

# Enabling Retreat Mode and Disabling HA on vSphere vSAN Cluster

PowerChute can execute PowerShell scripts via the Event Command File feature to enable Retreat Mode prior to placing the hosts into maintenance mode. When PowerChute re-starts, a separate script file is executed to disable Retreat Mode, which will re-instate the vCLS VMs.

There are 2 sets of scripts provided depending on whether PowerChute is installed on a Windows machine or you have deployed the PowerChute Linux-based Virtual Appliance to an ESXi host:

- [PowerChute Virtual Appliance](#)
- [PowerChute on Windows](#)

**NOTE:** In the provided scripts, replace the vCenter Server IP address (xx.xxx.xxx.xxx), username (pcnsadmin), password (password), and cluster name (C) to your credentials. You can edit the scripts using an ASCII text editor like Notepad.

Please enable **VM Prioritization** and add all VMs except the vCLS VMs to priority groups. Configure the Un-prioritized group with a 0-second duration for VM Shutdown and VM Startup:

## VM Prioritization

?

Define the order in which VMs in the cluster are migrated, shut down and started according to their priority.

Enable VM Prioritization



### Prioritize VMs

Set VM Priority by selecting VMs from the list on the left and drag them to a group on the right.

- RHEL76\_153
- rhelna5
- SiteB\_vMA-6.5
- vcent70
- vCLS (1)**
- vCLS (2)**
- vCLS (3)**
- vcsabl67\_242-DND
- VMware VCSA\_7.0U1\_DN
- Windows2019

- ★ High
  - VMware VCSA\_7.0U1\_DN
- ★ Medium
  - rhelna5
  - vcent70
  - SiteB\_vMA-6.5
  - vcsabl67\_242-DND
  - Windows2019
- ★ Low
- ★ Group 1

Remove

▶ Set VM Migration Duration

▶ Set VM Shutdown Duration

▶ Set VM Startup Duration

Apply

Undo

### Set VM Shutdown Duration

⚠ Values entered here will override the Duration values on the Virtualization Settings page

High	<input type="text" value="120"/>	seconds
Medium	<input type="text" value="120"/>	seconds
Low	<input type="text" value="0"/>	seconds
Group 1	<input type="text" value="0"/>	seconds
Group 2	<input type="text" value="120"/>	seconds
<b>Un-prioritized</b>	<input type="text" value="0"/>	seconds

▶ Set VM Startup Duration

Apply

Undo

Set VM Startup Duration

⚠ Values entered here will override the Duration values on the Virtualization Settings page

High	<input type="text" value="120"/>	seconds
Medium	<input type="text" value="120"/>	seconds
Low	<input type="text" value="0"/>	seconds
Group 1	<input type="text" value="0"/>	seconds
Group 2	<input type="text" value="120"/>	seconds
Un-prioritized	<input type="text" value="0"/>	seconds

Apply Undo

In vCenter Server Advanced settings, add a new entry with the name **config.vcls.clusters.domain-c<number>.enabled** and value = **True**.

To obtain a value for domain-c<number>, navigate to the Cluster which has vCLS running on it in the vSphere Web UI and copy the domain-c<number> from the URL e.g. the URL will be something like: <https://<fqdn-of-vCenter-server>/ui/app/cluster;nav=h/urn:vmomi:ClusterComputeResource:domain-c10001:eef257af-fa50-455a-af7a-6899324fabe6/summary>.

Copy the part in bold – in this case **domain-c10001**

The screenshot shows the vSphere Web UI interface. The browser address bar contains the URL: `10.179.14.249/ui/app/cluster;nav=h/urn:vmomi:ClusterComputeResource:domain-c10001:abcc10f-12b3-46d4-abe2-5f1eba04664c/configure/drs`. The `domain-c10001` portion of the URL is highlighted with a red rectangular box. The main content area shows the 'Cluster' configuration page for 'vSphere DRS is Turned ON'.

## Edit Advanced vCenter Server Settings



ⓘ Adding or modifying configuration parameters is unsupported and can cause instability. Configuration parameters cannot be removed once they are added. Continue only if you know what you are doing.

Name	Value	Summary
alarms.version	-1	Default alarm upgrade version
alarms.versionEx	111.0.18	Default alarm extended version
config.alarms.vim.version	vim.version.v7_0_1_1	--
config.drs.kvstore.local	False	--
config.license.client.IsNotificationsSync Seconds	30	--
config.license.client.oldServerIsNotificationsSyncSeconds	600	--
config.log.compressOnRoll	true	--
config.log.level	info	--
config.log.maxFileNum	30	--
config.log.maxFileSize	52428800	--

1 - 10 of 900 settings

Name: config.vcls.clusters.domain-c10001.enabled Value: True

ADD

CANCEL

SAVE

Name must start with 'config.' For example: config.log

10.179.14.249 | ACTIONS

Summary Monitor **Configure** Permissions Datacenters Hosts & Clusters VMs Datastores

Settings

- General
- Licensing
- Message of the Day
- Advanced Settings**
- vCenter HA

### Advanced vCenter Server Settings

Name	Value
config.vcls.clusters.domain-c10001.enabled	True

The following sections outline the steps required for configuring PowerChute to execute the scripts for the PowerChute Virtual Appliance (Linux CentOS 8) or for a Windows Installation.

## PowerChute Virtual Appliance

**Note:** Starting with PowerChute virtual appliance 4.4.2 PowerShell is pre-installed. When running PowerChute virtual appliance 4.4.2 or 4.4.3 you can skip to step 6.

1. Login to the PowerChute appliance via SSH or using VMware Remote Console and install PowerShell per the instructions in the following link - <https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-core-on-linux?view=powershell-7.1>  
See section Cent OS7.
  2. Install VMware PowerCLI – in the terminal type “pwsh” and then –
- details:

***Install-Module -Name VMware.PowerCLI***

3. To carry out an offline installation on a machine with internet access, run the command “Save-Module -Name VMware.PowerCLI -Path <path to download files>”. Transfer the downloaded Module folders to the PowerChute appliance via SCP and copy to the following path -  
***/opt/microsoft/powershell/7/Modules/***
4. Edit the .ps1 files using a text editor like vi and change the variables below to match your setup

***\$server = "<vCenter Server IP address>"***

***\$user = "<vCenter Server username>"***

***\$password = "<vCenter Server password>"***

***\$cluster = "<Name of the Cluster as it appears in vSphere Web Client UI>"***

5. Save the changes and then make the scripts executable by running the command –
6. Launch the PowerChute UI, open the **Event Configuration** page and click on **Run Command File** action for the event that you want to trigger a cluster shutdown for. Set the delay to match the delay which is set for the shutdown action.

Enter the full path to the PowerShell script and set a duration for the command file e.g. 300 seconds.

**NOTE:** This duration must allow time for the 3 vCLS VMs to be shut down and then removed from the inventory when Retreat Mode is enabled before PowerChute starts the maintenance mode tasks on each host.

**NOTE:** From PowerChute Network Shutdown v4.4 the script files must be placed in the ***/opt/APC/PowerChute/user\_files*** folder.

7. After the UPS Critical event has been resolved, an event driven command file can be used to disable retreat mode and re-instate the vCLS VMs on the cluster. It is recommended to use the following event in the pcnsconfig.ini file for this:

***event\_MonitoringStarted\_enableCommandFile = true***

***event\_MonitoringStarted\_commandFilePath =  
/opt/APC/PowerChute/user\_files/enableRetreatAppliance.ps1***

***event\_MonitoringStarted\_commandFileDelay = X***

***event\_MonitoringStarted\_commandFileDuration = 300***

This event will be triggered when the PowerChute Appliance starts up.

You can also the event below for an advanced configuration where the PowerChute Appliance is running on a Host outside the cluster:

***event\_CriticalEventResolved\_commandFileDelay = X***

***event\_CriticalEventResolved\_commandFilePath =  
/opt/APC/PowerChute/user\_files/enableRetreatAppliance.ps1***

***event\_CriticalEventResolved\_commandFileDuration = 300***

For example, you might have PowerChute deployed on a Management Host running the vSAN Witness Appliance with an Advanced UPS Configuration – in this scenario only the UPS protecting the Cluster Host switches to battery power so the PowerChute VM will not get re-started. In this case, the 2<sup>nd</sup> event type (CriticalEventResolved) can be used to disable retreat mode.

**NOTE:** In both cases, vCenter Server must be available before the PowerShell script can be used so the commandFileDelay value needs to allow enough time for the vCenter Server VM to power back up fully before the script is executed.

## PowerChute on Windows

1. Install VMware PowerCLI on the Windows machine where PowerChute is installed. Open PowerShell and type the command – ***Install-Module -Name VMware.PowerCLI***.

Alternatively, use the instructions in the following KBBase for an offline installation - <https://blogs.vmware.com/PowerCLI/2018/01/powercli-offline-installation-walkthrough.html>

2. Edit the PowerShell files (.ps1) using a text editor like NotePad to set the following variables for your environment:

***\$server = "<vCenter Server IP address>"***

***\$user = "<vCenter Server username>"***

***\$password = "<vCenter Server password>"***

***\$cluster = "<Name of the Cluster as it appears in vSphere Web Client UI>"***

Save both PowerShell script files.

3. Launch the PowerChute UI, open the **Event Configuration** page and click on **Run Command File** action.
4. Enter the full path to the disable.cmd file and set a duration for the command file e.g. 300 seconds.

**NOTE:** This duration must allow time for the 3 vCLS VMs to be shut down and then removed from the inventory when Retreat Mode is enabled before PowerChute starts the maintenance mode tasks on each host.

**NOTE:** From PowerChute v4.4, the command files must be saved to C:\Program Files\APC\PowerChute\user\_files folder.

The PowerShell scripts (.ps1 files) should be saved to the C:\ drive. Alternatively, you can edit the cmd file to point to a different location.

5. After the UPS Critical event has been resolved, an event driven command file can be used to disable retreat mode, re-instate the vCLS VMs and re-enable HA on the cluster. It is recommended to use the following event in the pcnsconfig.ini file for this:

***event\_MonitoringStarted\_enableCommandFile*** = true

***event\_MonitoringStarted\_commandFilePath*** = C:\Program Files\APC\PowerChute\user\_files\disable.cmd

***event\_MonitoringStarted\_commandFileDelay*** = X

***event\_MonitoringStarted\_commandFileDuration*** = 300

This event will be triggered when the PowerChute Appliance starts up.

You can also the event below for an advanced configuration where the PowerChute Appliance is running on a Host outside the cluster:

***event\_CriticalEventResolved\_commandFileDelay*** = X

***event\_CriticalEventResolved\_commandFilePath*** = C:\Program Files\APC\PowerChute\user\_files\enable.cmd

***event\_CriticalEventResolved\_commandFileDuration*** = 300

**NOTE:** In both cases, vCenter Server must be available before the powershell script can be used so the **commandFileDelay** value needs to allow enough time for the vCenter Server VM to power back up fully, before it is executed.

## SSH actions to prepare vSAN Cluster for Shutdown

Prior to placing the Hosts into maintenance mode and shutting them down, the following commands need to be executed:

1. esxcfg-advcfg -s 1 /VSAN/IgnoreClusterMemberListUpdates – this needs to be run on each host in the Cluster.

2. `python /usr/lib/vmware/vsan/bin/reboot_helper.py prepare` – this should only be executed on one host in the cluster. **NOTE:** This script will not run if there is a time drift > 2s between the various nodes so each host should be configured to use an NTP server to sync the time.

To run these commands, it is recommended to use SSH actions in PowerChute. Each SSH Action is executed sequentially:

SSH Actions		
host1_disable_updates	✓	🗑️ ✎
host2_disable_updates	✓	🗑️ ✎
host3_disable_updates	✓	🗑️ ✎
reboot_prepare	✓	🗑️ ✎
reboot_recover	✓	🗑️ ✎
host1_enable_updates	✓	🗑️ ✎
host2_enable_updates	✓	🗑️ ✎
host3_enable_updates	✓	🗑️ ✎

For each host in the vSAN Cluster, create an SSH action to run before Host Shutdown that executes the command `esxcfg-advcfg -s 1 /VSAN/IgnoreClusterMemberListUpdates` – this can be stored as a text file in the PowerChute/user\_files folder.

Then create an SSH action to run the reboot helper python script with prepare option. This should only be run on one host.

Next create an SSH action to run on startup that will run the reboot helper script with the recover option on startup on 1 host followed by SSH actions to run the command `esxcfg-advcfg -s 0 /VSAN/IgnoreClusterMemberListUpdates` on each host in the cluster.



**SSH Actions** [X]

Name	host1_disable_updates	
User Name	pcnsadmin	
Password	.....	
SSH Key File Path		
SSH Key File Password		
IP Address/Hostname	10.179.xxx.xxx	
Port	22	
Path to SSH command file	h_disable_clustermember_updates.txt	
SSH Action Delay	0	seconds
SSH Action Duration	30	seconds
Execute SSH Action	Before Host Shutdown	
Enable SSH Action	<input checked="" type="checkbox"/>	

OK Cancel

**SSH Actions** [X]

Name	reboot_prepare	
User Name	pcnsadmin	
Password	.....	
SSH Key File Path		
SSH Key File Password		
IP Address/Hostname	10.179.xxx.xxx	
Port	22	
Path to SSH command file	/opt/APC/PowerChute/user_files/reboc	
SSH Action Delay	0	seconds
SSH Action Duration	180	seconds
Execute SSH Action	Before Host Shutdown	
Enable SSH Action	<input checked="" type="checkbox"/>	

OK Cancel

**SSH Actions** [X]

Name	<input type="text" value="reboot_recover"/>	
User Name	<input type="text" value="pcnsadmin"/>	
Password	<input type="password" value="....."/>	
SSH Key File Path	<input type="text"/>	
SSH Key File Password	<input type="password"/>	
IP Address/Hostname	<input type="text" value="10.179.xxx.xxx"/>	
Port	<input type="text" value="22"/>	
Path to SSH command file	<input type="text" value="/opt/APC/PowerChute/user_files/rebo"/>	
SSH Action Delay	<input type="text" value="0"/>	seconds
SSH Action Duration	<input type="text" value="120"/>	seconds
Execute SSH Action	<input type="text" value="On Startup"/>	
Enable SSH Action	<input checked="" type="checkbox"/>	

OK Cancel

**SSH Actions** [X]

Name	<input type="text" value="host2_enable_updates"/>	
User Name	<input type="text" value="pcnsadmin"/>	
Password	<input type="password" value="....."/>	
SSH Key File Path	<input type="text"/>	
SSH Key File Password	<input type="password"/>	
IP Address/Hostname	<input type="text" value="10.179.xxx.xxx"/>	
Port	<input type="text" value="22"/>	
Path to SSH command file	<input type="text" value="/opt/APC/PowerChute/user_files/ssh_"/>	
SSH Action Delay	<input type="text" value="0"/>	seconds
SSH Action Duration	<input type="text" value="30"/>	seconds
Execute SSH Action	<input type="text" value="On Startup"/>	
Enable SSH Action	<input checked="" type="checkbox"/>	

OK Cancel

Once the SSH actions have been configured, the pcnsconfig.ini file needs to be updated to set the ESXi host prompt so that the SSH action will execute. To do this add the following line to each SSH action "ssh\_prompt\_regex = \\s" e.g.

```
[SSHAction4]
ssh_action_name = reboot_recover
ssh_target = 10.179.xxx.xxx
ssh_command_path = /opt/APC/PowerChute/user_files/reboot_helper_recover.txt
ssh_action_enabled = true
action_delay = 0
action_duration = 120
ssh_port = 22
ssh_action_sequence = startup
ssh_prompt_regex = \]\s
```

```
[SSHAction5]
ssh_action_name = host1_enable_updates
ssh_target = 10.179.xxx.xxx
ssh_command_path =
/opt/APC/PowerChute/user_files/ssh_enable_clustermember_updates.txt
ssh_action_enabled = true
action_delay = 0
action_duration = 30
ssh_port = 22
ssh_action_sequence = startup
ssh_prompt_regex = \]\s
```