Technical white paper

Building a Highly-Available ArcSight SmartConnector Cluster with Pacemaker

HA Operation of both Passive and Active ArcSight SmartConnectors

Allen Pomeroy, HP ESP Solution Architect

White paper version 2.0.6, Cluster configuration 2.0.6

Contents

Cost Effective SmartConnector HA

This paper describes the use of open source clustering software used to build a low-cost, reliable, high availability environment on CentOS Linux in which to run both passive and active SmartConnectors, providing automated failure recovery.

Introduction

At current time there is no inherent High-Availability capability for ArcSight SmartConnector installations other than HA management of connectors through multiple Connector Appliances. Once events have been acquired by a SmartConnector, the store-and-forward architecture provides a reliable event handling ecosystem, but the problem is what to do when a specific SmartConnector, or the system it is running on, fails. Traditionally customers would procure and employ hardware load balancers in front of SmartConnector Connector Appliances or Connector Concentrators, although that only really deals with passive connectors, such as syslog, SNMP or other listeners. Active connectors such as Windows, Database readers, etc would require a manual failure recovery in order to restore the service of event collection. Although customers can use commercial clustering technology, such as Veritas Cluster Server, those tools can require substantial capital investment. This paper describes the use of open source clustering software used to build a low-cost, reliable, high availability environment in which to run both passive and active SmartConnectors, providing active failure recovery and service continuance. This configuration is not endorsed or supported by HP Enterprise Security Products and is provided for informational purposes only.

This package includes documentation and scripts to setup a cluster from scratch in an automated manner. Access to cluster packages in CentOS or local customer provided repositories is needed by the setup scripts. Users of this package need to obtain a Linux binary of the HP ArcSight SmartConnector software – it is not included. The result of the included quickstart script will be a functional cluster with a syslog SmartConnector running and able to fail-over to a partner node in the case of primary node failure. The two cluster nodes must have at least two (2) network segments, although all traffic to/from the event sources can be on any customer network that is reachable via standard IPv4 routing – the cluster does not operate in-line but rather as a distinct IP node on the customer network.

Assuming a relatively fast connection to the Internet, or internal servers, for access to the CentOS software repositories, the quickstart script can complete the cluster setup in less than 15 minutes, but one should expect to take a day to review the cluster configuration, commands and proper operating procedures. Recovery from incorrect cluster commands or operations will almost assuredly require a cluster outage for re-configuration, resync or worse, backup/recovery. Given the relative low cost of simple 1U servers, it is strongly recommended that two pairs of nodes are used to create a test cluster and production cluster. Modest VMware or other virtual servers can be used to implement the test environment. TCP/UDP protocol ports that are used are specific to the unique cluster IP addresses, so there should not be any collisions – although care must be taken to choose unique multicast addresses for the cluster communication provided by corosync. This is not done automatically by the quickstart scripts.

Feed back is welcomed, both success stories and problems/bugs that are encountered, but *users need to self-support any implementations*. The current maintainer is

Allen Pomeroy allen.pomeroy@hp.com

Required Components

This solution was installed with the following components:

- Two (2) 64-bit Quad Core Intel servers with 4GB memory
- Two (2) internal disks, one for the OS and one for exclusive use of the cluster (20GB + 2GB)
- Quad Ethernet interface – dual network interfaces would work but channel bonding would not be possible
- CentOS 6.4 64-bit OS
- Corosync 1.4.1 cluster engine
- DRBD 8.3.15 network disk management
- Pacemaker 1.1.8 cluster manager
- PCS 0.9.26 cluster resource management tool
- HP ArcSight syslog Daemon SmartConnector 6.0.5

Any server hardware that is appropriate to host the software SmartConnectors would work for providing the clustering functions, since the clustering components add negligible overhead on server CPU, memory and network resources. Although the best configuration would be to use at least three (3) cluster members, this paper shows how to implement a cluster with only two (2) nodes. For a complete production quality solution, the customer should add cluster member fencing devices used to ensure no cluster split-brain conditions where both nodes try to be the primary at the same time. These are called STONITH solutions and will not be covered in this version of this paper. There needs to be multiple network connections between the servers to lessen the possibility of the split-brain condition by providing multiple heart-beat paths.

This configuration requires:

- Exactly two nodes (future papers will address node count > 2)
- Both nodes have at least two network interfaces: eth0, eth1 (bonding not tested yet)
- Both nodes have a *dedicated raw disk partition* for the cluster. Can be on the same disk as the OS if needed.
- Dedicated static IPv4 addresses for each network interface plus a VIP for the cluster, two separate network segments. Best would be at least one cross-over cable connecting the internal network interfaces, although a management network switch connection would suffice.
- Root public keys for authentication, enabling both nodes to execute commands on the partner as root with no password. These will be created by the quickstart script.
- Both cluster nodes have Internet access to CentOS software sites, or you have manually modified the YUM repositories to use local/internal repositories that have the required packages.
- User of this tool MUST already be a valid HP ArcSight user since the Linux SmartConnector binaries are not included in this package.

Limitations and Assumptions

This version of this paper shows how to implement an Active/Passive cluster with no other components other than those listed in the Required Components section. Specifically, no network based or shared disk is required – the solution can be implemented with two basic servers.

The user is assumed to have administrator level knowledge of IPv4 networking, Linux OS and basic knowledge of cluster concepts. See linux-ha.org and clusterlabs.org for additional clustering information.

Known limitations of this solution include:

- Servers are configured in an Active/Passive configuration. It is possible to configure for Active/Active where both servers are running SmartConnectors but there is no substantial advantage since the maximum loading configuration must not exceed 100/0 or 50/50 (with the total load 50% of the available capacity to ensure one server can carry the entire load during a fail-over condition. Future versions of this paper will allow Active/Active configurations.

- No GUI is used to setup or control the cluster components, however some command line scripts are provided to ease the common start, stop and failover tasks. A web interface is provided to display the cluster operational status by opening a browser and pointing to

  `http://{clusterip}/status.html`

- SmartConnectors are installed as a single instance with a VIP and filesystem that follow it to the currently active cluster node, still requiring an outage for SmartConnector maintenance (such as upgrades). It is possible to install multiple instances of a connector and internally load-balance similar connectors to allow rolling upgrades via ldirectord load balancing functions, however that will be added in a future version of this paper.

- SmartConnector performance should not be negatively impacted by use of the DRBD based filesystem since normal operation of connectors requires little disk I/O. In the case where the connector needs to cache events, the effective performance of the DRBD filesystem is unknown, but is sure to be lower than direct attached local disk.

- No guidance is given on how to upgrade the cluster components, although upgrade of the underlaying OS should be possible by ejecting the node from the cluster, applying the updates and re-inserting the node again. It is a good idea to have a duplicate configuration of the cluster in a test/lab environment to prove out changes before applying to production.

- STONITH capability is disabled since no way to test this capability existed during development of this paper. This introduces risk of production downtime due to conditions that can lead to split-brain in the cluster.

Attention

Many desirable security features available in CentOS 6.4 are disabled. SELinux and the iptables firewall are disabled. A future version of this paper will document required ports and recommend appropriate iptables firewall rules. *It is assumed this cluster configuration will be deployed in a secure network segment.*

System Installation and Configuration

Perform the following OS installation and configuration steps to get the nodes ready for cluster installation.

Install and Configure CentOS 6.4 64-bit Operating System

Install and configure CentOS 6.4 OS on both systems *in an identical manner* (except IP addresses and root passwords)

1. The default "Minimal Server" configuration was used.

2. Use custom disk layout options to reserve a plain Linux partition for the connectors that has enough cache space for the SmartConnectors to use to prevent any event loss in the case of the ArcSight nodes upstream become unavailable – use at least serveral GB. The sample configuration in this paper uses /dev/sda1,2 for the OS and /dev/sda3 for the DRBD connector disk.

Partition	Type, Size, Mount point
/dev/sda1	Linux (0x83), 512MB, /boot (ext4)
/dev/sda2	Linux LVM (0x8e), 19GB
/dev/vg_ca1/LogVol00	18GB, /root (ext4)
/dev/vg_ca1/LogVol01	1GB, swap
/dev/sda3	Linux (0x83), 1GB, no filesystem created

Note

Do NOT create a filesystem on the partition to be used for DRBD. The quickstart script will try to zero enough of any remenant signatures, but DRBD is VERY sensitive to detecting existing use, such as filesystems. Best to leave the partition as free space while the OS install occurs, then use fdisk to create the DRBD partition AFTER the OS install is complete to guarantee it has no signature.

3. Configure networking for both network interfaces – use manual IPv4 settings and only set DNS servers and default gateway on the 'production' or main interface. The second interface could (and probably should) be connected via a cross-over cable.

Although a graphical environment will make system configuration and management easier, the cluster components do not require it – the minimal configuration will not install a GUI.

Initial Configuration Steps Required

After the initial OS install is complete, there are some manual steps that are required to support the quickstart script. These steps are mandatory for successful setup.

1. As covered above, create DRBD partition on both nodes.

2. Ensure the date and time is correct, manually set if needed.

```
date MMDDhhmmYYYY
```

3. Install OpenSSH clients – CentOS Minimal does not include SCP binaries that are mandatory for the quickstart script.

```
yum install openssh-clients
```

4. FTP or SCP the Cluster Resource Manager setup and HP ArcSight Linux SmartConnector packages to node1. During this cluster setup, all commands are run from node1 unless specifically noted. Other than running status check commands on node2, it is advised to let the cluster setup script do all changes to node2.

Quickstart Instructions

Follow these steps to use the crm package cluster-quickstart.shl script and get the cluster running in 30 minutes or less.

1. Follow the System Installation and Configuration instructions to get CentOS to a suitable point for cluster installation.

2. Transfer the Quickstart tar file to node1 and extract. This will create a "crm" subdirectory with all scripts and configuration files needed.

```
tar xf crm-2.0.6.tar
cd crm
```

3. Gather all the relevant cluster node information (IPv4 addresses for interfaces and VIP, IPv4 multicast address for the cluster, nodenames, Logger IPv4 address and receiver name, etc). See the .clusterinfo.sample file for all the information that is mandatory for the installation quickstart script. It is recommended to read the detailed section of this whitepaper prior to running the quickstart script to understand what actions the script takes and what to expect.

Caution

Ensure the partition used for DRBD does not contain a mounted filesystem or any other data you want to keep. It will be zeroed out and overwritten with *no backup or way to undo the DRBD setup.*

---

4. Configure the two servers as described in the requirements above and ensure you have a dedicated disk partition on each node with no file system setup on it (eg. /dev/sda3 or /dev/sdb1). Make sure there is no data on that partition that you need to keep - it WILL be OVERWRITTEN with NO BACKUP. Chose a size for the partition that is sufficient to store multiple copies of the connectors and any event cache. Keep in mind these partitions will be synchronized over the internal network, so the larger the partition, the longer the initial sync will take. Note the cluster will NOT be able to fail over until the synchronization of partitions is complete.

5. In the "crm" subdirectory, copy either the .clusterinfo.template or .clusterinfo.sample to .clusterinfo , then edit with all the cluster details. Ensure it is executable.

```
cp .clusterinfo.sample .clusterinfo
vi .clusterinfo
chmod u+x .clusterinfo
```

6. Run the "cluster-quickstart.shl" script, paying close attention to the prompts. If a section fails, select Quit and fix the problem(s) encountered, then re-run the cluster-quickstart.shl script. Skip the sections that have already been setup successfully. ** Every section MUST complete successfully before the following sections will work **

```
[root@ca1 crm]# ./cluster-quickstart-2.0.6.shl


============================================================
Cluster Quickstart - version 2.0.6 - starting at Mon Sep 16 08:01:16 CDT 2013

                    ** WARNING ** WARNING ** WARNING **
   This cluster setup script WILL modify settings on your systems including
   disabling security features such as SElinux and IPtables.  Review cluster
   settings carefully, since the disk partition specified (in the .clusterinfo
   defaults file and/or variable setting) WILL BE OVERWRITTEN with no backup.
   All data currently on that partition on BOTH nodes WILL BE DESTROYED -
```

```
   be sure it exists and is backed up before you run this script.

   This install process is intended to be run on the first cluster node ONLY.
   Do NOT run any setup commands on the second cluster node unless explicitly
   instructed to do so.

 You can select one of three options at the start of each step:
 Proceed - Perform the commands in the step
 Skip - Skip the step and go onto the next
 Quit - Immediately stop and exit the quickstart script

Do you want to proceed with cluster quickstart?
Proceed/Skip/Quit? [p]/s/q:
```

7. For near real-time cluster status, either run "crm_mon" at a command line or open a browser and hit http://<clusterIP>/status.html after the cluster is running.

```
crm_mon
```

To get a snapshot of the cluster resources, use “pcs status”.

```
pcs status

Last updated: Mon Sep 16 08:46:23 2013
Last change: Mon Sep 16 08:44:28 2013 via crm_attribute on ca1
Stack: classic openais (with plugin)
Current DC: ca2 - partition with quorum
Version: 1.1.8-7.el6-394e906
2 Nodes configured, 2 expected votes
8 Resources configured.


Online: [ ca1 ca2 ]

Full list of resources:

 Master/Slave Set: ClusterDataClone [ClusterData]
     Masters: [ ca1 ]
     Slaves: [ ca2 ]
 ClusterFS      (ocf::heartbeat:Filesystem):    Started ca1
 ClusterIP      (ocf::heartbeat:IPaddr2):       Started ca1
 ClusterSrcIP   (ocf::heartbeat:IPsrcaddr):     Started ca1
 WebServer      (ocf::heartbeat:apache):        Started ca1
 ClusterStatus  (ocf::pacemaker:ClusterMon):    Started ca1
 SyslogUdp1     (ocf::arcsight:SmartConnector): Started ca1
```

8. At this point, the SyslogUdp1 SmartConnector will be running, reading events on the syslog port you've specified, sending CEF data to the Logger ‘SmartMessage Receiver’ that you've specified.

9. The safest way to force the SmartConnectors to move to the partner node is place the current node into standby mode. This will force all resources to move to the surviving partner node.

```
pcs cluster standby {node1}
```

This will gracefully move everything to the partner node. Give that time to stablize (1-5 minutes), then run

```
pcs cluster unstandby {node1}
```

to give control of the resources back to the cluster policy engine. Until there is a failure of node2, the resources will stay on node2 (the unstandby will not trigger a failback due to configured resource "stickiness" of the connector resources)

To stop and subsequently start the SmartConnector while the cluster is still running, use the resource command with stop or start arguments

```
pcs resource stop SyslogUdp1
```

10. Boot of the nodes should automatically start the corosync and pacemaker base components, after which the cluster resources will select a node as primary and start.

Cluster Installation Details

The following describes the actions that are taken in the cluster quickstart script and are not needed if the quickstart script is used. These details are only provided to help the user gain an understanding of the components and configuration of the cluster.

Note

All commands are run from node1 (primary) in the cluster unless specifically noted otherwise.

Configure OS

Build up /etc/hosts files with values specified in the `.clusterinfo` configuration file and distribute to both nodes of the cluster. Ensure to add all cluster IP addresses and IP addresses of interest – do not rely on DNS.

```
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6
#
# cluster nodes - physical interfaces
# ===================================
# node 1 - node1
10.20.1.61       node1.acme.com node1
172.16.100.61    node1-mgmt.acme.com node1-mgmt
#
# node 2 - node2
10.20.1.62       node2.acme.com node2
172.16.100.62    node2-mgmt.acme.com node2-mgmt
#
# cluster VIPs
# ============
10.20.1.60       ca
#
# ArcSight destinations
10.20.1.131      esm60-lab
10.20.1.132      esm60-prod
```

Caution

Generating and using SSH keys for the root user that contain blank pass-phrases and adding to authorized_keys files allows full root access to either cluster node from the other node(s). It is advisable to provide strong controls around the root accounts, such as forcing named user login before use of the root account.

Generate root ssh keys on both nodes of the cluster and exchange with both nodes, allowing the root user on either cluster member to execute commands on either node. Although this is not strictly needed for functioning of the clustering software, it makes configuration and operation much easier.

```
node1# ssh-keygen -t rsa -b 2048
node1# scp id_rsa.pub root@node2:.ssh/id_rsa.pub.node1.root
node2# cd .ssh
node2# cat id_rsa.pub.node1.root >> authorized_keys
(repeat for node2)
```

Disable network auto-configure services

```
vi /etc/sysconfig/network: NOZEROCONF=yes
service network restart
```

Update System Security Settings

Caution

Disabling SELinux and the iptables firewall will make both cluster nodes less secure and possibly enable a compromise. It is highly recommended to implement the cluster on a secure network segment.

Disable selinux and IPtables, then reboot

```
vi /etc/sysconfig/selinux: SELINUX=disabled
chkconfig iptables off
service iptables stop
```

Setup Network Interfaces

Configure network interfaces with required static settings. Ensure hostname is a short name (do not use FQDN). Disable auto configuration network packages with the NOZEROCONF flag (as shown above) or the IPsrcaddr cluster Resource Agent will likely fail.

```
/etc/sysconfig/network
NETWORKING=yes
HOSTNAME=node1
NOZEROCONF=yes

/etc/sysconfig/network-scripts/ifcfg-eth0
UUID="ec582324-6dea-4032-8eb6-beeaea5269ef"
NM_CONTROLLED="no"
HWADDR="00:0D:35:42:9E:6A"
BOOTPROTO="static"
DEVICE="eth0"
ONBOOT="yes"
IPADDR=172.16.100.61
NETMASK=255.255.255.0
NETWORK=172.16.100.0
IPV6INIT=no

/etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE="eth1"
UUID="2e84b4bb-4eb4-4f40-bed6-a17cafff3d2a"
NM_CONTROLLED="no"
BOOTPROTO="static"
HWADDR="00:0D:35:73:2E:69"
ONBOOT="yes"
IPADDR=10.20.1.61
NETMASK=255.255.255.0
NETWORK=10.20.1.0
GATEWAY=10.20.1.1
DNS1=10.20.1.251
DOMAIN=acme.com
IPV6INIT=no
```

Perform Package Updates

On both nodes, perform a package update to ensure all the latest patches and drivers

```
yum update
```

Install Cluster Packages

Install corosync, Pacemaker and DRBD packages – the DRBD packages come from a third party repository. These three packages together form the High Availability clustering capability that monitors member health, synchronizes a shared disk partition supporting a cluster filesystem, and cluster resource management (virtual IP, SmartConnectors, etc). Ensure these packages are installed on both cluster nodes.

- Corosync provides the heart beat and cluster messaging layer between the cluster nodes,
- pacemaker provides the cluster resource management capabilities, and
- DRBD provides a RAID1 network based disk volume on which we will run a connector filesystem.

```
rpm -ivh http://elrepo.org/elrepo-release-6-5.el6.elrepo.noarch.rpm

yum install -y ntp pacemaker corosync pcs httpd wget

yum --enablerepo=elrepo install -y kmod-drbd83 drbd83-utils
```

Install 32-bit compatibilities libraries to support the 32-bit SmartConnectors.

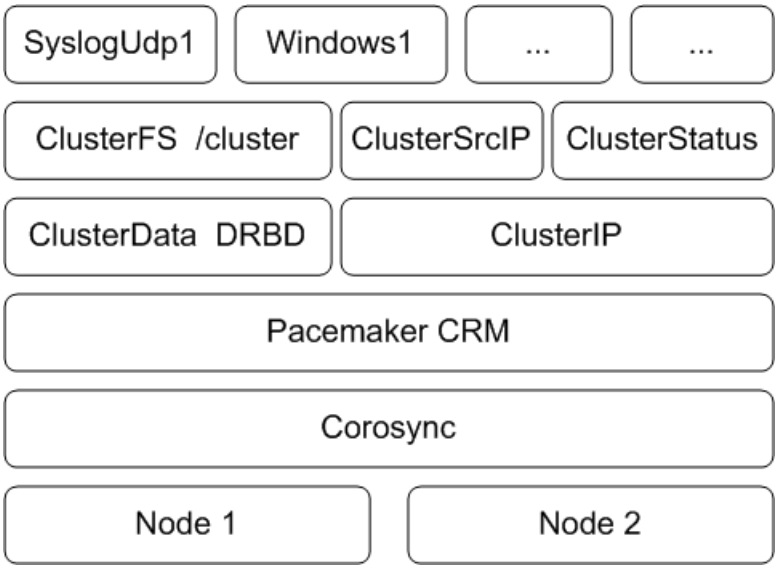
```
yum install -y glibc.i686 libX11.i686 libXext.i686 libXi.i686
libXtst.i686
```

Cluster Configuration Details

The following configuration will result in a syslog UDP SmartConnector and Windows Unified Connector that will run on either cluster node and send events to a single Logger SmartMessage Receiver destination. Additional destinations can be added by running the SmartConnector configuration. Description of the cluster resources configured:

Resource Primitive	Provides, Dependencies
ClusterData	Controls the DRBD disk partitions that will support the connector file system; Dependencies – DRBD, /dev/sdb1, management IP addresses
ClusterFS	Controls the floating cluster ext4 file system; Dependencies – ClusterData
ClusterIP	Controls the floating VIP used by the connectors; Dependencies – production network interfaces
ClusterSrcIP	Controls the outbound cluster traffic to ensure the outbound comes from the VIP; Dependencies – ClusterIP
ClusterStatus	Updates an index.html status file that is served over HTTP on the cluster VIP. Shows the state of all the cluster resources; Dependencies – ClusterIP and ClusterFS
SyslogUdp1	Syslog UDP Daemon listening on VIP UDP 514; Dependencies – All the previous resources

The building block resources work together as shown here



Configure Corosync

See www.corosync.org for details on the cluster message system.

Update configuration file

Edit the configuration file and set the network addresses. Be sure to chose a unique multicast IP address for this instance of corosync if multiple Corosync/Pacemaker clusters will be on the same network segments. The quickstart script currently does NOT ensure unique multicast addresses are used.

Note

Corosync can be configured to use encrypted hashes to authenticate other valid cluster nodes and this paper uses those features, but with a pre-generated authkey pre-shared secret. Production cluster nodes should regenerate a unique authkey file – this key gen typically MUST be run on the system console versus a SSH terminal session.

```
node1# cat etc/corosync/corosync.conf
#
# corosync.conf
#
totem {
  version: 2

  # crypto_cipher and crypto_hash: Used for mutual node authentication.
  # If you choose to enable this, then do remember to create a shared
  # secret with "corosync-keygen".
  secauth: on
  crypto_cipher: sha1
  crypto_hash: aes256

  # interface: define at least one interface to communicate
  # over. If you define more than one interface stanza, you must
  # also set rrp_mode.
  interface {
    ringnumber: 0
    bindnetaddr: 10.10.1.0
    mcastaddr: 239.255.1.1
    mcastport: 4000
    ttl: 1
  }
  interface {
    ringnumber: 1
    bindnetaddr: 10.20.1.0
    mcastaddr: 239.255.1.1
    mcastport: 4000
    ttl: 1
  }
  rrp_mode: passive
}

nodelist {
  node {
        ring0_addr: ca1
        ring1_addr: ca1-mgmt
        nodeid: 1
       }
  node {
        ring0_addr: ca2
        ring1_addr: ca2-mgmt
        nodeid: 2
       }
}

logging {
  # Log the source file and line where messages are generated
  fileline: off
  # Log to standard error
  to_stderr: no
  # Log to a log file
  to_logfile: yes
  logfile: /var/log/cluster/corosync.log
```

```
  # Log to the system log daemon. When in doubt, set to yes.
  to_syslog: yes
  # Log debug messages (very verbose)
  debug: off
  timestamp: on
  logger_subsys {
    subsys: QUORUM
    debug: off
  }
}

quorum {
  provider: corosync_votequorum
  expected_votes: 2
}

amf {
  mode: disabled
}
```

Install corosync configuration file

Copy in the corosync.conf file to the /etc/corosync directory on both nodes

```
node1# cp corosync.conf /etc/corosync/corosync.conf
node1# scp corosync.conf root@node2:/etc/corosync/corosync.conf
```

Start corosync

Start corosync on both nodes

```
node1# service corosync start
node1# ssh node2 -- service corosync start
```

Run the corosync configuration test to ensure corosync is running properly – also check /var/log/messages for major events

```
node1# corosync-cfgtool -s
Printing ring status.
Local node ID 1023480842
RING ID 0
     id    = 10.20.1.61
     status      = ring 0 active with no faults
RING ID 1
     id    = 172.16.100.61
     status      = ring 1 active with no faults
```

Check to see that corosync see quorum:

```
node1# corosync-quorumtool -l

Membership information
----------------------
    Nodeid      Votes Name
1023480842          1 node1.acme.com
1040258058          1 node2.acme.com
```

Configure DRBD

See www.drbd.org for details on the distributed replication block device system that will provide a network based RAID1 block device on which we will locate the shared cluster filesystem. DRBD is configured through a common configuration file that is the same on both nodes – it tells DRBD how to find both the partition it is to manage as well as the partner node.

Update configuration file

Edit the configuration file and set the network addresses and disk partition to be used, then distribute to both nodes. Be sure to chose a disk partition that is not automatically mounted or managed by the OS.

Warning

The drbdadm initialization commands will destroy data on the partition that is specified with no prior backup and no way to undo the operation. Be sure the partitions specified have no valid data before proceeding. The quickstart script will also use dd to zero out the start of each partition.

---

node1# cat /etc/drbd.d/clusterData.res

```
#
# version 1.2
#
resource clusterData {
 meta-disk internal;
 device  /dev/drbd0;
 syncer {
  verify-alg sha1;
 }
 net {
  allow-two-primaries;
  # split brain recovery options
  after-sb-0pri discard-zero-changes;
  after-sb-1pri discard-secondary;
  after-sb-2pri disconnect;
 }
 on NODENAME1A {
  disk   PARTITION;
  address  NODE1IP1:7789;
 }
 on NODENAME2A {
  disk   PARTITION;
  address  NODE2IP1:7789;
 }
}
```

Initialize the DRBD partitions

Tell DRBD to initialize the shared disk – for speed and convenience, run all these commands on node1

```
drbdadm create-md clusterData
modprobe drbd
drbdadm up clusterData
cat /proc/drbd
ssh node2 -- drbdadm --force create-md clusterData
ssh node2 -- modprobe drbd
ssh node2 -- drbdadm up clusterData
drbdadm -- --overwrite-data-of-peer primary clusterData
```

At this point, the DRBD shared disk should be synchronizing (primary > secondary). The real-time status of the DRBD partitions is shown in the /proc/drbd file.

Configure Pacemaker

See www.clusterlabs.org for details on the Pacemaker cluster resource manager. Pacemaker implements a cluster resource manager that includes health checks and a policy engine that controls where and when identified cluster resources are stopped and started. In this paper, the cluster resources are grouped into a Base Group (DRBD, file system, IP) and a Connector Group. This allows maintenance of cluster resources and ordered start/stop. See the crm configuration scripts (/root/crm/crm) for detail on the primitives, node limitations, ordering and dependencies that are setup.

Given the complexities of setting up cluster resource primitives and the dependancies, the details are not outlined here. Examine the cluster resource manager setup scripts in /root/crm/crm for the specific cluster configuration commands and sequence.

Current cluster status can be displayed by running the crm_mon command

```
crm_mon
```

```
Last updated: Tue Sep 17 11:19:58 2013
Last change: Tue Sep 17 11:03:36 2013 via crm_resource on ca1
Stack: classic openais (with plugin)
Current DC: ca2 - partition with quorum
Version: 1.1.8-7.el6-394e906
2 Nodes configured, 2 expected votes
8 Resources configured.


Online: [ ca1 ca2 ]

 Master/Slave Set: ClusterDataClone [ClusterData]
     Masters: [ ca1 ]
     Slaves: [ ca2 ]
ClusterFS       (ocf::heartbeat:Filesystem):    Started ca1
ClusterIP       (ocf::heartbeat:IPaddr2):       Started ca1
ClusterSrcIP    (ocf::heartbeat:IPsrcaddr):     Started ca1
WebServer       (ocf::heartbeat:apache):        Started ca1
ClusterStatus   (ocf::pacemaker:ClusterMon):    Started ca1
```

Configure ArcSight SmartConnectors

After the base cluster resources are setup, the quickstart script will display the values it expects the user to setup within the syslog SmartConnector, then stops and waits for the user to complete the SmartConnector installation.

Configure Remaining Pacemaker Services

After the connector is setup, the quickstart script adds the connector primitives to the cluster and sets up health monitoring. At this point, the cluster is operational for syslog events directed at the cluster VIP.

Operating the Cluster

Starting and Stopping Connectors

If a connector needs to be stopped or started (taken out of cluster control), that can be done with a resource stop or start command

```
pcs resource stop SyslogUdp1
pcs resource start SyslogUdp1
```

Startup and Shutdown of Cluster

Manual stop of the cluster can be accomplished by finding the node which is secondary and running a "shutdown -h now", then run the same on the primary.

Boot of the nodes should automatically start the corosync and pacemaker components, then the cluster resources will select a node and start.

Move Connectors to Partner Node

Moving connectors to the partner node is best done by asking the cluster to move the underlaying file system. Due to the dependencies established, this will gracefully move the connectors as well

```
pcs cluster standby node1
```

```
Node ca1: standby
Online: [ ca2 ]

Full list of resources:

 Master/Slave Set: ClusterDataClone [ClusterData]
     Masters: [ ca2 ]
     Stopped: [ ClusterData:1 ]
 ClusterFS       (ocf::heartbeat:Filesystem):    Started ca2
 ClusterIP       (ocf::heartbeat:IPaddr2):       Started ca2
 ClusterSrcIP    (ocf::heartbeat:IPsrcaddr):     Started ca2
 WebServer       (ocf::heartbeat:apache):        Started ca2
 ClusterStatus   (ocf::pacemaker:ClusterMon):    Started ca2
 SyslogUdp1      (ocf::arcsight:SmartConnector): Started ca2
```

After the cluster has stabilized, perform an unmove operation to give control of the resources back to the cluster policy engine

```
pcs cluster unstandby node1
```

```
Online: [ ca1 ca2 ]

Full list of resources:

 Master/Slave Set: ClusterDataClone [ClusterData]
     Masters: [ ca2 ]
     Slaves: [ ca1 ]
 ClusterFS       (ocf::heartbeat:Filesystem):    Started ca2
 ClusterIP       (ocf::heartbeat:IPaddr2):       Started ca2
 ClusterSrcIP    (ocf::heartbeat:IPsrcaddr):     Started ca2
 WebServer       (ocf::heartbeat:apache):        Started ca2
 ClusterStatus   (ocf::pacemaker:ClusterMon):    Started ca2
 SyslogUdp1      (ocf::arcsight:SmartConnector): Started ca2
```

Manually Resolving DRBD Split-brain

In the case DRBD detects dual primary owners of the cluster disk, there are policies set to help automatically recover from that situation. Where the automatic resolution fails, a manual intervention is needed to recover the DRBD synchronization.

This example assumes the cluster was last running on node1 – all commands are run on node1

Shutdown the cluster

```
service pacemaker stop
ssh node2 -- service pacemaker stop
```

Resolve the DRBD split brain

```
drbdadm down clusterData
ssh node2 -- drbdadm down clusterData
drbdadm attach clusterData
drbdadm primary clusterData
ssh node2 -- drbdadm attach clusterData
ssh node2 -- drbdadm secondary clusterData
ssh node2 -- drbdadm connect clusterData
drbdadm connect clusterData
```

Restart the cluster

```
service pacemaker start
ssh node2 -- service pacemaker start
```

Follow the cluster startup instructions to bring the SmartConnectors back up to a running state.

Syslog Events from the Cluster

To immediately generate syslog events to verify SyslogUdp1 connector operation and to monitor the cluster, update the /etc/rsyslog.conf file on both nodes to include the following where CLUSTERVIP is the VIP assigned to the cluster (and the UDP port matches what was used in the .clusterinfo file):

```
$WorkDirectory /var/lib/rsyslog # where to place spool files
$ActionQueueFileName fwdRule1 # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as
possible)
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
$ActionQueueType LinkedList   # run asynchronously
$ActionResumeRetryCount -1    # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#*.* @@remote-host:514
*.* @CLUSTERVIP:514
```

Be sure to restart the rsyslog daemon on both cluster nodes

```
service rsyslog restart
```

Monitoring Connector Activity in the Cluster

Best way to see what the connectors are doing is to sign into the primary cluster node, as it will have the CLUSTERFSDIR (/cluster) mounted.

Tail the connector wrapper log

```
tail -f /cluster/connectors/syslog-udp-1/current/logs/agent.out.wrapper.log
```

Look for indication that the Event Thread and the Health Thread are Up:

```
INFO | jvm 1 | 2013/02/10 01:05:17 | [Sun Feb 10 01:05:17 CST 2013] [INFO ] {C=0,
ET=Up, HT=Up, N=Syslog UDP Daemon 1 ...
```

You should see the ET and HT are Up and see the event count incrementing.

Troubleshooting / FAQ

Q1: Why would I use this configuration?

A1: In the case high-availability of event collection by ArcSight SmartConnectors is needed without the use of external hardware load-balancers. This configuration is specifically intended to be used on two vanilla 1U servers with specifications similar to the HP ArcSight C5500 Connector Appliances. Note that modifying actual Connector Appliances will very likely invalidate your support agreement. This configuration quickly gets a syslog UDP daemon SmartConnector running in a highly available two-node cluster using the Pacemaker cluster resource management tools described by clusterlabs.org.

Q2: Does this configuration cluster two Connector Appliances?

A2: No. See Answer 1. This only clusters a syslog UDP daemon SmartConnector. Remote management of connectors is configured so heartbeat can be tested and the connectors themselves can be managed by a hardware or software Connector Appliance.

Q3: Do I need any SAN or NAS disk to make the cluster work?

A3: No. In fact the initial configuration presented here needs a local disk partition on each node to be dedicated to the shared cluster file system. Theoretically SAN disk can be used since it would just look like a local disk partition, however this configuration is intended to be as cost effective as possible, so expects two garden variety general purpose servers with no external components.

Q4: How long does this take to get running?

A4: Not including time to load the CentOS 6.4 OS, it took about 15 minutes to have the cluster running by using the quick-start script and pre-configured connectors. Time from scratch including loading the OS, configuration of SmartConnectors and cluster shouldn't take more than four (4) hours.

Q5: Can I use bonded Ethernet interfaces?

A5: Should be possible. Just ensure the network interfaces are setup statically (eg. eth0 + eth1 = bond0 and bond0 has a static address defined, same for bond1). Cluster resource script IPaddr2 should detect the interface name correctly – this has not been tested though.

Q6: If the cluster goes down do the connectors continue running?

A6: Probably not. If the cluster was shutdown gracefully, definitely not since the configured VIP and shared file system would be stopped. It's unlikely both cluster nodes would fail, although with only two nodes, split-brain is a danger. This configuration can be far more resilient with the addition of another node, or STONITH devices – that will be addressed in the future.

Q7: Does each connector need its own VIP?

A7: Only if they will be using the same TCP or UDP port(s). This initial configuration only uses a single VIP.

Q8: How long does the cluster take to move a connector to the partner node?

A8: A properly functioning cluster can have the move of the VIP in less than three (3) seconds, however the connector shutdown and startup may take tens of seconds more. However 30 to 50 seconds can be possible if the cluster has problems stopping the connector on the source node. In the case of a cluster node failure where both the management and

production interfaces are offline (for example in the case a cluster node has crashed or lost power), the transition is usually only several seconds, definitely sub-minute.

Q9: What is the weakest part of the cluster?

A9: Currently there are two significant weaknesses: (1) No STONITH capability so the cluster will likely need manual intervention to recover from any split-brain scenario, (2) DRBD configuration, which needs to have additional auto-recovery from split-brain. Currently it is possible to corrupt the DRBD partitions if split-brain recovery is not performed in exactly the correct order.

Q10: Can I monitor the cluster with my network monitoring tools?

A10: Probably. There is no SNMP MIB, but there are several command line cluster status commands that will reveal the health of the cluster and the connectors it manages. Look at the ClusterMon resource for other ideas.

Q11: Can I call HP Enterprise Security Products support line to get support with this configuration?

A11: No. This must be fully supported by the admins that manage it, however there are some excellent online forums at clusterlabs.org.

Q12: The IPsrcaddr primitive immediately fails when the cluster is trying to start it. What's wrong?

A12: If you see the following from "ip route" then auto-configuration is enabled and will cause IPsrcaddr to fail. You must disable this with a NOZEROCONF=yes in /etc/sysconfig/network

```
169.254.0.0/16 dev eth0  scope link  metric 1002
169.254.0.0/16 dev eth1  scope link  metric 1003
```

Q13: How can I add more connectors?

A13: Follow the logic in the connector section of the quickstart script. Basic steps are:

a) Run the connector installation on the currently active cluster node, or an equivalent system (with the EXACT same file system path .. eg. /cluster/connectors/smartconnectorname)

b) Setup the connector to run as a service but not to automatically start on system boot

c) Copy the /etc/init.d startup script to the partner node

d) Ensure the connector starts and runs correctly via /etc/init.d script start and stop commands

e) Ensure the remote management port is setup and unique to the other connectors (edit agent.properties)

f) Add the connector to the cluster and to the ConnectorGroup

g) Update any control scripts used to start and stop the connectors within the cluster (eg. scripts/resources.shl)

Q14: Can I setup a Logger and a Manager destination for quickstart to automatically configure?

A14: Not at this time. Preconfigured Manager destinations will not work since the unique manager certificate must be imported. To use the manager destination, start the base cluster

resources, stop the connector in question then launch a reconfiguration of the connector from the ARCSIGHT_HOME/current/bin directory

Q15: When I run the cluster quickstart script, I see several error messages similar to:

WARNING: ClusterStatus-prefer-node1: referenced node xyz does not exist

Does this mean the install failed?

A15: No. These are messages generated by crm and as per release notes from ClusterLabs.org, they can safely be ignored.

Known Issues with Current Version

DRBD on RHEL 6.2

Although the corosync and pacemaker packages are available in the standard RHEL 6.2 distribution, it appears that DRBD is missing and would need to be acquired, configured and installed for the cluster to function. This is somewhat a moot point since the next revision of this whitepaper is targeted for RHEL 6.4

Dedicated web server required

Current configuration of the cluster requires dedicated use of the Apache web server on both cluster nodes. Future versions will enable co-location of cluster status with other content served by the Apache web server.

crm_mon exits with strange 'upgrade' errors

Known issue with crm_mon – occurs after certain commands that modify the Cluster Information Base (CIB). Workaround – just restart the crm_mon command.

Cluster failover needs fully synchronized DRBD partition

This is not really an 'issue' as the cluster needs a fully synchronized DRBD partition in order to properly activate on the secondary node(s). The impact is you need to wait until the initial sync is complete *before* any failover testing.

Further Reading

Work in this white paper is based on the quick start configuration published by **ClusterLabs.org** and related HA information found at **linux-ha.org.**

Quickstart Installation Log

The following is an excerpt of a successful cluster installation log. Note is has been edited for readability.

```
root@ca1 ~]# date 091607592013
Mon Sep 16 07:59:00 CDT 2013

[root@ca1 ~]# yum install -y openssh-clients

[root@ca1 ~]# ls
ArcSight-6.0.5.6782.0-Connector-Linux.bin  crm-2.0.5.tar

[root@ca1 ~]# tar xf crm-2.0.6.tar
[root@ca1 ~]# chmod +x ArcSight-6.0.5.6782.0-Connector-Linux.bin
[root@ca1 ~]# cd crm
[root@ca1 crm]# ls
agents  cluster-quickstart-2.0.6.shl  etc  pcs  usr

[root@ca1 crm]# ./cluster-quickstart-2.0.6.shl


================================================================
Cluster Quickstart - version 2.0.6 - starting at Mon Sep 16 08:01:16 CDT 2013

                    ** WARNING ** WARNING ** WARNING **
   This cluster setup script WILL modify settings on your systems including
   disabling security features such as SElinux and IPtables.  Review cluster
   settings carefully, since the disk partition specified (in the .clusterinfo
   defaults file and/or variable setting) WILL BE OVERWRITTEN with no backup.
   All data currently on that partition on BOTH nodes WILL BE DESTROYED -
   be sure it exists and is backed up before you run this script.

   This install process is intended to be run on the first cluster node ONLY.
   Do NOT run any setup commands on the second cluster node unless explicitly
   instructed to do so.

 You can select one of three options at the start of each step:
 Proceed - Perform the commands in the step
 Skip - Skip the step and go onto the next
 Quit - Immediately stop and exit the quickstart script

Do you want to proceed with cluster quickstart?
Proceed/Skip/Quit? [p]/s/q:


================================================================
Step 1: Setup cluster parameters

This step displays default values for cluster configuration so you can verify
and make changes to the cluster parameters (highly recommended)
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Set value for clusternode1 [ca1]:
Set value for clusternode1internal [ca1-mgmt]:
Set value for clusternode1ip1 [10.20.1.193]:
Set value for clusternode1ip2 [10.10.1.193]:
Set value for clusternode2 [ca2]:
Set value for clusternode2internal [ca2-mgmt]:
Set value for clusternode2ip1 [10.20.1.194]:
Set value for clusternode2ip2 [10.10.1.194]:
Set value for clusternetwork1 [10.20.1.0]:
Set value for clusternetwork2 [10.10.1.0]:
```

```
Set value for clusterdomain [pomeroy.us]:
Set value for clusterip [10.20.1.195]:
Set value for clustername [sc1]:
Set value for clusterfsdir [/c]: /cfs
Set value for clusterdrbdpartition [/dev/sda3]:
Set value for arstsyslog1destip1 [10.20.1.55]:
Set value for arstsyslog1destport1 [443]:
Set value for arstsyslog1destreceiver1 [SmartMessage Receiver]:
Set value for arstsyslog1mgmtport [9001]:
Set value for arstsyslog1agentport [514]:


============================================================
Step 2: Setting up root ssh keys

This step generates and exchanges cluster node root ssh keys. Do NOT run
 .. this step if the root account on both nodes have mutual public key
 .. authentication setup already. ssh clients MUST have been installed
 .. on ca1 AND ca2 before this step.
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Generating keys on ca1
 .. press <enter> at following filename and passphrase prompts
 .. since we want the default filename/location and empty passphrase

    ----------------------------------------
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:
c4:6a:8d:79:a4:fe:ac:05:a9:c4:19:ba:1f:d1:57:1b root@ca1

    ----------------------------------------
Generating keys on ca2 via ssh
 .. enter root password, then
 .. press <enter> at following filename and passphrase prompts
 .. since we want the default filename/location and empty passphrase.
 .. Ensure to accept any host key prompts

    ----------------------------------------
The authenticity of host '10.20.1.194 (10.20.1.194)' can't be established.
RSA key fingerprint is a7:7a:b5:a6:13:51:cf:59:5e:27:30:01:df:1b:d2:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.20.1.194' (RSA) to the list of known hosts.
root@10.20.1.194's password:
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
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:
e7:e7:5d:03:d6:c4:28:c0:bd:19:33:0e:3d:f6:4d:b2 root@ca2

    ----------------------------------------
Exchanging keys
 .. enter ca2 root password
```

```
root@10.20.1.194's password:
authorized_keys                                                100%  390     0.4KB/s   00:00
id_rsa.pub.ca1.root                                            100%  390     0.4KB/s   00:00
id_rsa.pub.ca2.root                                            100%  390     0.4KB/s   00:00


============================================================
Step 3: Build common hosts file

This step builds a common /etc/hosts file and installs on both nodes
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Building hosts file

    ----------------------------------------
Distributing hosts files - if ssh keys are not setup yet, you will need
 .. to enter ca2 root password when prompted
Making directory ca2 /etc
Backing up file etc/hosts
Copying file ca1 etc/hosts
Copying file ca2 etc/hosts
The authenticity of host 'ca2 (10.20.1.194)' can't be established.
RSA key fingerprint is a7:7a:b5:a6:13:51:cf:59:5e:27:30:01:df:1b:d2:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ca2' (RSA) to the list of known hosts.
hosts                                                          100%  519     0.5KB/s   00:00


============================================================
Step 4: Network/security service configuration

This step updates /etc/sysconfig/network, disables both SElinux and IPtables
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Restarting network services on ca1
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface eth1:                                [  OK  ]
Restarting network services on ca2
Shutting down interface eth0:  [  OK  ]
Shutting down interface eth1:  [  OK  ]
Shutting down loopback interface:  [  OK  ]
Bringing up loopback interface:  [  OK  ]
Bringing up interface eth0:  [  OK  ]
Bringing up interface eth1:  [  OK  ]

    ----------------------------------------
Disabling iptables
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
iptables: Flushing firewall rules: [  OK  ]
iptables: Setting chains to policy ACCEPT: filter [  OK  ]
iptables: Unloading modules: [  OK  ]

    ----------------------------------------
Disabling SELinux
```

```
==============================================================
Step 5: Install cluster related packages

This step installs corosync, pacemaker, pcs, DRBD and related packages
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Installing on ca1
Retrieving http://elrepo.org/elrepo-release-6-5.el6.elrepo.noarch.rpm
warning: /var/tmp/rpm-tmp.9VkV3o: Header V4 DSA/SHA1 Signature, key ID baadae52: NOKEY
Preparing...                ########################################### [100%]
   1:elrepo-release         ########################################### [100%]
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirror.liberty.edu
 * elrepo: elrepo.org
 * extras: mirror.san.fastserv.com
 * updates: mirror.cogentco.com
elrepo                                                                   | 2.9 kB     00:00
elrepo/primary_db                                                        | 675 kB     00:01
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Finished Dependency Resolution

Dependencies Resolved

Install      61 Package(s)
Upgrade       3 Package(s)

Total download size: 57 M

Installed:
  corosync.x86_64 0:1.4.1-15.el6_4.1   glibc.i686 0:2.12-1.107.el6_4.4     httpd.x86_64 0:2.2.15-
29.el6.centos
  libX11.i686 0:1.5.0-4.el6            libXext.i686 0:1.3.1-2.el6          libXi.i686 0:1.6.1-3.el6
  libXtst.i686 0:1.2.1-2.el6           ntp.x86_64 0:4.2.4p8-3.el6.centos   pacemaker.x86_64 0:1.1.8-
7.el6
  pcs.noarch 0:0.9.26-10.el6_4.1       wget.x86_64 0:1.12-1.8.el6

Dependency Installed:
  apr.x86_64 0:1.3.9-5.el6_2                              apr-util.x86_64 0:1.3.9-3.el6_0.1
  apr-util-ldap.x86_64 0:1.3.9-3.el6_0.1                  cifs-utils.x86_64 0:4.8.1-18.el6
  cluster-glue-libs.x86_64 0:1.0.5-6.el6                  clusterlib.x86_64 0:3.0.12.1-49.el6_4.2
  corosynclib.x86_64 0:1.4.1-15.el6_4.1                   gnutls.x86_64 0:2.8.5-10.el6_4.2
  httpd-tools.x86_64 0:2.2.15-29.el6.centos               keyutils.x86_64 0:1.4-4.el6
  libX11-common.noarch 0:1.5.0-4.el6                      libXau.i686 0:1.0.6-4.el6
  libevent.x86_64 0:1.4.13-4.el6                          libgssglue.x86_64 0:0.1-11.el6
  libibverbs.x86_64 0:1.1.6-5.el6                         libnl.x86_64 0:1.1.4-1.el6_4
  libqb.x86_64 0:0.14.2-3.el6                             librdmacm.x86_64 0:1.0.17-
0.git4b5c1aa.el6
  libtalloc.x86_64 0:2.0.7-2.el6                          libtasn1.x86_64 0:2.3-3.el6_2.1
  libtdb.x86_64 0:1.2.10-1.el6                            libtirpc.x86_64 0:0.2.1-6.el6_4
  libtool-ltdl.x86_64 0:2.2.6-15.5.el6                    libxcb.i686 0:1.8.1-1.el6
  libxslt.x86_64 0:1.1.26-2.el6_3.1                       lm_sensors-libs.x86_64 0:3.1.1-17.el6
  mailcap.noarch 0:2.1.31-2.el6                           net-snmp-libs.x86_64 1:5.5-44.el6_4.4
  nfs-utils.x86_64 1:1.2.3-36.el6                         nfs-utils-lib.x86_64 0:1.1.5-6.el6
  nss-softokn-freebl.i686 0:3.14.3-3.el6_4                ntpdate.x86_64 0:4.2.4p8-3.el6.centos
  pacemaker-cli.x86_64 0:1.1.8-7.el6                      pacemaker-cluster-libs.x86_64 0:1.1.8-
7.el6
  pacemaker-libs.x86_64 0:1.1.8-7.el6                     perl.x86_64 4:5.10.1-131.el6_4
  perl-Module-Pluggable.x86_64 1:3.90-131.el6_4           perl-Pod-Escapes.x86_64 1:1.04-131.el6_4
```

```
  perl-Pod-Simple.x86_64 1:3.13-131.el6_4                  perl-TimeDate.noarch 1:1.16-11.1.el6
  perl-libs.x86_64 4:5.10.1-131.el6_4                      perl-version.x86_64 3:0.77-131.el6_4
  pkgconfig.x86_64 1:0.23-9.1.el6                          quota.x86_64 1:3.17-18.el6
  resource-agents.x86_64 0:3.9.2-21.el6_4.3                rpcbind.x86_64 0:0.2.0-11.el6
  samba-common.x86_64 0:3.6.9-151.el6_4.1                  samba-winbind.x86_64 0:3.6.9-151.el6_4.1
  samba-winbind-clients.x86_64 0:3.6.9-151.el6_4.1         tcp_wrappers.x86_64 0:7.6-57.el6

Dependency Updated:
  glibc.x86_64 0:2.12-1.107.el6_4.4                       glibc-common.x86_64 0:2.12-1.107.el6_4.4
  nss-softokn-freebl.x86_64 0:3.14.3-3.el6_4

Complete!
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.liberty.edu
 * elrepo: elrepo.org
 * extras: mirror.san.fastserv.com
 * updates: mirror.cogentco.com
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package drbd83-utils.x86_64 0:8.3.15-1.el6.elrepo will be installed
---> Package kmod-drbd83.x86_64 0:8.3.15-1.el6_3.elrepo will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Installed:
  drbd83-utils.x86_64 0:8.3.15-1.el6.elrepo              kmod-drbd83.x86_64 0:8.3.15-1.el6_3.elrepo

Complete!

    ----------------------------------------
Installing on ca2

...

Complete!


================================================================
Step 6: Set up NTP client

This step enables, starts NTP .. and will use default CentOS time pool
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Enabling NTP on ca1
Starting ntpd:                                             [  OK  ]

    ----------------------------------------
Enabling NTP on ca2
Starting ntpd: [  OK  ]


================================================================
Step 7: Build corosync configuration file

This step builds and distributes the corosync conf file
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Building and distributing corosync.conf file to both nodes
```

```
Backing up file etc/corosync/corosync.conf
Copying file ca1 etc/corosync/corosync.conf
Copying file ca2 etc/corosync/corosync.conf
corosync.conf                                                 100% 1387     1.4KB/s   00:00
Backing up file etc/corosync/authkey
Copying file ca1 etc/corosync/authkey
Copying file ca2 etc/corosync/authkey
authkey                                                       100%  128     0.1KB/s   00:00
Backing up file etc/corosync/service.d/pcmk
Copying file ca1 etc/corosync/service.d/pcmk
Copying file ca2 etc/corosync/service.d/pcmk
pcmk                                                          100%   87     0.1KB/s   00:00


============================================================
Step 8: Start and test corosync on both nodes

This step starts corosync on both nodes and tests the setup
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Starting corosync daemon both nodes
Starting Corosync Cluster Engine (corosync):               [  OK  ]
Starting Corosync Cluster Engine (corosync): [  OK  ]
Printing ring status.
Local node ID -1056896502
RING ID 0
        id      = 10.10.1.193
        status  = ring 0 active with no faults
RING ID 1
        id      = 10.20.1.193
        status  = ring 1 active with no faults
Printing ring status.
Local node ID -1040119286
RING ID 0
        id      = 10.10.1.194
        status  = ring 0 active with no faults
RING ID 1
        id      = 10.20.1.194
        status  = ring 1 active with no faults
Nodeid     Votes  Name
3238070794     1  ca1-mgmt.pomeroy.us
3254848010     1  ca2-mgmt.pomeroy.us

    ----------------------------------------
Select Skip to end tests,
 .. Proceed to re-start corosync and re-run tests
Proceed/Skip/Quit? [p]/s/q: s

    ----------------------------------------
Stopping corosync daemon both nodes
Signaling Corosync Cluster Engine (corosync) to terminate: [  OK  ]
Waiting for corosync services to unload:.                  [  OK  ]
Signaling Corosync Cluster Engine (corosync) to terminate: [  OK  ]
Waiting for corosync services to unload:[  OK  ]

    ----------------------------------------

If tests were successful, select Proceed
 .. otherwise, select Quit and fix the problem then re-run this step
Proceed/Skip/Quit? [p]/s/q:
```

```
================================================================
Step 9: Build DRBD configuration file

This step builds and distributes the clusterData resource file to both nodes
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Building and distributing DRBD clusterData.res file to both nodes
Backing up file etc/drbd.d/clusterData.res
Copying file ca1 etc/drbd.d/clusterData.res
Copying file ca2 etc/drbd.d/clusterData.res
clusterData.res                                                100%  408     0.4KB/s   00:00


================================================================
Step 10: Setting up DRBD partitions on both nodes

This configures DRBD partition on both nodes and sets up primary/secondary

** WARNING ** WARNING ** WARNING ** WARNING ** WARNING **

This step WILL destroy data on /dev/sda3 on both nodes.
Do NOT perform this step if it has been run previously, unless you
 .. really know what you're doing. It will likely fail and require
 .. manual recovery.
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Zeroing partition header space on both nodes

    ----------------------------------------
Writing drbd metadata to partition on ca1

  --==  Thank you for participating in the global usage survey  ==--
The server's response is:

you are the 13106th user to install this version
Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success

    ----------------------------------------
Bringing up partition on ca1

    ----------------------------------------
ca1 /proc/drbd:
version: 8.3.15 (api:88/proto:86-97)
GIT-hash: 0ce4d235fc02b5c53c1c52c53433d11a694eab8c build by phil@Build64R6, 2012-12-20 20:09:51
 0: cs:WFConnection ro:Secondary/Unknown ds:Inconsistent/DUnknown C r----s
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:1017236

    ----------------------------------------
Writing drbd metadata to partition on ca2
NOT initialized bitmap
Writing meta data...
initializing activity log
New drbd meta data block successfully created.

    ----------------------------------------
Bringing up partition on ca2
```

```
    ----------------------------------------
ca2 /proc/drbd:
version: 8.3.15 (api:88/proto:86-97)
GIT-hash: 0ce4d235fc02b5c53c1c52c53433d11a694eab8c build by phil@Build64R6, 2012-12-20 20:09:51
 0: cs:WFConnection ro:Secondary/Unknown ds:Inconsistent/DUnknown C r----s
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:1017236

    ----------------------------------------
Forcing ca1 to be primary
Partition sync in progress now, check progress with
  cat /proc/drbd

    ----------------------------------------
Making filesystem on drbd device
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
63616 inodes, 254309 blocks
12715 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=264241152
8 block groups
32768 blocks per group, 32768 fragments per group
7952 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.

    ----------------------------------------
Creating /cfs mount point on both nodes

    ----------------------------------------
Creating /cfs status webserver DocumentRoot

    ----------------------------------------
Updating httpd conf file both nodes
Backing up file etc/httpd/conf/httpd.conf
Copying file ca1 etc/httpd/conf/httpd.conf
Copying file ca2 etc/httpd/conf/httpd.conf
httpd.conf                                                    100%   34KB  33.6KB/s   00:00


============================================================
Step 11: Setup base cluster

This builds and starts up the base cluster (partition, fs, IP, status)
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Installing ArcSight SmartConnector cluster resource agent to both nodes
Making directory ca2 /usr/lib/ocf/resource.d/arcsight
Backing up file usr/lib/ocf/resource.d/arcsight/SmartConnector
Copying file ca1 usr/lib/ocf/resource.d/arcsight/SmartConnector
Copying file ca2 usr/lib/ocf/resource.d/arcsight/SmartConnector
```

```
SmartConnector                                                100% 7532     7.4KB/s   00:00

    ----------------------------------------
Setting up base cluster
Backing up file etc/corosync/corosync.conf
Copying file ca1 etc/corosync/corosync.conf
Copying file ca2 etc/corosync/corosync.conf
corosync.conf                                                 100% 1387     1.4KB/s   00:00

    ----------------------------------------
Restarting corosync and pacemaker daemons
Starting Corosync Cluster Engine (corosync):               [  OK  ]
Starting Corosync Cluster Engine (corosync): [  OK  ]
Starting Pacemaker Cluster Manager:                        [  OK  ]
Starting Pacemaker Cluster Manager: [  OK  ]
Waiting 30 seconds for cluster daemons to fully start .. you can
run crm_mon via the command line to monitor progress

    ----------------------------------------
Configuring cluster defaults


============================================================
Step 12: Setup cluster resources (partition, fs, IP)

This step sets up the cluster resource primitives and dependencies
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Configure the Cluster for DRBD
CIB updated

    ----------------------------------------
Configure the Cluster FS on DRBD
Adding ClusterDataClone ClusterFS (kind: Mandatory) (Options: first-action=promote then-action=start)
CIB updated

    ----------------------------------------
Configure Cluster IP, SrcIP, Web, ClusterMon
Adding ClusterFS ClusterIP (kind: Mandatory) (Options: first-action=start then-action=start)
CIB updated
Adding ClusterIP ClusterSrcIP (kind: Mandatory) (Options: first-action=start then-action=start)
CIB updated
Adding ClusterSrcIP WebServer (kind: Mandatory) (Options: first-action=start then-action=start)
CIB updated
Adding WebServer ClusterStatus (kind: Mandatory) (Options: first-action=start then-action=start)
CIB updated


============================================================
Step 13: Setup syslog SmartConnector

This step pauses the cluster installation to allow you to install a
 .. syslog SmartConnector to be controlled by the cluster. After you
 .. select Proceed, you will be shown values to use and the cluster
 .. will pause until the SmartConnector install is complete and you
 .. chose to resume the cluster setup.

               ** NOTICE ** NOTICE ** NOTICE **

The base cluster steps MUST be run before this connector installation
 .. step will succeed, since the base resources (ClusterData, ClusterFS and
 .. ClusterIP) are required for this connector setup step to work.
```

```
If you have restarted the quickstart script to get to this
 .. step, ensure the above mentioned cluster resources are running by
 .. running "pcs status" or "crm_mon" from a separate command window.
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------

             ** WARNING ** WARNING ** WARNING **

You MUST use the following values while installing the syslog SmartConnector:

Connector install folder: /cfs/connectors/syslog-udp-1
Do NOT create links
Connector name: syslog-udp-1      DeviceLocation: sc1
syslog port to listen on: 514
IP address to run on: 10.20.1.195 .. do *NOT* specify ALL
Service Internal Name:  syslog-udp-1
Service Display Name:   Syslog UDP Daemon 1
Standalone or as a service: Service
Start the service automatically: No
Destination Host Name/IP: 10.20.1.55  Port: 443
Receiver Name: SmartMessage Receiver


    ----------------------------------------
1. Start the SmartConnector install by running the Linux SmartConnector
   in another window or terminal session.
2. Run the SmartConnector setup by:
    cd /cfs/connectors/syslog-udp-1/current/bin
    ./runagentsetup.sh
3. To verify proper SmartConnector operation, start the SmartConnector
   in another window by running:
    /etc/init.d/arc_syslog_udp_1 start

   Look for connector startup events on your
   10.20.1.55:443:SmartMessage Receiver
   destination.  You may also see the connector wrapper log here:
    tail -f /cfs/connectors/syslog-udp-1/current/logs/agent.out.wrapper.log

4. Stop the connector by running:
    /etc/init.d/arc_syslog_udp_1 stop

Press <Enter> to continue


    ----------------------------------------
Adding remote management properties to agent.properties file

    ----------------------------------------
Copying init.d script to ca2
arc_syslog_udp_1                                              100% 2494     2.4KB/s   00:00


============================================================
Step 14: Setup SmartConnector cluster resources

This step sets up the connector cluster resource primitives and dependencies
Proceed/Skip/Quit? [p]/s/q:

    ----------------------------------------
Building syslog SmartConnector cluster primitives and dependencies
Adding ClusterSrcIP SyslogUdp1 (kind: Mandatory) (Options: first-action=start then-action=start)
```

```
CIB updated


============================================================
Setup complete


Cluster quickstart is complete.  If all has turned out well, you have
a syslog connector configured and running under control of the cluster - you
should see some base events from the cluster nodes in the Logger that has been
configured as the destination.  The rsyslog daemon on both cluster nodes has
been configured to sent events to the syslog SmartConnector running on
  10.20.1.195:514/udp

Browse to http://10.20.1.195/status.html for the current cluster status, or
run "pcs status" or "crm_mon" to see the status via command line.

To force a migration to ca2, tell the cluster to put ca1 into
standby mode with:
  pcs cluster standby ca1
Be patient as everything moves, then give control of the resource location
back to the cluster by:
  pcs cluster unstandby ca1
All the cluster resources will stay on ca2 until it is put into
standby or experiences a resource failure that prompts the cluster to move
the resources to the partner node.

To setup additional connectors, see the commands that run during the
SmartConnector setup step and adjust to suit the new connector.

Read the Building a HA SmartConnector Cluster and Clusters from Scratch for
more background and operational information.

Cluster quickstart exiting at Mon Sep 16 08:42:29 CDT 2013
```