory (MB)** | The amount of memory, in MB, for the updated VM. |

### *Required Properties (NIC)*

You must configure the following properties when updating **NIC** settings.

| | |
|---|---|
| **VM (MOR)** | THE MOR off the VM to update |

### *Required Properties (VM Properties)*

You must configure the following properties when updating **VM Property** settings.

| | |
|---|---|
| **VM (MOR)** | Managed object reference of the VM being updated. |

### *Optional Properties*

You can use the following optional properties, as necessary, to control how this activity runs.

| | |
|---|---|
| **Configuration** | Specifies which server configuration the activity should be using, when using multiple configurations. The specified value must match one of the configuration names specified in the IP configuration options. |
| **Description** | VM description for the updated VM. |
| **Guest OS ID** | Guest OS for the updated VM. |
| **VM Name** | VM name for the updated VM. |

### *Optional Properties (CD/DVD Drive)*

You can use the following properties, as needed, when updating **CD/DVD Drive** settings.

| | |
|---|---|
| **CD/DVD Drive ISO Datastore (MOR)** | Managed object reference of the datastore where the ISO file is stored. This property is available only when *CD/DVD Drive Type = "Datastore ISO File"* |

| | |
|---|---|
| **CD/DVD Drive Key** | Identifies the CD/DVD device to be updated. To obtain the device key you can use the Get vSphere Object activity with *Object Type = "VM Device"*. When not specified, the activity updates the first CD/DVD device it finds, if any. |
| **Connect at Power On** | Specifies if CD/DVD Drive should be connected when the VM powers on. |

## Optional Properties (CPU)

You can use the following properties, as needed, when updating **CPU** settings.

| | |
|---|---|
| **Number of Cores Per Socket** | Number of CPU cores per socket for the updated VM. |
| **Total Number of Cores** | Total number of CPU for the updated VM. |

## Optional Properties (Disk Drive)

You can use the following properties, as needed, when updating **Disk Drive** settings.

| | |
|---|---|
| **Disk Drive Key** | Identifies the disk drive device to be updated. To obtain the device key you can use the Get vSphere Object activity with *Object Type = "VM Device"*. When not specified, the activity updates the first disk drive device it finds, if any. |
| **Disk Size (MB)** | The size of the disk drive, in MB, for the updated VM. |

## Optional Properties (NIC)

You can use the following properties, as needed, when updating **NIC** settings.

| | |
|---|---|
| **Connect at Power On** | Specifies if NIC should be connected when the VM powers on. |
| **Network (MOR)** | Managed object reference of the network or distributed virtual port group to be assigned to the NIC in the updated VM. |
| **NIC Key** | Identifies the NIC device to be updated. To obtain the device key you can use the Get vSphere Object activity with *Object Type = "VM Device"*. When not specified, the activity updates the first NIC device it finds, if any. |

## Published Data

This activity does not publish any activity-specific data items.