VxRail Smart Upgrade Bundle PowerShell Script v1.0

Technical Note

Abstract

The VxRail Smart Upgrade Bundle PowerShell script v1.0 reduces the upgrade bundle distribution time from a central location at a data center to remote sites.

February 2022

Copyright

Contents

Executive summary

The VxRail Smart Upgrade Bundle PowerShell Script v1.0 reduces the upgrade bundle distribution time from a central location (depot) at a data center to remote sites. Starting with VxRail v4.7.300, you can download the upgrade bundle to a location in the data center, and then schedule VxRail cluster upgrades at each remote site.

Intended use and audience

This document is intended for customers, Dell Service providers who are authorized to work on a VxRail cluster, and VxRail administrators.

Revisions

Date	Description
February 2022	Update version information to align package file name change
August 2020	Updated Documentation reference
November 2019	Initial release

Overview

Starting with VxRail v4.7.300, you can also generate a partial upgrade bundle that contains only changed packages from the central depot. In many upgrade scenarios, partial bundles will be much smaller than the full bundle. For example, a VxRail-managed VMware vCenter Server image which is often larger than 1.5 G can be removed from the partial bundle for VxRail clusters that do not have a VxRail-managed VMware vCenter Server image. Sending the partial bundle instead of the full bundle to a remote site reduces the bundle loading time and improves user experience.

Supported scenarios

The VxRail Smart Upgrade Bundle PowerShell Script v1.0 provides PowerShell cmdlets for central depot services that include partial bundle generation, bundle uploading, LCM upgrading, and status reporting.

The following user scenarios are supported:

- Generate a partial bundle for a VxRail cluster on a data center file server.
- Perform upgrade bundle package upload from data center to remote VxRail cluster.
- Perform an LCM upgrade on the remote site VxRail cluster from data center.
- Perform an upgrade progress query of the remote VxRail from data center.

LCM Workflow

Following is a summary of the VxRail Smart Upgrade Bundle PowerShell Script v1.0 process.

1. Set up a directory for the upgrade bundle in a file system that can be accessed by the PowerShell scripts. In this document, this is referred to as a “depot.”
2. Download the full bundle from the Dell support site and store it in the depot you set up in Step 1.

Note: You are responsible for the disk space in the depot.

3. (Optional) Generate partial bundle for upgrade target remote site using depot scripts.

```
Initialize-PartialBundle
```

4. (Optional) Upload the generated partial bundle to the remote site VxRail cluster.

```
Send-PartialBundle
```

Note: You can interrupt and cancel the upload anytime during the uploading.

5. Start the upgrade of remote site VxRail cluster.

```
Invoke-VxRailUpgrade
```

6. Check the upgrade status.

```
Get-VxRailUpgradeStatus
```

7. Retry the upgrade from the last time of failure.

```
Invoke-VxRailUpgradeRetry
```

8. After the upgrade completes, delete directories that are no longer needed. Following is a graphical representation of the LCM workflow.

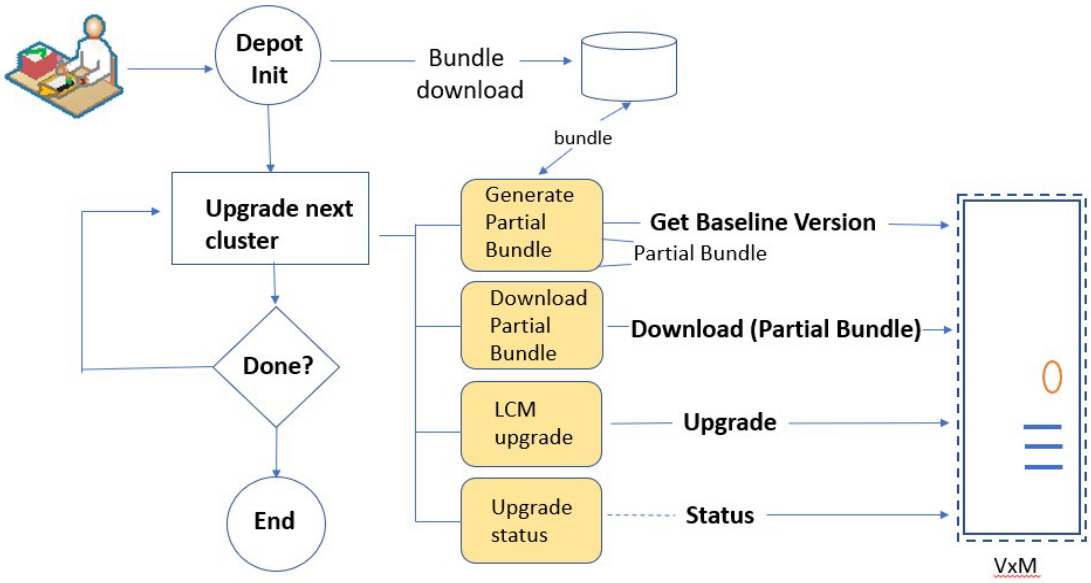

Figure 1. LCM workflow

How to import the Depot PowerShell cmdlet

Before you begin, ensure that you are using the following versions:

- .NET Framework v4.5+
- PowerShell v5

1. Modify the PSModulePath Installation Path.

- If your system environment does not have PSModulePath, create a new user environment variable called *PSModulePath* using the Environment Variables Editor in the System Properties dialog box.
- If the system environment already has PSModulePath, add a new user environment variable into PSModulePath using the Environment Variables Editor in the System Properties dialog box.

  For more information see Microsoft's documentation: https://docs.microsoft.com/enus/powershell/developer/module/modifying-the-psmodulepath-installation-path.

2. Ensure that the Depot module is located in the PSModulePath.

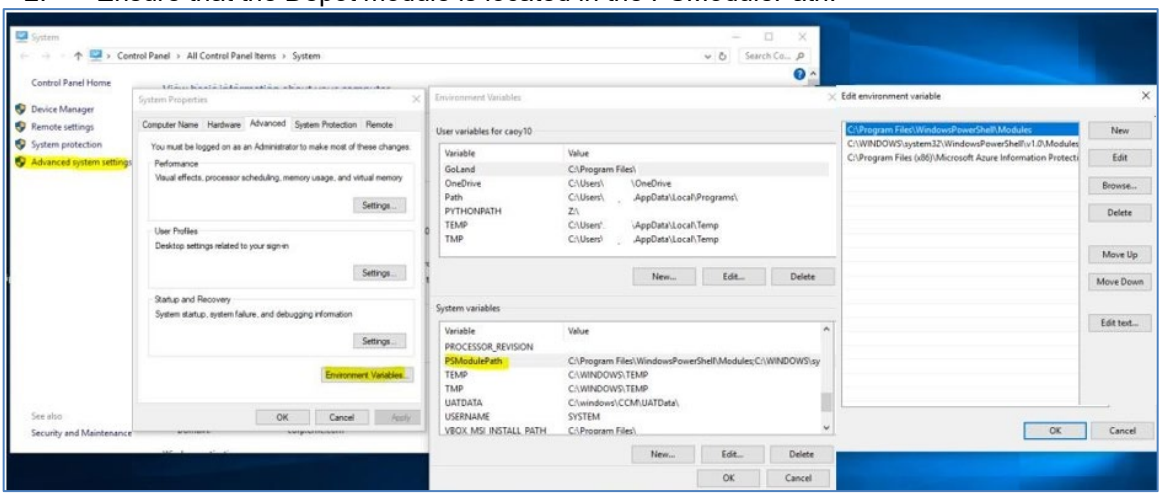

Figure 2. Environment variables editor

3. Run the command '`Import-Module VxRail.API.Depot`'.
4. Run the `Get-Module` command to confirm installation:

```
Windows PowerShell
PS C:\Users\caoy10> Get-Module

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   3.1.0.0    Microsoft.PowerShell.Management     {Add-Computer, Add-Content, Checkpoint-Computer, Clear-Con...
Manifest   3.1.0.0    Microsoft.PowerShell.Utility        {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
Script     1.2        PSReadline                          {Get-PSReadlineKeyHandler, Get-PSReadlineOption, Remove-PS...
Script     1.0.0.0    VxRail.API.Common
Script     1.0.0.0    VxRail.API.Depot                    {Get-VxRailUpgradeStatus, Initialize-PartialBundle, Invoke...


PS C:\Users\caoy10>
```

Figure 3. Importing the Depot PowerShell cmdlet

Available commands

Following are the commands available for the VxRail Smart Upgrade Bundle PowerShell script v1.0:

```
Get-Command -Module VxRail.API.Depot
```

```
PS C:\WINDOWS\system32> Get-Command -Module VxRail.API.Depot

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Function        Get-VxRailUpgradeStatus                            1.0.0.0    VxRail.API.Depot
Function        Initialize-PartialBundle                           1.0.0.0    VxRail.API.Depot
Function        Invoke-VxRailUpgrade                               1.0.0.0    VxRail.API.Depot
Function        Invoke-VxRailUpgradeRetry                          1.0.0.0    VxRail.API.Depot
Function        Send-PartialBundle                                 1.0.0.0    VxRail.API.Depot


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

Figure 4. Available commands

Partial Bundles

Initialize-PartialBundle

This command internally calls VxRail manager APIs to get all component versions and compare those versions against manifest file to identify which components need to be upgraded. It will generate a partial bundle, when possible, that contains only parts of the package that need to be upgraded. You can use this partial bundle to transfer to a local server or upload directly at remote sites. You can then perform LCM upgrades using the offline upgrade.

Syntax:

Initialize-PartialBundle [-Server] <string> [-Username] <string> [-Password] <string> [-Source] <string> [[-Output] <string>]

The -Output parameter is optional and can be used to specify where the partial bundle file is generated. If not specified, the partial bundle will be generated in the same directory as the upgrade bundle.

Sample:

Initialize-PartialBundle -Server '10.62.81.xxx' -Username 'administrator@vsphere.local' -Password 'xxxxx' -Source 'C:\company\workspace\sandbox-VXRAIL_COMPOSITE-4.7.99825971669_for_4.7.x.zip'

The following figure shows the initialization of the partial bundle.

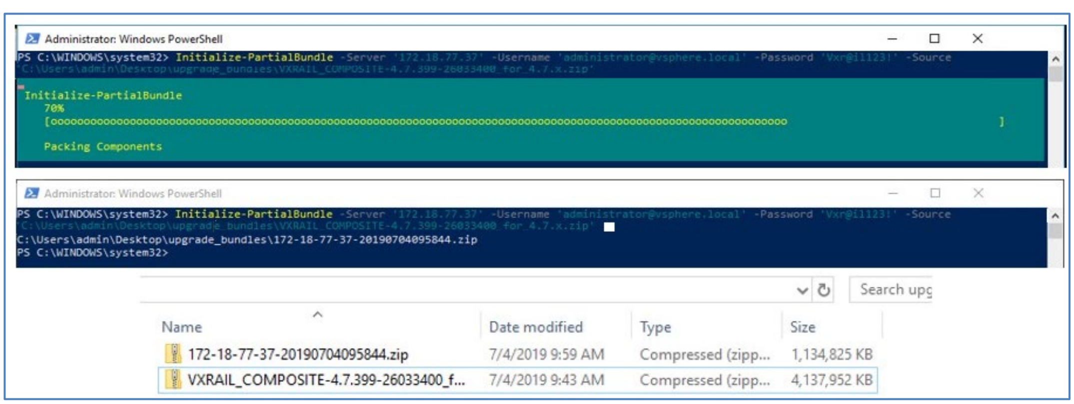

Figure 5. Initializing the partial bundle.

Send-PartialBundle

1. Leverage the partial bundle generation script to generate a partial bundle.
2. Upload the partial bundle to VxRail Manager using the `Send-PartialBundle` command. Syntax:
   Send-PartialBundle [-Server] <uri> [-Username] <string> [-Password] <string> [-FilePath] <string> [-Resume]

   If a previous `Send-PartialBundle` command failed for any reason while the upload was in progress, the `-Resume` parameter can be used to resume the upload process where it had stopped.

   Sample:

   Send-PartialBundle -Server '10.62.81.xxx' -Username 'administrator@vsphere.local' -Password 'xxx' -FilePath 'C:\company\workspace\sandbox-VXRAIL_COMPOSITE-7.0.00025967635_for_4.7.x.zip' -Resume

The following figure shows the partial bundle being sent:

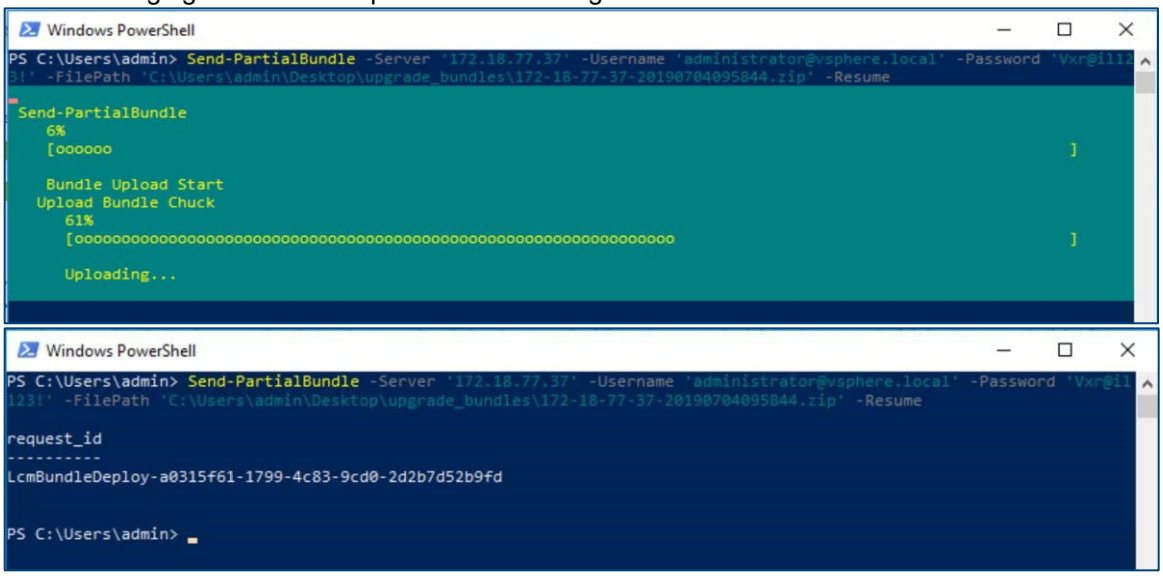

Figure 6. Uploading partial bundle.

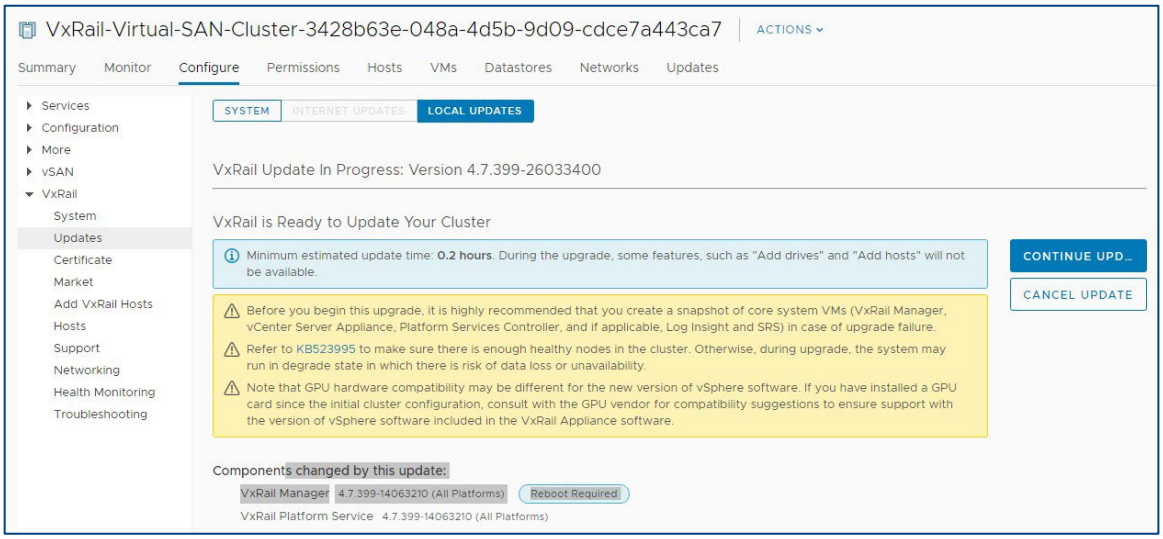

Figure 7. VxRail ready to update your cluster.

Invoke-VxRailUpgrade

1. Start the LCM upgrade using the `Invoke-VxRailUpgrade` command:
   Syntax:
   Invoke-VxRailUpgrade [-Server] <string> [-Username] <string> [-Password] <string> [[-Config] <string>]

   Sample:

   Invoke-VxRailUpgrade -Server '10.62.91.xxx' -Username 'administrator@vsphere.local' Password 'xxx' -Config 'C:\Users\caoy10\workspace\test.ini'

   Ini file structure:
   vxrail.vxm_root_user.username=ROOT-USERNAME

   vxrail.vxm_root_user.password=ROOT-PASSWORD

   vcenter.vc_admin_user.username=VC-USERNAME

   vcenter.vc_admin_user.password= VC-PASSWORD

For the `Invoke-VxRailUpgradeRetry` command, there is no need to supply the ini config file.

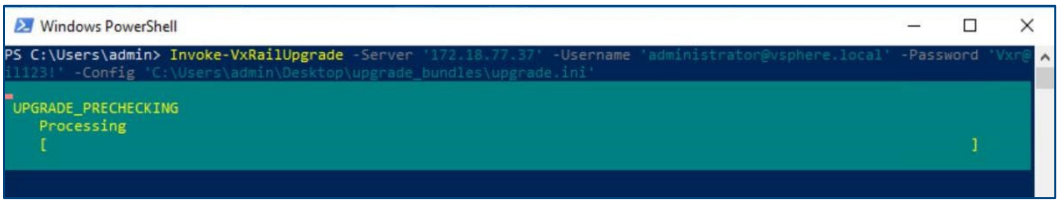

Figure 8. Upgrade in progress.

Get-VxRailUpgradeStatus

1. Query the upgrade progress using the `Get-VxRailUpgradeStatus` command.
2. Auto flag: Add this flag for automatic polling, as shown in the second screenshot below. Syntax:
   Get-VxRailUpgradeStatus [-Server] <string> [-Username] <string> [-Password] <string> [-Auto]

   Use the -Auto flag to continuously poll the upgrade until complete. See Figure Sample:

   Get-VxRailUpgradeStatus -Server '10.62.91.xxx' -Username 'administrator@vsphere.local' Password 'xxx' -Auto

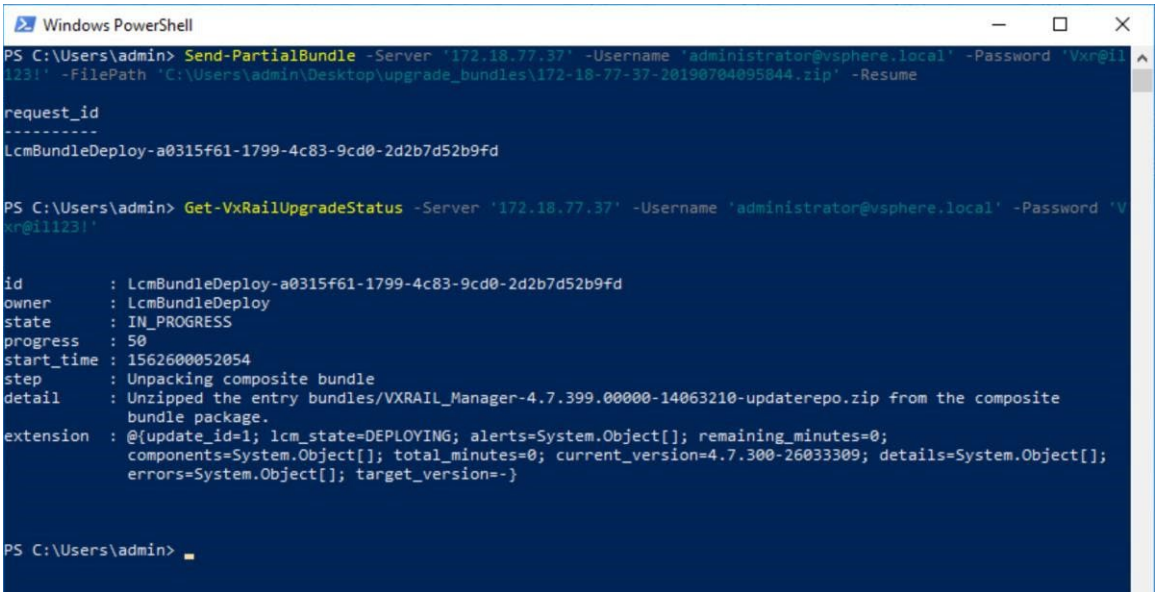

Figure 9. Status of the upgrade.

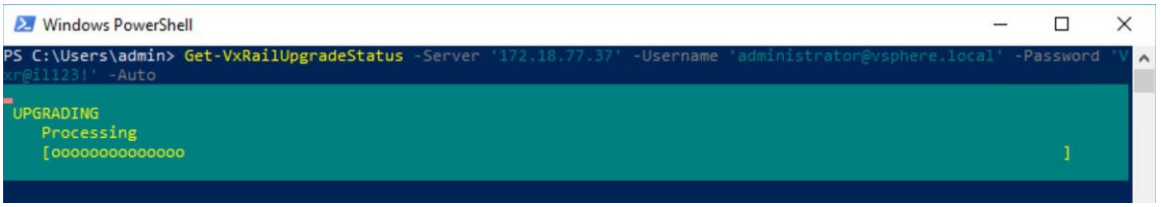

Figure 10. -Auto flag example.

More detailed information about APIs

The base URL for the VxRail API is: `https://<VxM_IP>rest/vxm/v1/`. The base URL is for a single VxRail cluster only. `<VxM_IP>` is the network IP address of the VxRail appliance.

For customer-supplied vCenter with multiple clusters, each cluster has its own distinct base URL.

The API User's Guide, located on the support website: https://dl.dell.com/content/manual55640916-dell-vxrail-api-user-guide.pdf?language=en-us&ps=true contains definitions of the VxRail APIs.

VxRail RESTful API documentation is also available onboard your VxRail Appliance in software versions 4.7.300 and later. To access onboard API documentation, enter the following address in a web browser on your host:

```
https://<VxM_IP>/rest/vxm/api-doc.html
```