Red Hat Enterprise Linux 7 High Availability Add-On Administration

Configuring and Managing the High Availability Add-On

Red Hat Enterprise Linux 7 High Availability Add-On Administration

---

Configuring and Managing the High Availability Add-On

Abstract

High Availability Add-On Administration describes the configuration and management of the High Availability Add-On for Red Hat Enterprise Linux 7.

Table of Contents

Chapter 1. Creating a Red Hat High-Availability Cluster with Pacemaker

This chapter describes the procedure for creating a Red Hat High Availability two-node cluster using **pcs**. After you have created a cluster, you can configure the resources and resource groups that you require.

Configuring the cluster provided in this chapter requires that your system include the following components:

- 2 nodes, which will be used to create the cluster. In this example, the nodes used are **z1.example.com** and **z2.example.com**.
- Network switches for the private network, required for communication among the cluster nodes and other cluster hardware such as network power switches and Fibre Channel switches.
- A power fencing device for each node of the cluster. This example uses two ports of the APC power switch with a host name of **zapc.example.com**.

This chapter is divided into three sections.

- Section 1.1, “Cluster Software Installation” provides the procedure for installing the cluster software.
- Section 1.2, “Cluster Creation” provides the procedure for configuring a two-node cluster.
- Section 1.3, “Fencing Configuration” provides the procedure for configuring fencing devices for each node of the cluster.

1.1. Cluster Software Installation

The procedure for installing and configuring a cluster is as follows.

1. On each node in the cluster, install the Red Hat High Availability Add-On software packages along with all available fence agents from the High Availability channel.

```
# yum install pcs fence-agents-all
```

2. If you are running the **firewalld** daemon, execute the following commands to enable the ports that are required by the Red Hat High Availability Add-On.

Note

You can determine whether the **firewalld** daemon is installed on your system with the **rpm -q firewalld** command. If the **firealld** daemon is installed, you can determine whether it is running with the **firewall-cmd --state** command.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
```

3. In order to use **pcs** to configure the cluster and communicate among the nodes, you must set a password on each node for the user ID **hacluster**, which is the the **pcs** administration account. It is recommended that the password for user **hacluster** be the same on each node.

```
# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

4. Before the cluster can be configured, the **pcsd** daemon must be started and enabled to boot on startup on each node. This daemon works with the **pcs** command to manage configuration across the nodes in the cluster.

   On each node in the cluster, execute the following commands to start the **pcsd** service and to enable **pcsd** at system start.

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

5. Authenticate the **pcs** user **hacluster** for each node in the cluster on the node from which you will be running **pcs**.

   The following command authenticates user **hacluster** on **z1.example.com** for both of the nodes in the example two-node cluster, **z1.example.com** and **z2.example.com**.

```
root@z1 ~]# pcs cluster auth z1.example.com z2.example.com
Username: hacluster
Password:
z1.example.com: Authorized
z2.example.com: Authorized
```

1.2. Cluster Creation

This procedure creates a Red Hat High Availability Add-On cluster that consists of the nodes **z1.example.com** and **z2.example.com**.

1. Execute the following command from **z1.example.com** to create the two-node cluster **mycluster** that consists of nodes **z1.example.com** and **z2.example.com**. This will propagate the cluster configuration files to both nodes in the cluster. This command includes the **--start** option, which will start the cluster services on both nodes in the cluster.

```
[root@z1 ~]# pcs cluster setup --start --name my_cluster \
z1.example.com z2.example.com
z1.example.com: Succeeded
z1.example.com: Starting Cluster...
z2.example.com: Succeeded
z2.example.com: Starting Cluster...
```

2. Enable the cluster services to run on each node in the cluster when the node is booted.

Note

For your particular environment, you may choose to leave the cluster services disabled by skipping this step. This allows you to ensure that if a node goes down, any issues with your cluster or your resources are resolved before the node rejoins the cluster. If you leave the cluster services disabled, you will need to manually start the services when you reboot a node by executing the **pcs cluster start** command on that node.

```
# pcs cluster enable --all
```

You can display the current status of the cluster with the **pcs cluster status** command.

```
[root@z1 ~]# pcs cluster status
Cluster Status:
 Last updated: Thu Jul 25 13:01:26 2013
 Last change: Thu Jul 25 13:04:45 2013 via crmd on z2.example.com
 Stack: corosync
 Current DC: z2.example.com (2) - partition with quorum
 Version: 1.1.10-5.el7-9abe687
 2 Nodes configured
 0 Resources configured
```

1.3. Fencing Configuration

You must configure a fencing device for each node in the cluster. For general information about configuring fencing devices, see the *Red Hat Enterprise Linux 7 High Availability Add-On Reference*.

Note

When configuring a fencing device, you should ensure that your fencing device does not share power with the node that it controls.

This example uses the APC power switch with a host name of **zapc.example.com** to fence the nodes, and it uses the **fence_apc_snmp** fencing agent. Because both nodes will be fenced by the same fencing agent, you can configure both fencing devices as a single resource, using the **pcmk_host_map** and **pcmk_host_list** options.

You create a fencing device by configuring the device as a **stonith** resource with the **pcs stonith create** command. The following command configures a **stonith** resource named **myapc** that uses the **fence_apc_snmp** fencing agent for nodes **z1.example.com** and **z2.example.com**. The **pcmk_host_map** option maps **z1.example.com** to port 1, and **z2.example.com** to port 2. The login value and password for the APC device are both **apc**. By default, this device will use a monitor interval of 60s for each node.

Note that you can use an IP address when specifying the host name for the nodes.

```
[root@z1 ~]# pcs stonith create myapc fence_apc_snmp params \
ipaddr="zapc.example.com" pcmk_host_map="z1.example.com:1;z2.example.com:2" \
pcmk_host_check="static-list" pcmk_host_list="z1.example.com,z2.example.com" \
login="apc" passwd="apc"
```

Note

When you create a **fence_apc_snmp stonith** device, you may see the following warning message, which you can safely ignore:

```
Warning: missing required option(s): 'port, action' for resource type:
stonith:fence_apc_snmp
```

The following command displays the parameters of an existing STONITH device.

```
[root@rh7-1 ~]# pcs stonith show myapc
 Resource: myapc (class=stonith type=fence_apc_snmp)
  Attributes: ipaddr=zapc.example.com
pcmk_host_map=z1.example.com:1;z2.example.com:2 pcmk_host_check=static-list
pcmk_host_list=z1.example.com,z2.example.com login=apc passwd=apc
  Operations: monitor interval=60s (myapc-monitor-interval-60s)
```

Chapter 2. Configuring an Apache Web Server in a Red Hat High Availability Cluster with the pcs Command

This chapter describes how to configure an Apache web server in a two-node Red Hat Enterprise Linux High Availability Add-On cluster using **pcs** to configure cluster resources. In this use case, clients access the Apache web server through a floating IP address. The web server runs on one of two nodes in the cluster. If the node on which the web server is running becomes inoperative, the web server starts up again on the second node of the cluster with minimal service interruption.

Figure 2.1, “Apache Web Server in a Red Hat High Availability Two-Node Cluster” shows a high-level overview of the cluster. The cluster is a two-node Red Hat High Availability cluster which is configured with a network power switch and with shared storage. The cluster nodes are connected to a public network, for client access to the Apache web server through a virtual IP. The Apache server runs on either Node 1 or Node 2, each of which has access to the storage on which the Apache data is kept.

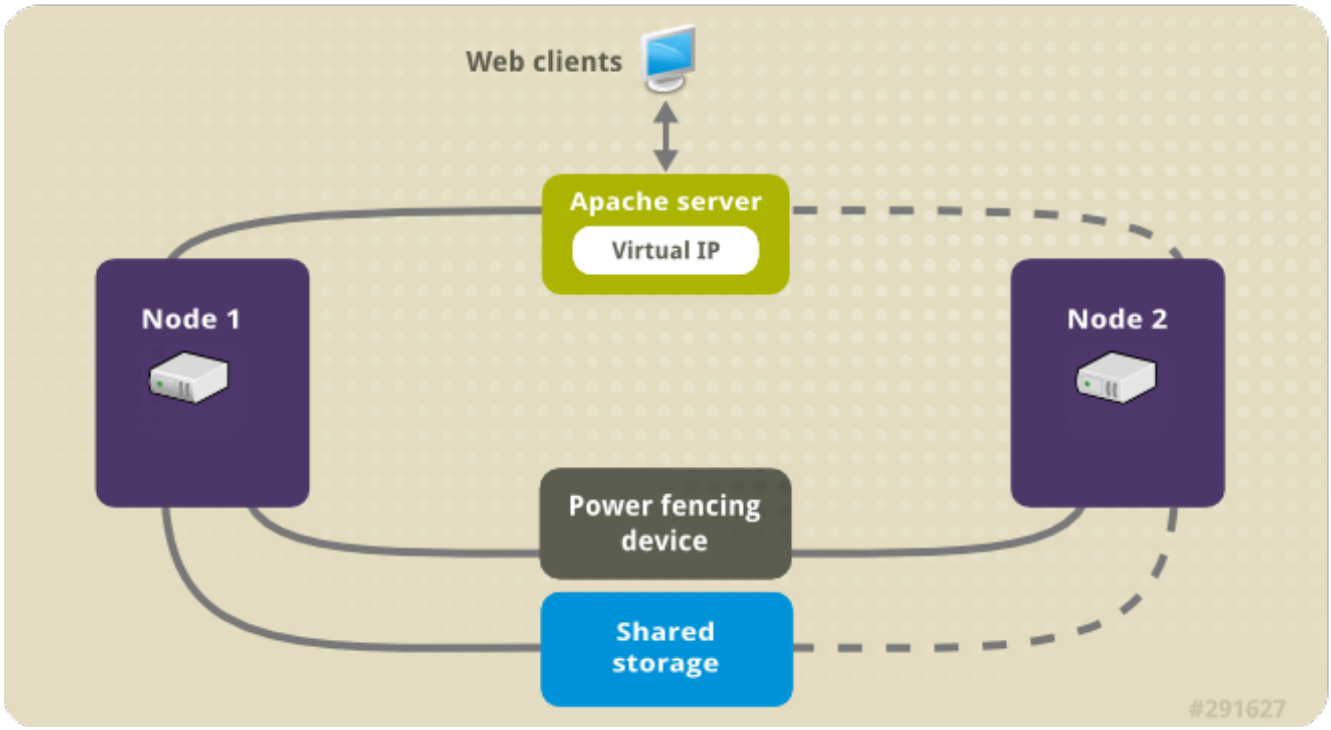

Figure 2.1. Apache Web Server in a Red Hat High Availability Two-Node Cluster

This use case requires that your system include the following components:

- A 2-node Red Hat High Availability cluster with power fencing configured for each node. This procedure uses the cluster example provided in Chapter 1, *Creating a Red Hat High-Availability Cluster with Pacemaker*.
- A public virtual IP address, required for the Apache web server.
- Shared storage for the nodes in the cluster, using iSCSI or Fibre Channel.

The cluster is configured with an Apache resource group, which contains the cluster components that the web server requires: an LVM resource, a file system resource, an IP address resource, and a web server resource. This resource group can fail over from one node of the cluster to the other, allowing either node to run the web server. Before creating the resource group for this cluster, you will perform the following procedures:

1. Configure an **ext4** file system mounted on the logical volume **my_lv**, as described in Section 2.1, "Configuring an LVM Volume with an ext4 File System".
2. Configure a web server, as described in Section 2.2, "Web Server Configuration".
3. Ensure that only the cluster is capable of activating the volume group that contains **my_lv**, and that the volume group will not be activated outside of the cluster on startup, as described in Section 2.3, "Exclusive Activation of a Volume Group in a Cluster".

After performing these procedures, you create the resource group and the resources it contains, as described in Section 2.4, "Creating the Resources and Resource Groups with the pcs Command".

2.1. Configuring an LVM Volume with an ext4 File System

This use case requires that you create an LVM logical volume on storage that is shared between the nodes of the cluster.

The following procedure creates an LVM logical volume and then creates an **ext4** file system on that volume. In this example, the shared partition **/dev/sdb1** is used to store the LVM physical volume from which the LVM logical volume will be created.

Note

LVM volumes and the corresponding partitions and devices used by cluster nodes must be connected to the cluster nodes only.

Since the **/dev/sdb1** partition is storage that is shared, you perform this procedure on one node only,

1. Create an LVM physical volume on partition **/dev/sdb1**.

```
# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
```

2. Create the volume group **my_vg** that consists of the physical volume **/dev/sdb1**.

```
# vgcreate my_vg /dev/sdb1
  Volume group "my_vg" successfully created
```

3. Create a logical volume using the volume group **my_vg**.

```
# lvcreate -L450 -n my_lv my_vg
  Rounding up size to full physical extent 452.00 MiB
  Logical volume "my_lv" created
```

   You can use the **lvs** command to display the logical volume.

```
# lvs
  LV      VG      Attr      LSize   Pool Origin Data%  Move Log Copy%
Convert
  my_lv   my_vg   -wi-a---- 452.00m
  ...
```

4. Create an **ext4** file system on the logical volume **my_lv**.

```
# mkfs.ext4 /dev/my_vg/my_lv
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label=
OS type: Linux
...
```

2.2. Web Server Configuration

The following procedure configures an Apache web server.

1. Ensure that the Apache HTTPD server is installed on each node in the cluster. You also need the **wget** tool installed on the cluster to be able to check the status of the Apache web server.

   On each node, execute the following command.

   ```
   # yum install -y httpd wget
   ```

2. In order for the Apache resource agent to get the status of the Apache web server, ensure that the following text is present in the **/etc/httpd/conf/httpd.conf** file on each node in the cluster, and ensure that it has not been commented out. If this text is not already present, add the text to the end of the file.

   ```
   <Location /server-status>
     SetHandler server-status
     Order deny,allow
     Deny from all
     Allow from 127.0.0.1
   </Location>
   ```

3. Create a web page for Apache to serve up. On one node in the cluster, mount the file system you created in Section 2.1, “Configuring an LVM Volume with an ext4 File System”, create the file **index.html** on that file system, then unmount the file system.

   ```
   # mount /dev/my_vg/my_lv /var/www/
   # mkdir /var/www/html
   # mkdir /var/www/cgi-bin
   # mkdir /var/www/error
   # restorecon -R /var/www
   # cat <<-END >/var/www/html/index.html
   <html>
   <body>Hello</body>
   </html>
   END
   # umount /var/www
   ```

2.3. Exclusive Activation of a Volume Group in a Cluster

The following procedure configures the volume group in a way that will ensure that only the cluster is capable of activating the volume group, and that the volume group will not be activated outside of the cluster on startup. If the volume group is activated by a system outside of the cluster, there is a risk of corrupting the volume group's metadata.

This procedure modifies the **volume_list** entry in the **/etc/lvm/lvm.conf** configuration file. Volume groups listed in the **volume_list** entry are allowed to automatically activate on the local node outside of the cluster manager's control. Volume groups related to the node's local root and home directories should be included in this list. All volume groups managed by the cluster manager must be excluded from the **volume_list** entry. Note that this procedure does not require the use of **clvmd**.

Perform the following procedure on each node in the cluster.

1. Determine which volume groups are currently configured on your local storage with the following command. This will output a list of the currently-configured volume groups. If you have space allocated in separate volume groups for root and for your home directory on this node, you will see those volumes in the output, as in this example.

```
# vgs --noheadings -o vg_name
  my_vg
  rhel_home
  rhel_root
```

2. Add the volume groups other than **my_vg** (the volume group you have just defined for the cluster) as entries to **volume_list** in the **/etc/lvm/lvm.conf** configuration file. For example, if you have space allocated in separate volume groups for root and for your home directory, you would uncomment the **volume_list** line of the **lvm.conf** file and add these volume groups as entries to **volume_list** as follows:

```
volume_list = [ "rhel_root", "rhel_home" ]
```

Note

If no local volume groups are present on a node to be activated outside of the cluster manager, you must still initialize the **volume_list** entry as **volume_list = []**.

3. Rebuild the **initrd** boot image to guarantee that the boot image will not try to activate a volume group controlled by the cluster. Update the **initrd** device cluster with the following command. This command may take up to a minute to complete.

```
# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

4. Reboot the node.

Note

If you have installed a new Linux kernel since booting the node on which you created the boot image, the new **initrd** image will be for the kernel that was running when you created it and not for the new kernel that is running when you reboot the node. You can ensure that the correct **initrd** device is in use by running the **uname -r** command before and after the reboot to determine the kernel release that is running. If the releases are not the same, update the **initrd** file after rebooting with the new kernel and then reboot the node.

5. When the node has rebooted, check whether the cluster services have started up again on that node by executing the **pcs cluster status** command on that node. If this yields the message **Error: cluster is not currently running on this node** then run the following command.

```
# pcs cluster start
```

Alternately, you can wait until you have rebooted each node in the cluster and start cluster services on each of the nodes with the following command.

```
# pcs cluster start --all
```

2.4. Creating the Resources and Resource Groups with the pcs Command

This use case requires that you create four cluster resources. To ensure these resources all run on the same node, they are configured as part of the resource group **apachegroup**. The resources to create are as follows, listed in the order in which they will start.

1. An **LVM** resource named **my_lvm** that uses the LVM volume group you created in Section 2.1, “Configuring an LVM Volume with an ext4 File System”.
2. A **Filesystem** resource named **my_fs**, that uses the filesystem device **/dev/my_vg/my_lv** you created in Section 2.1, “Configuring an LVM Volume with an ext4 File System”.
3. An **IPaddr2** resource, which is a floating IP address for the **apachegroup** resource group. The IP address must not be one already associated with a physical node. If the **IPaddr2** resource's NIC device is not specified, the floating IP must reside on the same network as the statically assigned IP addresses used by the cluster nodes, otherwise the NIC device to assign the floating IP address can not be properly detected.
4. An **apache** resource named **Website** that uses the **index.html** file and the Apache configuration you defined in Section 2.2, “Web Server Configuration”.

The following procedure creates the resource group **apachegroup** and the resources that the group contains. The resources will start in the order in which you add them to the group, and they will stop in the reverse order in which they are added to the group. Run this procedure from one node of the cluster only.

1. The following command creates the LVM resource **my_lvm**. This command specifies the **exclusive=true** parameter to ensure that only the cluster is capable of activating the LVM logical volume. Because the resource group **apachegroup** does not yet exist, this command creates the resource group.

```
[root@z1 ~]# pcs resource create my_lvm LVM volgrpname=my_vg \
exclusive=true --group apachegroup
```

When you create a resource, the resource is started automatically. You can use the following command to confirm that the resource was created and has started.

```
# pcs resource show
 Resource Group: apachegroup
     my_lvm (ocf::heartbeat:LVM): Started
```

You can manually stop and start an individual resource with the **pcs resource disable** and **pcs resource enable** commands.

2. The following commands create the remaining resources for the configuration, adding them to the existing resource group **apachegroup**.

```
[root@z1 ~]# pcs resource create my_fs Filesystem \
device="/dev/my_vg/my_lv" directory="/var/www" fstype="ext4" --group \
apachegroup

[root@z1 ~]# pcs resource create VirtualIP IPaddr2 ip=198.51.100.3 \
cidr_netmask=24 --group apachegroup

[root@z1 ~]# pcs resource create Website apache \
configfile="/etc/httpd/conf/httpd.conf" \
statusurl="http://127.0.0.1/server-status" --group apachegroup
```

3. After creating the resources and the resource group that contains them, you can check the status of the cluster. Note that all four resources are running on the same node.

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 16:38:51 2013
Last change: Wed Jul 31 16:42:14 2013 via crm_attribute on z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured


Online: [ z1.example.com z2.example.com ]

Full list of resources:
 myapc (stonith:fence_apc_snmp): Started z1.example.com
 Resource Group: apachegroup
     my_lvm (ocf::heartbeat:LVM): Started z1.example.com
     my_fs (ocf::heartbeat:Filesystem): Started z1.example.com
     VirtualIP (ocf::heartbeat:IPaddr2): Started z1.example.com
     Website (ocf::heartbeat:apache): Started z1.example.com
```

Note that if you have not configured a fencing device for your cluster, as described in Section 1.3, “Fencing Configuration”, by default the resources do not start.

4. Once the cluster is up and running, you can point a browser to the IP address you defined as the **IPaddr2** resource to view the sample display, consisting of the simple word "Hello".

```
Hello
```

If you find that the resources you configured are not running, you can run the **pcs resource debug-start *resource*** command to test the resource configuration. For information on the **pcs resource debug-start** command, see the *High Availability Add-On Reference* manual.

2.5. Testing the Resource Configuration

In the cluster status display shown in Section 2.4, "Creating the Resources and Resource Groups with the pcs Command", all of the resources are running on node **z1.example.com**. You can test whether the resource group fails over to node **z2.example.com** by using the following procedure to put the first node in **standby** mode, after which the node will no longer be able to host resources.

1. The following command puts node **z1.example.com** in **standby** mode.

```
root@z1 ~]# pcs cluster standby z1.example.com
```

2. After putting node **z1** in standby mode, check the cluster status. Note that the resources should now all be running on **z2**.

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 17:16:17 2013
Last change: Wed Jul 31 17:18:34 2013 via crm_attribute on z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured


Node z1.example.com (1): standby
Online: [ z2.example.com ]

Full list of resources:

 myapc (stonith:fence_apc_snmp): Started z1.example.com
 Resource Group: apachegroup
     my_lvm (ocf::heartbeat:LVM): Started z2.example.com
     my_fs (ocf::heartbeat:Filesystem): Started z2.example.com
     VirtualIP (ocf::heartbeat:IPaddr2): Started z2.example.com
     Website (ocf::heartbeat:apache): Started z2.example.com
```

   The web site at the defined IP address should still display, without interruption.

3. To remove **z1** from **standby** mode, run the following command.

```
root@z1 ~]# pcs cluster unstandby z1.example.com
```

Note

Removing a node from **standby** mode does not in itself cause the resources to fail back over to that node. For information on controlling which node resources can run on, see the chapter on configuring cluster resources in the *Red Hat High Availability Add-On Reference*.

2.6. Cluster pcs Command Summary

For a quick summary of the cluster configuration procedure, this section provides a listing of the **pcs** commands for this use case that create the Apache web server in a cluster, including the configuration commands that created the cluster itself.

After you have set a password for user **hacluster** on both nodes and and started the **pcsd** service, the commands to create the cluster and configure fencing for the cluster nodes are as follows.

```
[root@z1 ~]# pcs cluster auth z1.example.com z2.example.com

[root@z1 ~]# pcs cluster setup --start --name my_cluster z1.example.com \
z2.example.com

[root@z1 ~]# pcs stonith create myapc fence_apc_snmp params \
ipaddr="zapc.example.com" pcmk_host_map="z1.example.com:1;z2.example.com:2" \
pcmk_host_check="static-list" pcmk_host_list="z1.example.com,z2.example.com" \
login="apc" passwd="apc"
```

Note

When you create a **fence_apc_snmp stonith** device, you may see the following warning message, which you can safely ignore:

```
Warning: missing required option(s): 'port, action' for resource type:
stonith:fence_apc_snmp
```

After you have set up the initial LVM volume and Apache web server, the following commands configure the resources and resource groups for the cluster.

```
[root@z1 ~]# pcs resource create my_lvm LVM volgrpname=my_vg exclusive=true \
--group apachegroup

[root@z1 ~]# pcs resource create my_fs Filesystem device="/dev/my_vg/my_lv" \
directory="/var/www" fstype="ext4" --group apachegroup

[root@z1 ~]# pcs resource create VirtualIP IPaddr2 ip=198.51.100.3 \
cidr_netmask=24 --group apachegroup

[root@z1 ~]# pcs resource create Website apache \
configfile="/etc/httpd/conf/httpd.conf" \
statusurl="http://127.0.0.1/server-status" --group apachegroup
```

Revision History

Revision 0.1-33 **Mon Jun 2 2014** **Steven Levine**

Version for 7.0 GA release

Revision 0.1-31 **Wed May 21 2014** **Steven Levine**

Resolves: #886235

Document volume_list usage

Revision 0.1-29 **Tue May 20 2014** **Steven Levine**

Rebuild for style changes and updated draft

Revision 0.1-20 **Wed Apr 9 2014** **Steven Levine**

Updated Beta draft

Revision 0.1-8 **Fri Dec 6 2013** **Steven Levine**

Beta draft

Revision 0.0-1 **Wed Jan 16 2013** **Steven Levine**

First version for Red Hat Enterprise Linux 7