Shutting down a HyperFlex Cluster with PowerChute Network Shutdown

This document provides guidance on using PowerChute Network Shutdown to gracefully shut down a HyperFlex Cluster in response to a power outage using a script that is executed by the "Shutdown Command File" feature in PowerChute. This applies to HyperFlex Edge Cluster (2 Node deployment with Intersight) and 3-4 Node Clusters. For HyperFlex Edge, details on handling Fabric Interconnects do not apply.

There are two solutions provided depending on how PowerChute is deployed. If installed on Windows, a powershell script is used, whereas a linux shell script is used for the PowerChute Virtual Appliance that can be deployed as a VM on a Management Host outside the HyperFlex Cluster. For HyperFlex Edge clusters, the PowerChute Appliance can be deployed on local storage on one of the nodes.

In both cases the VM Prioritization feature in PowerChute is used to avoid shutting down the HyperFlex Storage Controller VMs before the User VMs and Cluster Services have been shut down.

Please refer to the PowerChute Installation Guide and VMware User Guides for details on how to install PowerChute for VMware here.

NOTE: HyperFlex Stretched Cluster is not currently supported.

Windows Powershell Script

If running PowerChute on a Windows machine outside the cluster, the sample powershell script attached to this Kbase can be used to shut down the HyperFlex Cluster.

Edit the variables below at the top of the file to match your environment:

$cluster_ip = "<HyperFlex Cluster IP address>"

$username = "<HyperFlex Local Admin user>"

$password = "<HyperFlex Password>"

$vsphere_user = "<ESXi host username>" (This account must exist on each ESXi host)

$vsphere_pwd = "<ESXi host password>"

$cluster_stop_duration = <Time in seconds for Cluster Stop command to complete>

$RETRY_COUNT=1 <Do not modify this value>

$RETRY_LIMIT=5 <Number of times PowerChute should wait/re-try if the cluster shutdown takes longer than the cluster stop duration>

$RETRY_INTERVAL=60 <Time in seconds to wait between re-tries>

$HXHosts = "hx-1.scvmmgal.net"," hx-2.scvmmgal.net" <IP address/FQDN of each ESXi Host in the HyperFlex cluster>

ShutdownHX.cmd file

Enter the path to the powershell script in this file e.g.

cmd.exe /C powershell.exe ; *C:\ Scripts_HyperFlex\shutdown_cluster_win.ps1*

NOTE:

1. Powershell 5.1 required – this can be installed via Windows Management Framework 5.1.
2. Change powershell execution policy to allow the script to run by executing the command – Set-ExecutionPolicy RemoteSigned.
3. The script uses VMware PowerCli to connect to the ESXi hosts to power off the Controller VMs. In powershell 5.1 this can be installed by running the command –

   Install-Module -Name VMware.PowerCli

   Alternatively, you can create a folder e.g. C:\PowerCLI and run this command

   Save-Module -Name VMware.PowerCLI -Path C:\PowerCLI

   In the powershell script, set this path as shown:

   *$env:PSModulePath = $env:PSModulePath + ";**c:\PowerCLI**"*

Linux Shell Script for PowerChute Virtual Appliance

Edit the variables below in the shell script:

USER=<HyperFlex local admin user>

PASSWORD=<HyperFlex password>

ESXUSER=<ESXi user>

CLUSTER_IP="192.168.100.241" (HyperFlex Cluster IP Address)

HOST_IP[1]="192.168.100.2" (ESXi Host IP address)

HOST_IP[2]="192.168.100.3" (ESXi Host IP address)

HOST_IP[3]=

HOST_IP[4]=

NUM_CVMS=3

SHUTDOWN_TIME=240

RETRY_COUNT=1

RETRY_LIMIT=5

RETRY_INTERVAL=60

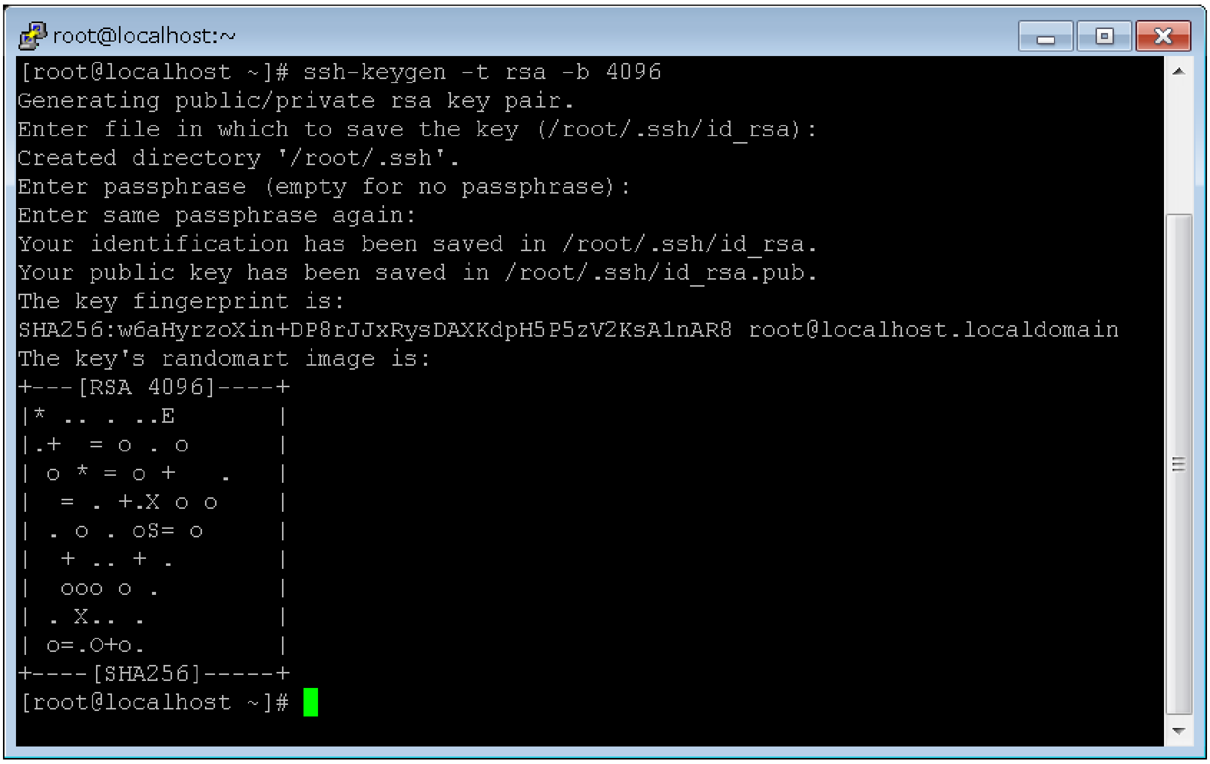
Copy the script to a folder on the PowerChute Appliance VM via SCP. The SSHD service should be running after the Appliance is deployed and the initial setup is completed.

Add execute permissions for the script using the command – **chmod +x shutdown_hyperflex.sh**

Setting up Shared SSH Keys for connection to ESXi hosts

The following outlines the steps needed to create shared SSH keys to allow the script to connect to the ESXi hosts without a password prompt in order to shut down the Storage Controller VMs.

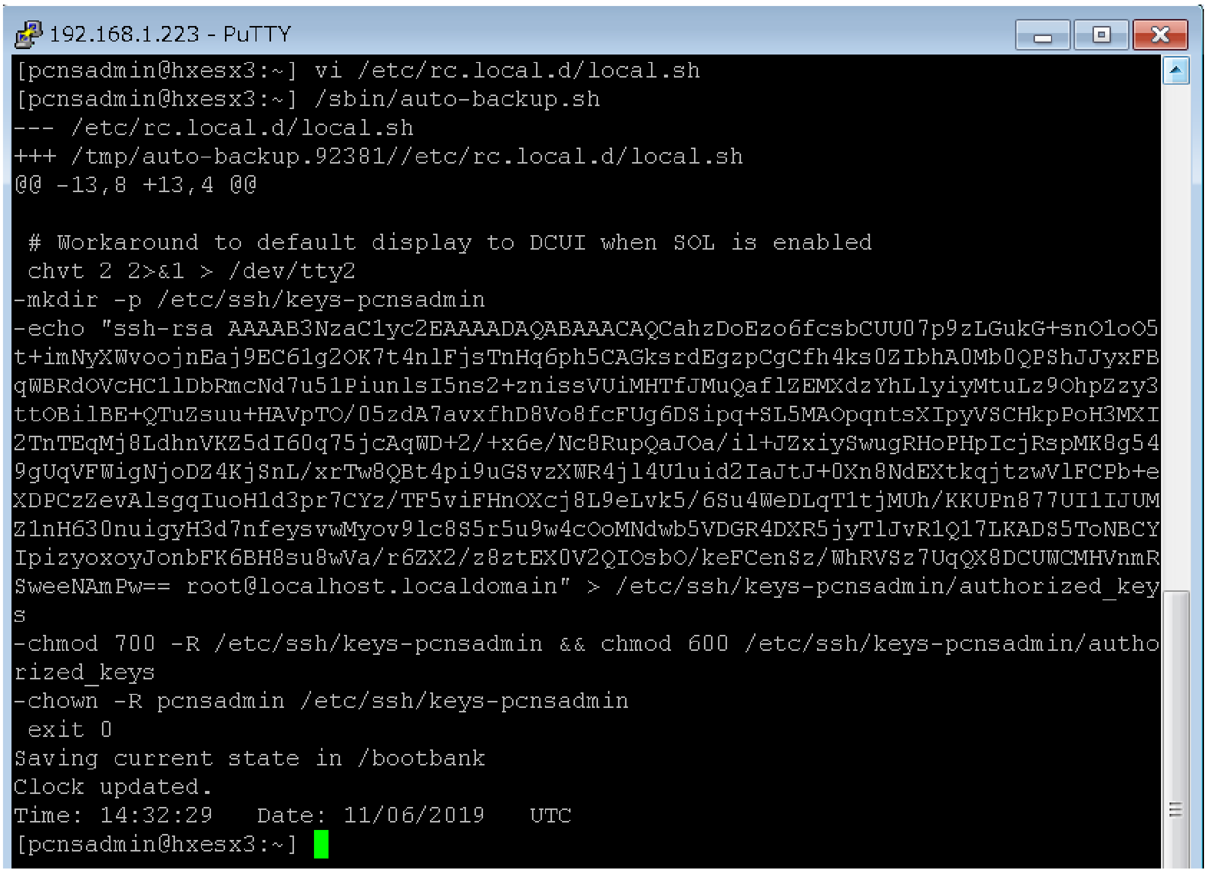
1. Connect to the PowerChute Appliance over SSH using a tool like putty or use VMware Web Console.
2. Download the Jq binary from https://stedolan.github.io/jq/ and upload via SCP to the PowerChute Appliance. Copy the file to the /usr/bin folder and re-name it as **jq.** Grant execute permissions also using command **chmod +x /usr/bin/jq**
3. Generate SSH keypair (used to connect to the ESXi hosts to power off the controller VMs using vim-cmd) using the following command: **ssh-keygen -t rsa -b 4096**
   Leave passphrase blank when prompted. This will create 2 files – id_rsa and id_rsa.pub in the .ssh directory on the PowerChute Appliance.



```
[root@localhost ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:w6aHyrzoXin+DP8rJJxRysDAXKdpH5P5zV2KsA1nAR8 root@localhost.localdomain
The key's randomart image is:
+---[RSA 4096]----+
|* .. . ..E       |
|.+  = o . o      |
| o * = o +   .   |
|  = . +.X o o    |
| . o . oS= o     |
|  + .. + .       |
|  ooo o .        |
| . X.. .         |
| o=.O+o.         |
+----[SHA256]-----+
[root@localhost ~]#
```

4. Connect to each ESXi host and run the following commands to set up key pairs for SSH connection to each host:
   a. mkdir -p /etc/ssh/keys-**<username>**
   b. echo **"ssh-rsa AAAAB3..."** > /etc/ssh/keys-**<username>**/authorized_keys (This is the public key contained in file /root/.ssh/id_rsa.pub file that was created in step 3)
   c. chmod 700 -R /etc/ssh/keys-**<username>** && chmod 600 /etc/ssh/keys-**<username>**/authorized_keys
   d. chown -R **<username>** /etc/ssh/keys-<username>
5. Add the lines above to “/etc/rc.local.d/local.sh” on the ESXi host before “exit 0” – this is necessary to ensure that they keys-<username> directory and authorized_keys file persist on the ESXi host following a reboot.

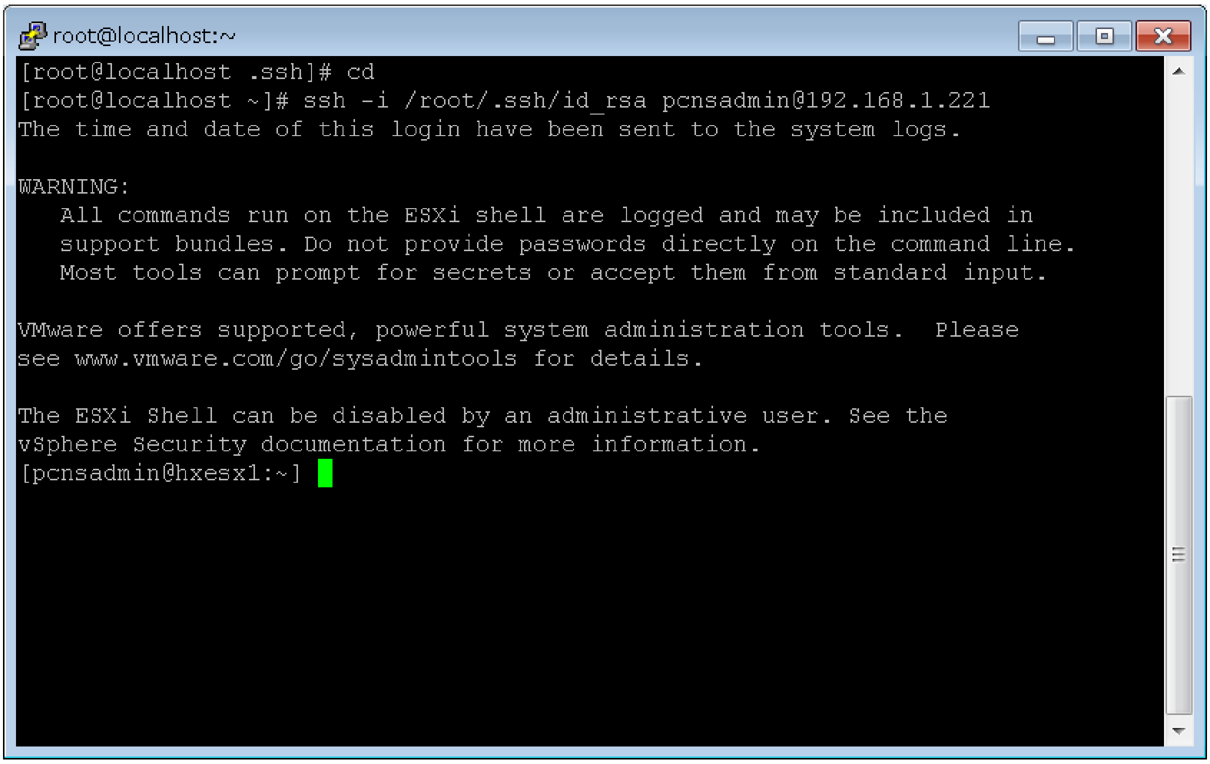
6. Run the command "/sbin/auto-backup.sh"



```
[pcnsadmin@hxesx3:~] vi /etc/rc.local.d/local.sh
[pcnsadmin@hxesx3:~] /sbin/auto-backup.sh
--- /etc/rc.local.d/local.sh
+++ /tmp/auto-backup.92381//etc/rc.local.d/local.sh
@@ -13,8 +13,4 @@

 # Workaround to default display to DCUI when SOL is enabled
 chvt 2 2>&1 > /dev/tty2
-mkdir -p /etc/ssh/keys-pcnsadmin
-echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCahzDoEzo6fcsbCUU07p9zLGukG+snO1oO5
t+imNyXWvoojnEaj9EC61g2OK7t4nlFjsTnHq6ph5CAGksrdEgzpCgCfh4ks0ZIbhA0Mb0QPShJJyxFB
qWBRdOVcHC1lDbRmcNd7u51PiunlsI5ns2+znissVUiMHTfJMuQaflZEMXdzYhLlyiyMtuLz9OhpZzy3
ttOBilBE+QTuZsuu+HAVpTO/05zdA7avxfhD8Vo8fcFUg6DSipq+SL5MAOpqntsXIpyVSCHkpPoH3MXI
2TnTEqMj8LdhnVKZ5dI60q75jcAqWD+2/+x6e/Nc8RupQaJOa/il+JZxiySwugRHoPHpIcjRspMK8g54
9gUqVFWigNjoDZ4KjSnL/xrTw8QBt4pi9uGSvzXWR4jl4U1uid2IaJtJ+0Xn8NdEXtkqjtzwVlFCPb+e
XDPCzZevAlsgqIuoH1d3pr7CYz/TF5viFHnOXcj8L9eLvk5/6Su4WeDLqTltjMUh/KKUPn877UIlIJUM
Z1nH630nuigyH3d7nfeysvwMyov9lc8S5r5u9w4cOoMNdwb5VDGR4DXR5jyTlJvR1Q17LKADS5ToNBCY
IpizyoxoyJonbFK6BH8su8wVa/r6ZX2/z8ztEX0V2QIOsbO/keFCenSz/WhRVSz7UqQX8DCUWCMHVnmR
SweeNAmPw== root@localhost.localdomain" > /etc/ssh/keys-pcnsadmin/authorized_key
s
-chmod 700 -R /etc/ssh/keys-pcnsadmin && chmod 600 /etc/ssh/keys-pcnsadmin/autho
rized_keys
-chown -R pcnsadmin /etc/ssh/keys-pcnsadmin
 exit 0
Saving current state in /bootbank
Clock updated.
Time: 14:32:29   Date: 11/06/2019   UTC
[pcnsadmin@hxesx3:~]
```

7. Verify that you can connect to each host using the command: ssh -i /root/.ssh/id_rsa <username>@<esxi_host_ip_address> . This is necessary to add the hosts to a list of trusted hosts when prompted.



```
[root@localhost .ssh]# cd
[root@localhost ~]# ssh -i /root/.ssh/id_rsa pcnsadmin@192.168.1.221
The time and date of this login have been sent to the system logs.

WARNING:
   All commands run on the ESXi shell are logged and may be included in
   support bundles. Do not provide passwords directly on the command line.
   Most tools can prompt for secrets or accept them from standard input.

VMware offers supported, powerful system administration tools.  Please
see www.vmware.com/go/sysadmintools for details.

The ESXi Shell can be disabled by an administrative user. See the
vSphere Security documentation for more information.
[pcnsadmin@hxesx1:~]
```

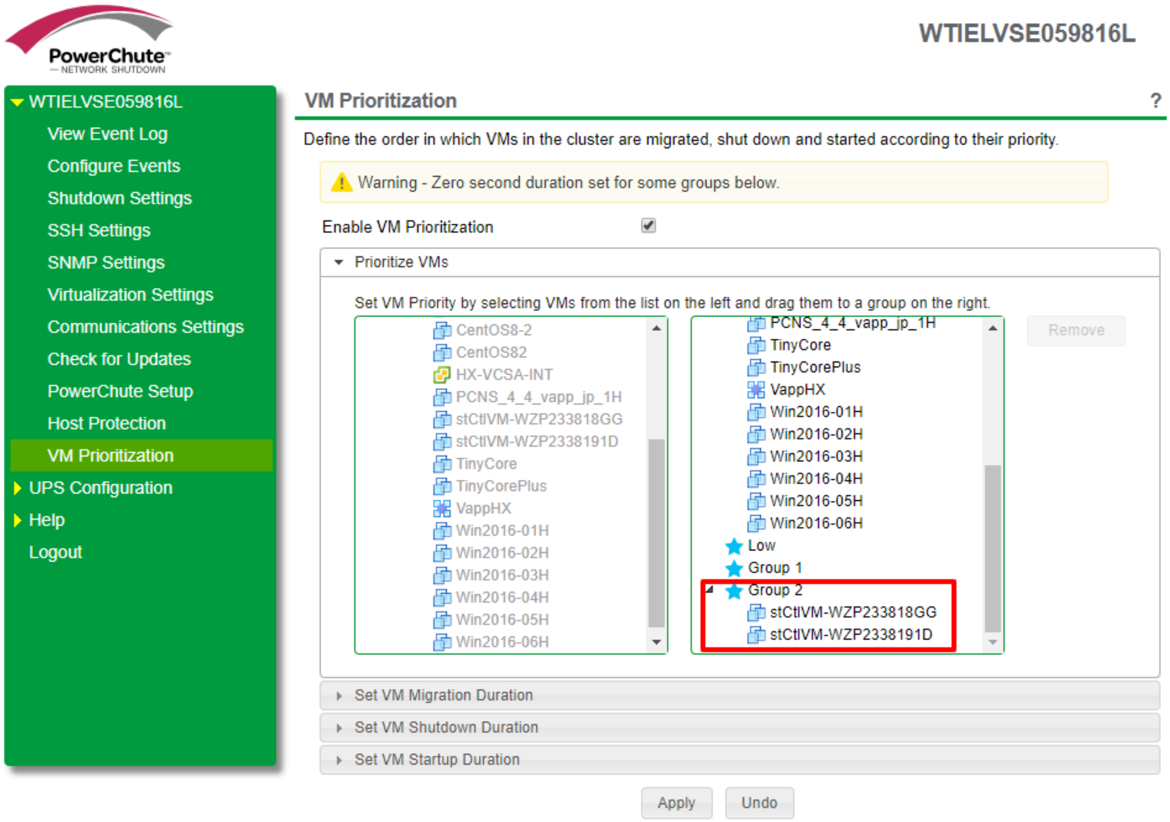
8. Repeat steps 4-7 for each ESXi host. NOTE: If using the root account on the ESXi host then /etc/ssh/keys-root/authorized_keys is already persisted and you would only need to add the public key (id_rsa.pub) to this file.

PowerChute Setup

Install PowerChute on a Windows machine or deploy the Virtual Appliance per the instructions in the installation guide. Configure PowerChute for “Managed by vCenter” and Single or Redundant UPS Configuration. When adding the vCenter Server credentials, a shared user account that exists on each of the ESXi Hosts being protected should be used in the event that vCenter Server is unavailable when a power outage occurs.

Having completed the installation and configuration, open the VM Prioritization page.

Enable VM Prioritzation – add User VMs to priority groups and add the Storage Controller VMs to lowest priority group:



VM Prioritization

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

⚠ Warning - Zero second duration set for some groups below.

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.

CentOS8-2
CentOS82
HX-VCSA-INT
PCNS_4_4_vapp_jp_1H
stCtlVM-WZP233818GG
stCtlVM-WZP2338191D
TinyCore
TinyCorePlus
VappHX
Win2016-01H
Win2016-02H
Win2016-03H
Win2016-04H
Win2016-05H
Win2016-06H

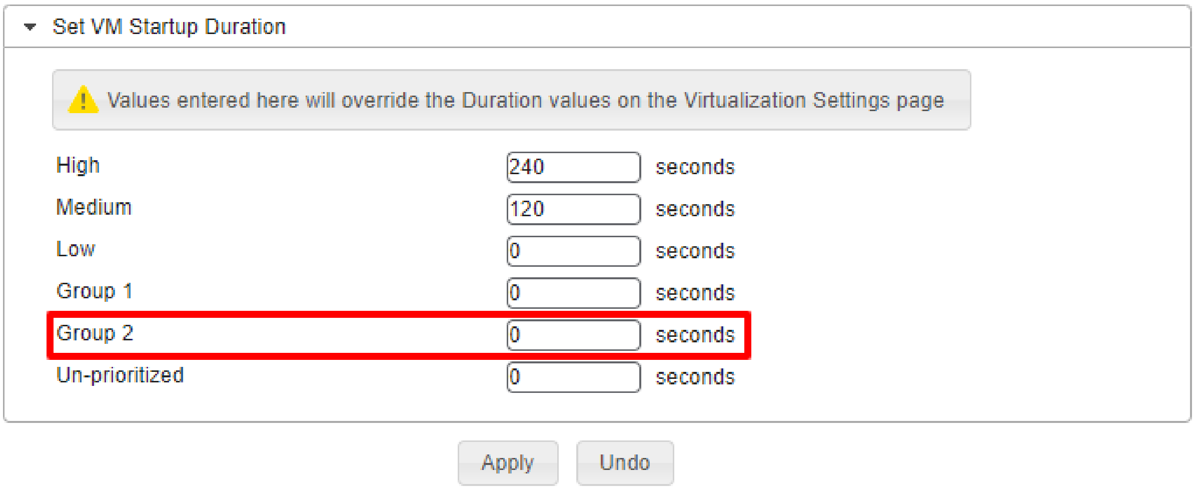
PCNS_4_4_vapp_jp_1H
TinyCore
TinyCorePlus
VappHX
Win2016-01H
Win2016-02H
Win2016-03H
Win2016-04H
Win2016-05H
Win2016-06H
Low
Group 1
Group 2
stCtlVM-WZP233818GG
stCtlVM-WZP2338191D

Set VM Migration Duration

Set VM Shutdown Duration

Set VM Startup Duration

Set the VM Shutdown duration for the group with the Controller VMs to 0 seconds. This ensures that PowerChute will not shut down the Controller VMs during the VM shutdown step.



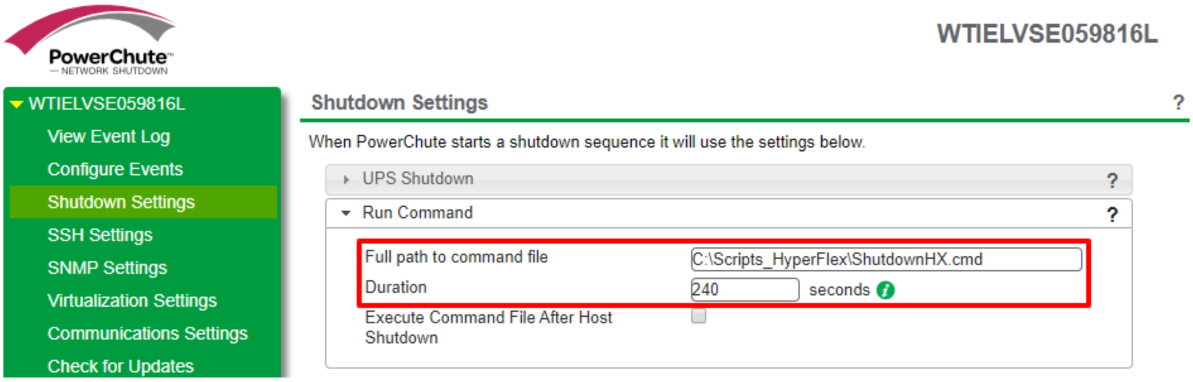
Set VM Startup Duration

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

High	240	seconds
Medium	120	seconds
Low	0	seconds
Group 1	0	seconds
Group 2	0	seconds
Un-prioritized	0	seconds

Next, configure PowerChute to execute the desired script to handle shutting down the HyperFlex cluster service and then power off the Storage Controller VMs prior to shutting down the ESXi hosts.

To do this, open the Shutdown Settings page and expand the Run Command File section. Enter the full path to the cmd file (for Windows ShutdownHX.cmd) or shell script file (PowerChute Appliance – Linux based OS) and a duration value in seconds.



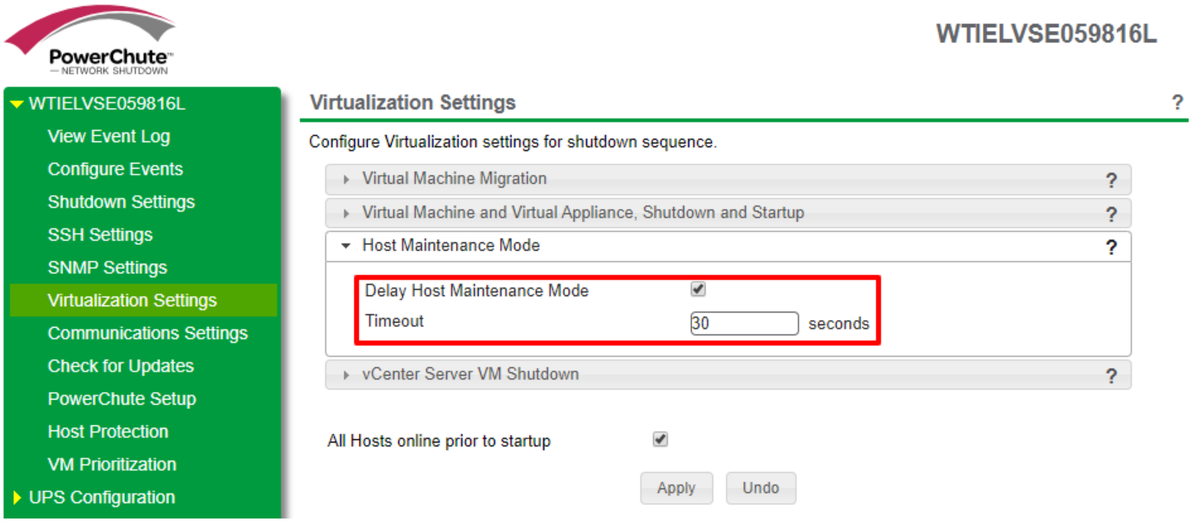
Shutdown Settings

When PowerChute starts a shutdown sequence it will use the settings below.

Full path to command file	C:\Scripts_HyperFlex\ShutdownHX.cmd
Duration	240 seconds
Execute Command File After Host Shutdown	□

The duration value is the time allowed for the script to shut down the HyperFlex cluster service and power off the Controller VMs.

We also recommend enabling “Delay Maintenance Mode” option on the Virtualization Settings page.



Configure Virtualization settings for shutdown sequence.

Delay Host Maintenance Mode ☑

Timeout 30 seconds

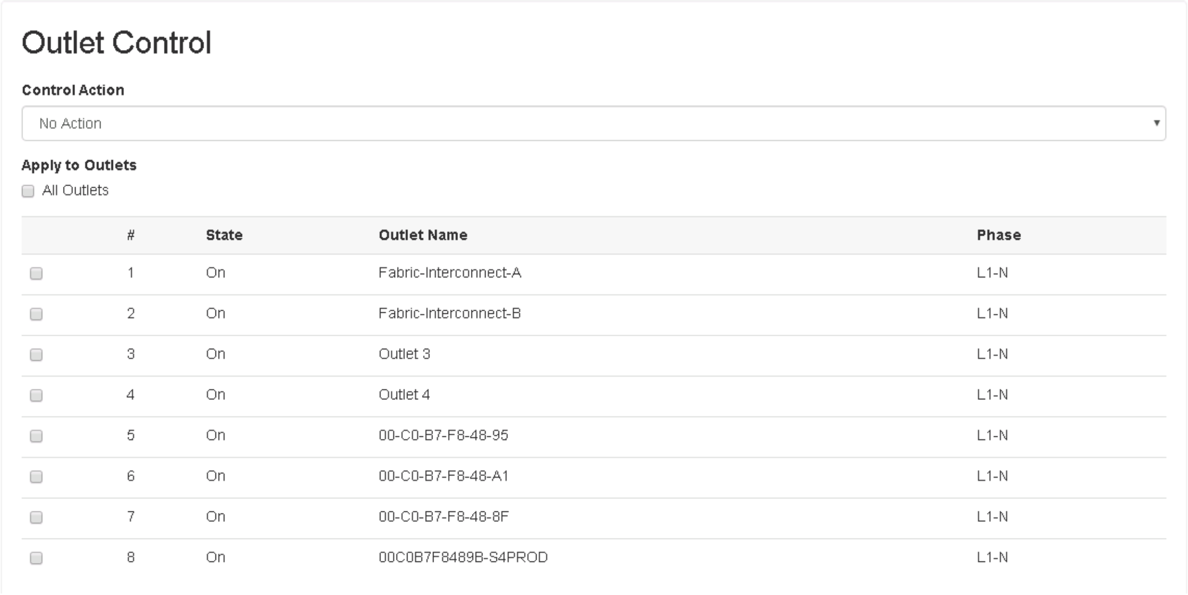
All Hosts online prior to startup ☑

With this option enabled, PowerChute will wait until all User VMs and the CVMs are shut down and powered off before placing the hosts into maintenance mode and shutting them down.

NOTE: If the PowerChute appliance is deployed on one of the nodes, then this node will not enter Maintenance mode.

Powering off the Fabric Interconnects in order

This assumes that the Fabric Interconnects are connected to an APC Switched Rack PDU as shown below. This section does not apply if using a 2 Node HyperFlex Edge cluster with Intersight.



Control Action

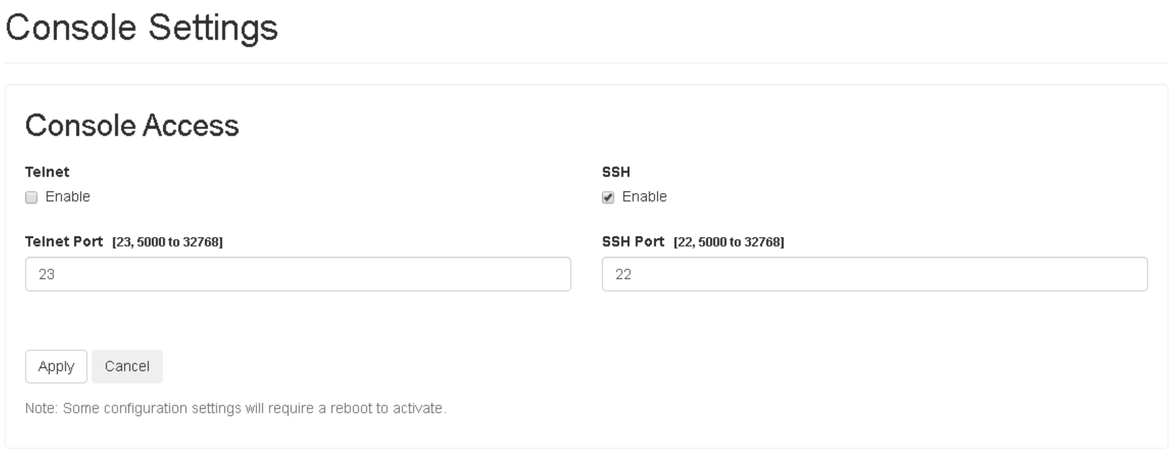
No Action

Apply to Outlets

□ All Outlets

	#	State	Outlet Name	Phase
□	1	On	Fabric-Interconnect-A	L1-N
□	2	On	Fabric-Interconnect-B	L1-N
□	3	On	Outlet 3	L1-N
□	4	On	Outlet 4	L1-N
□	5	On	00-C0-B7-F8-48-95	L1-N
□	6	On	00-C0-B7-F8-48-A1	L1-N
□	7	On	00-C0-B7-F8-48-8F	L1-N
□	8	On	00C0B7F8489B-S4PROD	L1-N

SSH console access should be enabled on the PDU:



Console Settings

Console Access

Telnet

□ Enable

Telnet Port [23, 5000 to 32768]

23

SSH

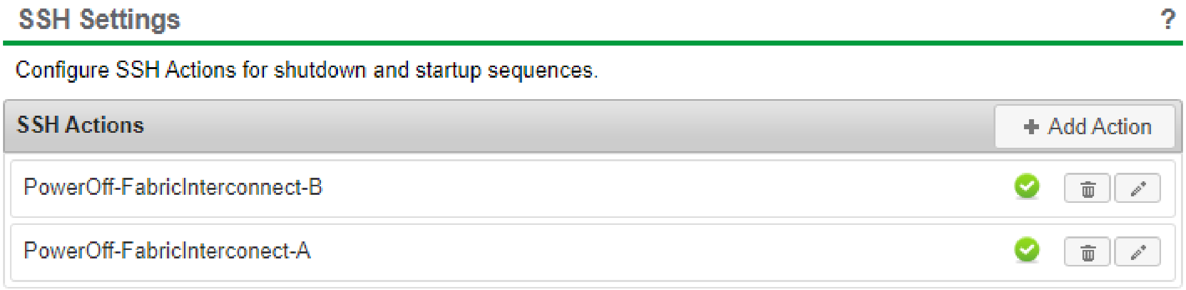
☑ Enable

SSH Port [22, 5000 to 32768]

22

Note: Some configuration settings will require a reboot to activate.

In PowerChute we configure SSH actions which will send a CLI console command to turn off each outlet on the PDU in sequence.



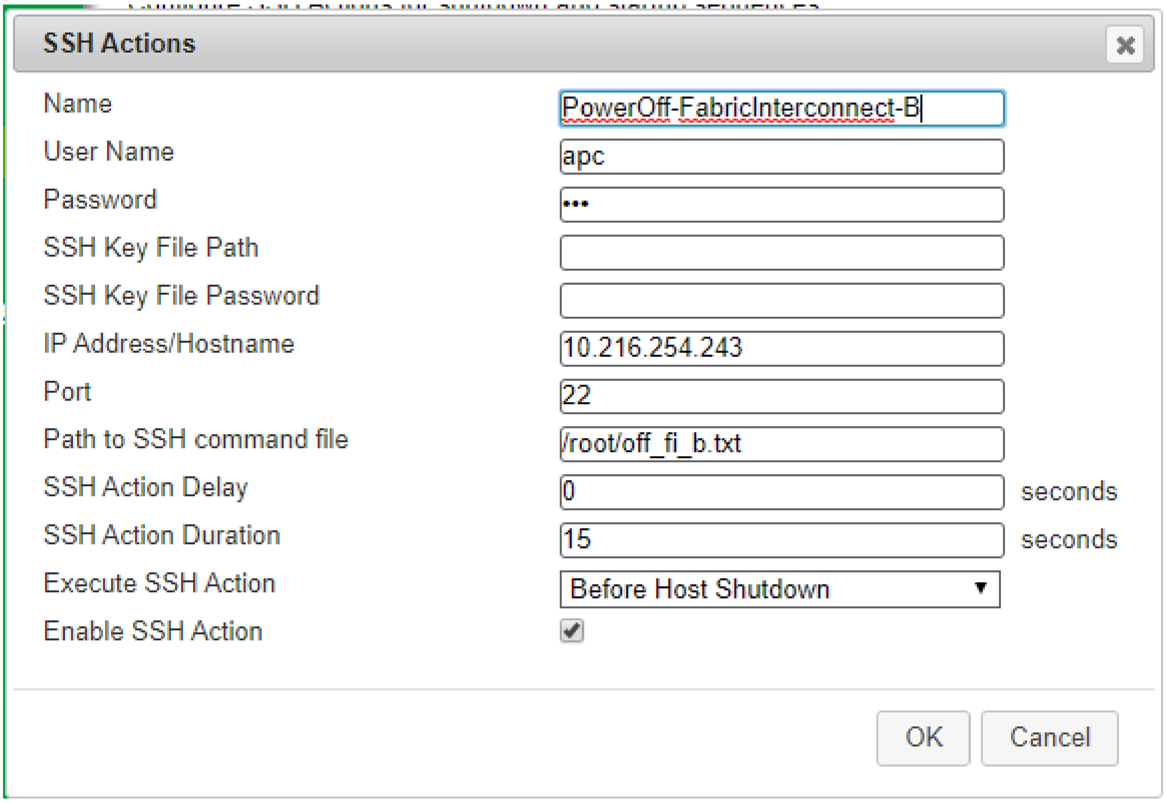
SSH Settings

Configure SSH Actions for shutdown and startup sequences.

SSH Actions

PowerOff-FabricInterconnect-B ✓

PowerOff-FabricInterconect-A ✓



SSH Actions

Name	PowerOff-FabricInterconnect-B	
User Name	apc	
Password	•••	
SSH Key File Path		
SSH Key File Password		
IP Address/Hostname	10.216.254.243	
Port	22	
Path to SSH command file	/root/off_fi_b.txt	
SSH Action Delay	0	seconds
SSH Action Duration	15	seconds
Execute SSH Action	Before Host Shutdown ▼	
Enable SSH Action	☑	

The SSH command file is a text file on the PowerChute machine (in this case it is the PowerChute Appliance, a Linux VM) which contains the CLI command to execute on the PDU:

olOff 2 (olOff <pdu_outlet_number>)

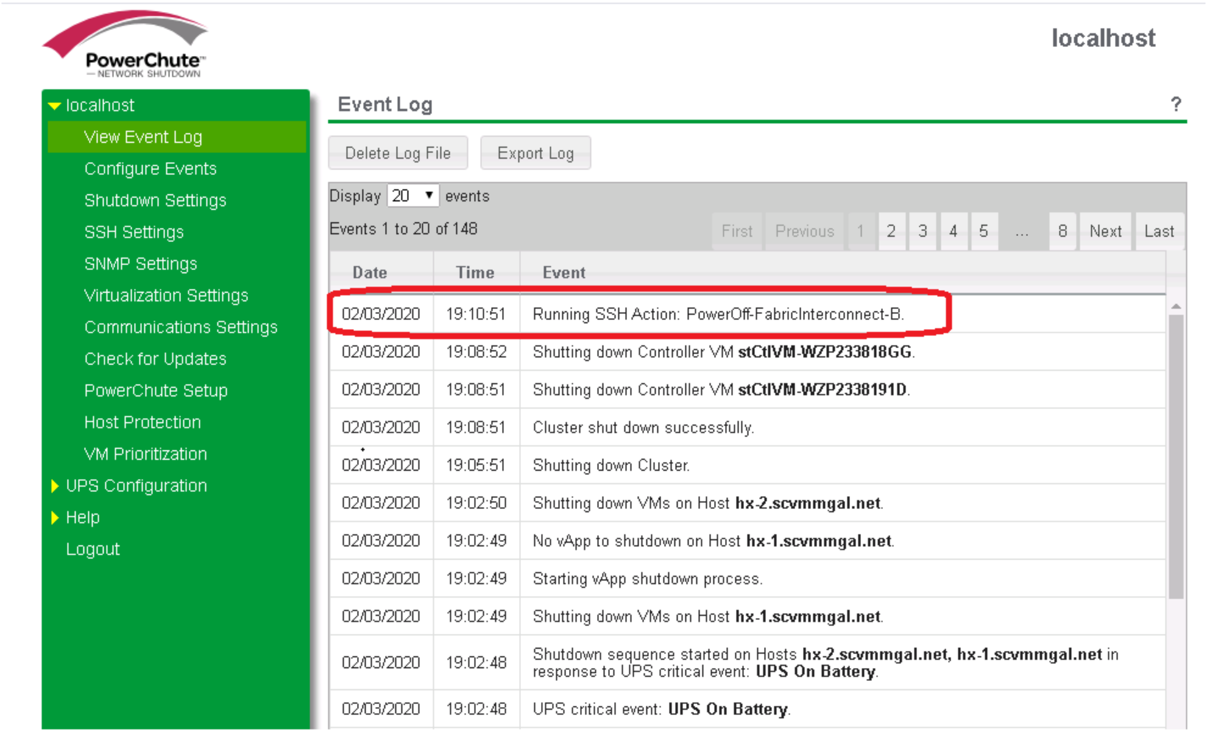
For Fabric Interconnect A, this is **olOff 1**

Since PowerChute is deployed as a VM, the value of "Execute SSH Action" is set to run "Before Host Shutdown". If PowerChute is deployed on a physical windows machine this can be configured to execute after the hosts are shut down instead.

If the Fabric Interconnects need to be powered off after the ESXi machines, then PowerChute must be installed on a physical machine.

The SSH actions will be executed in sequence so the PDU outlet powering Fabric Interconnect B will be powered off and then 15 seconds later (***SSH Action Duration)*** Fabric Interconnect A will be shut down when its PDU outlet is turned off.

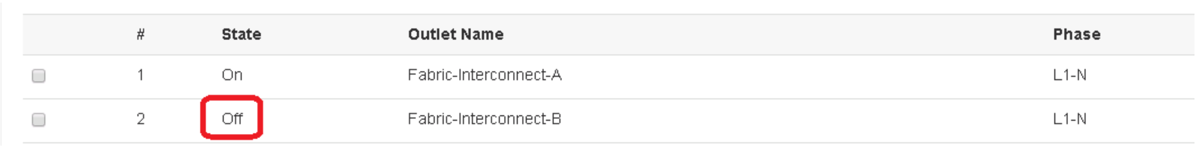
The actions are executed in the order that they are added on the SSH Actions page.



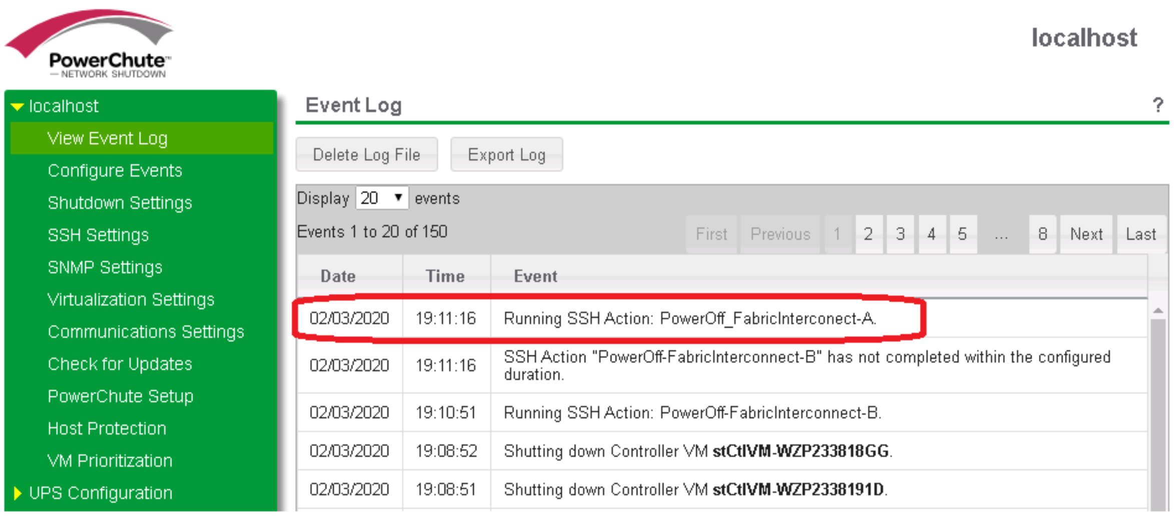
Event Log

Events 1 to 20 of 148

Date	Time	Event
02/03/2020	19:10:51	Running SSH Action: PowerOff-FabricInterconnect-B.
02/03/2020	19:08:52	Shutting down Controller VM **stCtlVM-WZP233818GG**.
02/03/2020	19:08:51	Shutting down Controller VM **stCtlVM-WZP2338191D**.
02/03/2020	19:08:51	Cluster shut down successfully.
02/03/2020	19:05:51	Shutting down Cluster.
02/03/2020	19:02:50	Shutting down VMs on Host **hx-2.scvmmgal.net**.
02/03/2020	19:02:49	No vApp to shutdown on Host **hx-1.scvmmgal.net**.
02/03/2020	19:02:49	Starting vApp shutdown process.
02/03/2020	19:02:49	Shutting down VMs on Host **hx-1.scvmmgal.net**.
02/03/2020	19:02:48	Shutdown sequence started on Hosts **hx-2.scvmmgal.net, hx-1.scvmmgal.net** in response to UPS critical event: **UPS On Battery**.
02/03/2020	19:02:48	UPS critical event: **UPS On Battery**.



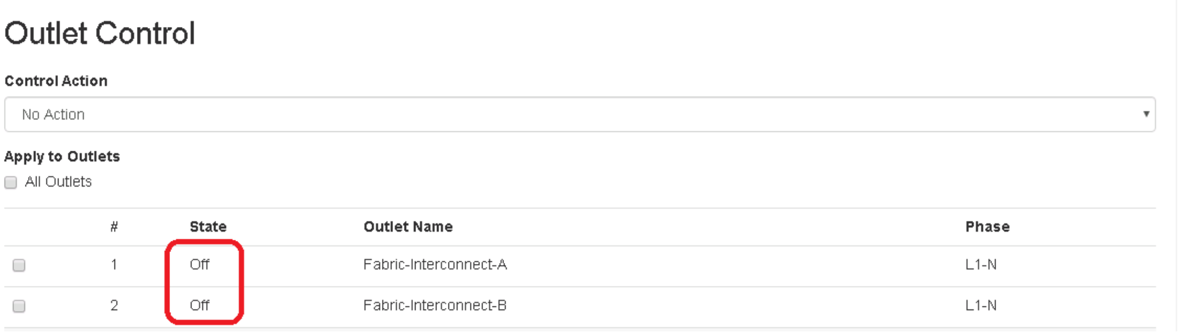
	#	State	Outlet Name	Phase
□	1	On	Fabric-Interconnect-A	L1-N
□	2	Off	Fabric-Interconnect-B	L1-N



Event Log

Events 1 to 20 of 150

Date	Time	Event
02/03/2020	19:11:16	Running SSH Action: PowerOff_FabricInterconnect-A.
02/03/2020	19:11:16	SSH Action "PowerOff-FabricInterconnect-B" has not completed within the configured duration.
02/03/2020	19:10:51	Running SSH Action: PowerOff-FabricInterconnect-B.
02/03/2020	19:08:52	Shutting down Controller VM **stCtlVM-WZP233818GG**.
02/03/2020	19:08:51	Shutting down Controller VM **stCtlVM-WZP2338191D**.



Outlet Control

Control Action

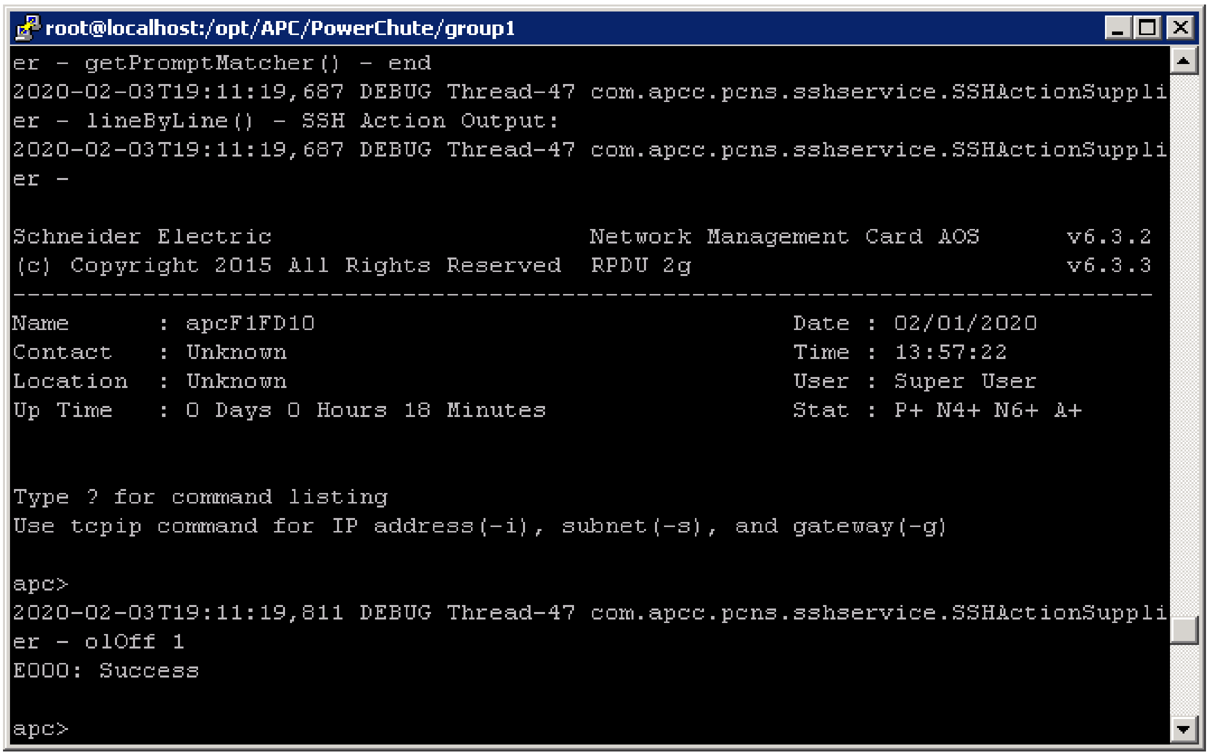
No Action ▾

Apply to Outlets

□ All Outlets

	#	State	Outlet Name	Phase
□	1	Off	Fabric-Interconnect-A	L1-N
□	2	Off	Fabric-Interconnect-B	L1-N

NOTE: PowerChute reports that the action did not complete within the configured duration – the action is executed but the PDU does not return an exit code that can be queried so a timeout is reported:



root@localhost:/opt/APC/PowerChute/group1

```
er - getPromptMatcher() - end
2020-02-03T19:11:19,687 DEBUG Thread-47 com.apcc.pcns.sshservice.SSHActionSuppli
er - lineByLine() - SSH Action Output:
2020-02-03T19:11:19,687 DEBUG Thread-47 com.apcc.pcns.sshservice.SSHActionSuppli
er -

Schneider Electric                       Network Management Card AOS      v6.3.2
(c) Copyright 2015 All Rights Reserved  RPDU 2g                          v6.3.3
-------------------------------------------------------------------------------
Name       : apcF1FD10                                 Date : 02/01/2020
Contact    : Unknown                                   Time : 13:57:22
Location   : Unknown                                   User : Super User
Up Time    : 0 Days 0 Hours 18 Minutes                 Stat : P+ N4+ N6+ A+


Type ? for command listing
Use tcpip command for IP address(-i), subnet(-s), and gateway(-g)

apc>
2020-02-03T19:11:19,811 DEBUG Thread-47 com.apcc.pcns.sshservice.SSHActionSuppli
er - olOff 1
E000: Success

apc>
```

NOTE: Each Fabric Interconnect is powered off sequentially. This approach does not keep track which is the Primary and which is the Secondary.

Re-starting following a shutdown.

There are two solutions provided depending on how PowerChute is deployed. If installed on Windows, a powershell script is used, whereas a linux shell script is used for the PowerChute Virtual Appliance.

A Switched Rack PDU or UPS with Outlet groups can be used to control the order that the Fabric Interconnects and HyperFlex Nodes are powered back on.

For a windows installation, CiscoIMC-PowerTool 1.4.2 is used to power on the ESXi nodes via CIMC when PowerChute machine boots after power is restored.

For the PowerChute Appliance, since it runs as a VM on one of the ESXi hosts, CIMC is not used to start the hosts. In this scenario the BIOS is configured to restore power following an outage to boot the HX nodes.

Powershell Startup Script

CiscoIMC-PowerTool 1.4.2 must be installed and the following variables need to be changed:

$username = "<hx local admin>"

$password = "<hx local admin password>"

$cluster_start_duration = 300 (Time in seconds to allow Cluster start job to complete)

$cvm_start_duration = 120 (Time in seconds to allow CVMs to power on)

$HXHosts = "hx-1.scvmmgal.net","hx-2.scvmmgal.net" (List of HX nodes)

$vsphere_user = "<ESXi admin user>" (This user account must exist on each ESXi host)

$vsphere_password = "<ESXi admin user password>"

$exit_maintenance_duration = 30 (Time allowed for hosts to exit Maintenance mode)

$cimc_user = "<CIMC User>"

$cimc_pwd = "<CIMC_Password>" | ConvertTo-SecureString -AsPlainText -Force

$CIMC_IPS = "10.216.252.247","10.216.252.248"

$exit_maintenance_duration = 30 (Time to bring each host out of Maintenance mode)

$poweron_duration = 420 (Time to wait for hosts to boot before attempting to take them out of maintenance mode)

$vcsa_vm = "HX-VCSA-INT" (In some setups hostname mismatch between vCenter and ESXi can cause issues re-starting the VCSA VM. To avoid this add the VCSA VM name here)

$vcsa_start_duration = 120

Import-Module "**<PATH to CiscoIMC PowerTool>\**CiscoImcPS.psd1"

A command file (StartHX.cmd) is used to run the powershell script above:

cmd.exe /c powershell ; **c:\Scripts_HyperFlex\**start_cluster_win.ps1

Update the path to match the location where you downloaded the script files.

To execute the command file after PowerChute starts, the following values are set in the pcnsconfig.ini file in the group1 folder where PowerChute is installed.

[Events]

event_MonitoringStarted_enableCommandFile = true

event_MonitoringStarted_commandFilePath = C:\ Scripts_HyperFlex \StartHX.cmd

event_MonitoringStarted_commandFileDelay = 60

Linux Shell Script

The script start_hyperflex.sh can be used to take the hosts out of maintenance mode, power on the CVMs followed by the HyperFlex cluster service and the VCSA VM if needed.

To run the shell script on start the pcnsconfig.ini file in /opt/APC/PowerChute/group1 folder can be edited as below:

```
[Events]

event_MonitoringStarted_enableCommandFile = true

event_MonitoringStarted_commandFilePath = /root/start_hyperflex.sh

event_MonitoringStarted_commandFileDelay = 60
```

In the shell script:

```
USER=<user> (HyperFlex Cluster local admin)

PASSWORD=<password> (HyperFlex cluster local admin password)

CLUSTER_IP=<Cluster IP address>

HOST_IP[1]=<ESXi host IP address>

HOST_IP[2]=<ESXi host IP address>

HOST_IP[3]=

HOST_IP[4]=

NUM_CVMS=2 (Enter number to match the number of CVMs/Nodes in the cluster)

CLUSTER_START_TIME=300 (Time allowed for cluster to start before powering on User VMs)

CVM_START_TIME=120 (Time allowed for CVMs to power up before starting Cluster service)

VCSA_VM=<Name of the VCSA Virtual Machine>
```

To allow enough time for the Controller VMs and Cluster services to have started before attempting to start User VMs, the setting below can be used:

```
[HostConfigSettings]

VMware_connect_timeout = 10

VMware_read_timeout = 15

vm_prioritization_enabled = true

delay_after_exit_maintenance_mode = 300
```