PowerCLI Core

Document Version 1.10

NOTE: A text version of this document can be found as README.md in the download zip file for this fling.

Welcome!

PowerCLI Core uses Microsoft PowerShell Core and .Net Core to enable users of VMware Photon OS, Linux, Mac and Docker to now use the same cmdlets which were previously only available on windows.

PowerCLI Core enables a multi-platform scripting language which will allow you to manage your VMware infrastructure on any OS. Scripts written previously against the windows version are now made portable to a number of operating systems and can simply be loaded and run on these new OS versions without change.

PowerCLI Core can be downloaded from the VMware Flings site hereand used with he below instructions to be deployed.

PowerCLI Core vs PowerCLI for Windows

This initial version provides access to the core vSphere module including over 280 cmdlets allowing you to manage most of the major features of vCenter and ESXi. The below table shows the difference between the windows version and what is currently offered for PowerCLI Core:

Module	**Description**	**PowerCLI for Windows**	**PowerCLI Core**
Core	vCenter and ESXi Cmdlets	√	√
VDS	vSphere Distributed Switch Cmdlets	√	√
CIS	REST API functionality based Cmdlets	√	√
Storage	Storage Cmdlets	√	X
License	License View Cmdlet	√	X
VUM	Update Manager Cmdlets	√	X
Auto Deploy	Auto Deploy Cmdlets	√	X
Image Builder	Image Builder Cmdlets	√	X
VCD	vCloud Director Cmdlets	√	X
vCloud Air	vCloud Air Cmdlets	√	X

Installation

Generic Install

These are the general steps for the platforms supported by PowerShell.**For more detailed instructions see below.**

Step 1 - Install PowerShell by following the instructions for your operating system here

Step 2 - Locate the modules directory on your platform. Run in a command prompt:

```
powershell "\$env:PSMODULEPATH"
```

Step 3 - Unzip the PowerCLI.ViCore.zip and PowerCLI.Vds.zip into the modules directory identified in Step 2.

On Unix-like platforms the module directory usually is ~/.local/share/powershell/Modules

Mac Install

Install

These are the detailed instructions.

Step 1 - Download and Install PowerShell for Mac OS X using the instructions and packages from here this will also include the install of homebrew

Step 2 - Make sure you did not miss this step from the PowerShell installation instruction:

```
brew install openssl
brew install curl --with-openssl
```

Step 3 - Create the following directory if it does not exists by running the following command:

```
mkdir -p ~/.local/share/powershell/Modules
```

Step 4 - Extract the PowerCLI modules into the directory you created above by running the following command:

```
unzip PowerCLI.ViCore.zip -d ~/.local/share/powershell/Modules
unzip PowerCLI.Cis.zip -d ~/.local/share/powershell/Modules
unzip PowerCLI.Vds.zip -d ~/.local/share/powershell/Modules
```

Linux Install

Installing on VMware Photon OS 1.0

Step 1 - On the Photon machine Edit a new file in the following location /etc/yum.repos.d/powershell.repo and place the following content in it:

```
[powershell]
name=VMware Photon Linux 1.0(x86_64)
baseurl=https://vmware.bintray.com/powershell
gpgcheck=0
enabled=1
skip_if_unavailable=True
```

Step 2 - Install PowerShell onto Photon OS and create the modules folder needed later:

```
tdnf install -y powershell
mkdir -p ~/.local/share/powershell/Modules
```

Step 3 - From your download machine, copy the PowerCLI Modules from the downloaded fling zip file to the Photon machine, for example use scp as below:

```
scp PowerCLI* root@PHOTON_IP_ADDRESS:/root/.local/share/powershell/Modules
```

Step 4 - From your Photon machine, install Unzip on the photon box

```
tdnf install -y unzip
```

Step 5 - Extract the PowerCLI modules into the directory you copied them into above by running the following command:

```
cd /root/.local/share/powershell/Modules
unzip PowerCLI.ViCore.zip
unzip PowerCLI.Vds.zip
unzip PowerCLI.Cis.zip
```

Installing on Ubuntu 14.04.5 Server (64-bit)

Step 1 - Download PowerShell for Linux from here on your Ubuntu machine and install as below:

```
curl -SLO
https://github.com/PowerShell/PowerShell/releases/download/v6.0.0-
alpha.18/powershell_6.0.0-alpha.18-1ubuntu1.14.04.1_amd64.deb
sudo dpkg -i powershell_6.0.0-alpha.18-1ubuntu1.14.04.1_amd64.deb
sudo apt-get install -f
```

Step 2 - Create the following directory if it does not exist by running the following command:

```
mkdir -p ~/.local/share/powershell/Modules
```

Step 3 - From your download machine, copy the PowerCLI Modules from the downloaded fling zip file to the Photon machine under your users folders (replace "username" with your username), for example use scp as below:

```
scp PowerCLI*
username@UBUNTU_IP_ADDRESS:/home/username/.local/share/powershell/Modules
```

Step 4 - From the Ubunutu server extract the PowerCLI modules into the directory you created above by running the following command:

```
cd ~/.local/share/powershell/Modules
unzip PowerCLI.ViCore.zip
unzip PowerCLI.Vds.zip
unzip PowerCLI.Cis.zip
```

Launch PowerCLI

Step 1 - Open terminal

Step 2 - Start Powershell in the terminal by running the following command:

```
powershell
```

Step 3 - Import the PowerCLI Modules into your PowerShell Session:

```
Get-Module -ListAvailable PowerCLI* | Import-Module
```

Step 3a - (Optional - **Please Read**) If the SSL certificates of your vCenter are not trusted by your OS, disable SSL certificate validation for PowerCLI by running:

```
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore
```

Step 4 - Connect to your vCenter Server using Connect-VIServer

```
PS> Connect-VIServer -Server 192.168.1.51 -User
administrator@vSphere.local -Password VMware1!

Name                           Port  User
----                           ----  ----
192.168.1.51                   443   VSPHERE.LOCAL\Administrator
```

Docker Image

More information can be found on the docker PowerCLI Core image on[docker hub here]

Step 1 - Run the following to download the container from the docker hub:

```
docker pull vmware/powerclicore
```

Step 2 - Launch the container

```
docker run --rm -it vmware/powerclicore
```

More options for working with and running the container can be found here

Frequently Asked Questions

1. I am receiving "An error occurred while sending the request."

I am received the following error when using Connect-VIServer:

```
WARNING: Invalid server certificate. Use Set-PowerCLIConfiguration to set
the value for
the InvalidCertificateAction option to Prompt if you'd like to connect
once or to add a
permanent exception for this server.
connect-viserver : 10/17/16 3:00:15 PM     Connect-VIServer
              An error occurred while
sending the request.
At line:1 char:1
+ connect-viserver 10.192.116.20 -User administrator@vsphere.local -Pas
...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Connect-VIServer],
ViError
    + FullyQualifiedErrorId :
Client20_ConnectivityServiceImpl_Reconnect_Exception,VMwa
   re.VimAutomation.ViCore.Cmdlets.Commands.ConnectVIServer
```

This error is because the certificate on your vCenter server is not trusted by the machine you are making the connection from.

To fix this issue, replace the certificate chain on your machine or use the Set-PowerCLIConfiguration cmdlet to ignore certificate issues as below:

```
Set-PowerCLIConfiguration -InvalidCertificateAction Ignore
```

2. I am receiving an error - "The libcurl library ... do not support custom handling of certificates. A libcurl built with OpenSSL is required."

If you are using OS X, please update PowerShell to the latest version. This error is caused by limited support of .Net Core for your platform. Currently, .Net Core reqires that libcurl is built with OpenSSL and the one installed by your OS vendor is not. An option is to install a package that deploys a libcurl library built with OpenSSL and redirect the PowerShell executable to load it. If a package is not available for your OS, consider building libcurl with OpenSSL from source. To build libcurl from source see this for more info, in general download the source code then run in the :

```
./configure --with-ssl ; make
```

To deploy libcurl to /usr/local/lib:

```
sudo make install
```

Once you have a compatible libcurl, to make PowerShell load it, there are these options:

- (Not recomended) Change the variable `LD_LIBRARY_PATH` for your session to include the path where the new libcurl file is located. **This might affect any other executables that you launch in the same session (that inherit the variable`LD_LIBRARY_PATH`).**

```
export LD_LIBRARY_PATH=<path to libcurl directory>:$LD_LIBRARY_PATH
```

- Create a "shortcut" for PowerShell that would change the variable `LD_LIBRARY_PATH` only for the powershell process. Note that processes created by powershell would inherit the`LD_LIBRARY_PATH` value. Create a bash script that you would use to run powershell:

```
#!/bin/bash
export LD_LIBRARY_PATH=<path to libcurl directory>:$LD_LIBRARY_PATH
powershell
```

If you build from source, `make install` probably deployed libcurl in`/usr/local/lib`.

For Cent OS 7, a package you can use is the libcurl-openssl package from here: http://ftp.riken.jp/Linux/cern/centos/7/cern/x86_64/repoview/letter_l.group.html The path where it deploys the libcurl library is`/opt/shibboleth/lib64/`.

For more information, see

Known Issues

- PowerShell Core does not provide aliases for some of the well known PowerShell cmdlets, watch out for aliases like sleep and sort as these will run native linux commands, it is recommended you use the full cmdlet names like Sort-Object and Start-Sleep for example.
- The Get-VMHostHardware cmdlet has not yet been fully ported to PowerCLI Core and will provide an error when run
- The Get-VMHostPciDevice cmdlet has not yet been fully ported to PowerCLI Core and will provide an error when run
- The Open-VMConsoleWindow cmdlet has not yet been fully ported to PowerCLI Core and will provide an error when run
- The *-Tag, *-TagCategory, *-TagAssignment cmdlets are not supported with vSphere 6.5
- The Content Library Cmdlets have not yet been fully ported to PowerCLI Core and will provide an error when run
- The Credential store Cmdlets have not yet been fully ported to PowerCLI Core and will provide an error when run