VCS cheat sheet -complete

June 13, 2018 ramesh test123 0

Veritas Cluster Server (also known as VCS and also sold bundled in the SFHA product) is a High-availability cluster software, for Unix, Linux and Microsoft Windows computer systems, created by Veritas Software (now part of Symantec). It provides application cluster capabilities to systems running databases, file sharing on a network, electronic commerce websites or other applications.

LLT (Low-Latency Transport)
veritas uses a high-performance, low-latency protocol for cluster communications. LLT runs directly on top of the data link provider interface (DLPI) layer ver ethernet and has several major junctions:

- sending and receiving heartbeats
- monitoring and transporting network traffic over multiple network links to every active system within the cluster
- load-balancing traffic over multiple links
- maintaining the state of communication
- providing a nonroutable transport mechanism for cluster communications.

Group membership services/Atomic Broadcast (GAB)
GAB provides the following:

- **Group Membership Services** – GAB manitains the overall cluster membership by the way of its Group Membership Sevices function. Heartbeats are used to determine if a system is active member, joining or leaving a cluster. GAB

SEARCH …

RECENT POSTS

AWK shell script

SED shell script

Difference between solaris 9 & solaris 10

Solaris Zones or Zone containers

RHEL –IPTABLES (tcpwrapers in solaris) :-

RECENT COMMENTS

ARCHIVES

September 2018

June 2018

October 2016

CATEGORIES

test123

META

determines what the position of a system is in within a cluster.

- **Atomic Broadcast** – Cluster configuration and status information is distributed dynamically to all system within the cluster using GAB's Atomic Broadcast feature. Atomic Broadcast ensures all active system receive all messages, for every resource and service group in the cluster. Atomic means that all system receive the update, if one fails then the change is rolled back on all systems.

Log in

Entries RSS

Comments RSS

WordPress.org

High Availability Daemon (HAD)

The HAD tracks all changes within the cluster configuration and resource status by communicating with GAB. Think of HAD as the manager of the resource agents. A companion daemon called **hashadow** moniotrs HAD and if HAD fails hashadow attempts to restart it. Like wise if hashadow daemon dies HAD will restart it. HAD maintains the cluster state information. HAD uses the **main.cf** file to build the cluster information in memory and is also responsible for updating the configuration in memory.

VCS architecture

So putting the above altogether we get:

- Agents monitor resources on each system and provide status to HAD on the local system
- HAD on each system send status information to GAB
- GAB broadcasts configuration information to all cluster members
- LLT transports all cluster communications to all cluster nodes
- HAD on each node takes corrective action, such as failover, when necessary

Service Groups

There are three types of service groups:

- **Failover** – The service group runs on one system at any one time.
- **Parallel** – The service group can run simultaneously on more than one system at any time.
- **Hybrid** – A hybrid service group is a combination of a failover service group and a parallel service group used in VCS 4.0 replicated data clusters, which are based on Veritas Volume Replicator.

When a service group appears to be suspended while being brought online you can **flush** the service group to enable

corrective action. Flushing a service group stops VCS from attempting to bring resources online or take them offline and clears any internal wait states.

Resources

Resources are objects that related to hardware and software, VCS controls these resources through these actions:

- Bringing resource online (starting)
- Taking resource offline (stopping)
- Monitoring a resource (probing)

When you link a parent resource to a child resource, the dependency becomes a component of the service group configuration. You can view the dependencies at the bottom of the **main.cf** file.

LLT and GRAB

VCS uses two components, LLT and GAB to share data over the private networks among systems.
These components provide the performance and reliability required by VCS.

LLT	LLT (Low Latency Transport) provides fast, kernel-to-kernel comms and monitors network connections. The system admin configures the LLT by creating a configuration file (llttab) that describes the systems in the cluster and private network links among them. The LLT runs in layer 2 of the network stack
GAB	GAB (Group membership and Atomic Broadcast) provides the global message order required to maintain a synchronised state among the systems, and monitors disk comms such as that required by the VCS heartbeat utility. The system admin configures GAB driver by creating a configuration file (gabtab).

LLT and GAB files

/etc/llthosts	The file is a database, containing one entry per system, that links the LLT

	system ID with the hosts name. The file is identical on each server in the cluster.
/etc/llttab	The file contains information that is derived during installation and is used by the utility lltconfig.
/etc/gabtab	The file contains the information needed to configure the GAB driver. This file is used by the gabconfig utility.
/etc/VRTSvcs/conf/config/main.cf	The VCS configuration file. The file contains the information that defines the cluster and its systems.

Gabtab Entries

```
/sbin/gabdiskconf – i /dev/dsk/c1t2d0s2 -s 16 -S 1123
/sbin/gabdiskconf – i /dev/dsk/c1t2d0s2 -s 144 -S 1124
/sbin/gabdiskhb -a /dev/dsk/c1t2d0s2 -s 16 -p a -s 1123
/sbin/gabdiskhb -a /dev/dsk/c1t2d0s2 -s 144 -p h -s 1124
/sbin/gabconfig -c -n2
```

gabdiskconf	-i Initialises the disk region -s Start Block -S Signature
gabdiskhb (heartbeat disks)	-a Add a gab disk heartbeat resource -s Start Block -p Port -S Signature
gabconfig	-c Configure the driver for use -n Number of systems in the cluster.

LLT and GAB Commands

Verifying that links are active for LLT	lltstat -n
verbose output of the	lltstat -nvv | more

lltstat command	
open ports for LLT	lltstat -p
display the values of LLT configuration directives	lltstat -c
lists information about each configured LLT link	lltstat -l
List all MAC addresses in the cluster	lltconfig -a list
stop the LLT running	lltconfig -U
start the LLT	lltconfig -c
verify that GAB is operating	gabconfig -a Note: port a indicates that GAB is communicating, port h indicates that VCS is started
stop GAB running	gabconfig -U
start the GAB	gabconfig -c -n <number of nodes>
override the seed values in the gabtab file	gabconfig -c -x

GAB Port Memberbership

List Membership	gabconfig -a
Unregister port f	/opt/VRTS/bin/fsclustadm cfsdeinit
Port Function	a gab driver b I/O fencing (designed to guarantee data integrity) d ODM (Oracle Disk Manager) f CFS (Cluster File System) h VCS (VERITAS Cluster Server: high availability daemon) o VCSMM driver (kernel module needed for Oracle and VCS interface) q QuickLog daemon v CVM (Cluster Volume Manager) w vxconfigd (module for cvm)

Cluster daemons

High Availability	had

Daemon	
Companion Daemon	hashadow
Resource Agent daemon	<resource>Agent
Web Console cluster managerment daemon	CmdServer

Cluster Log Files

Log Directory	/var/VRTSvcs/log
primary log file (engine log file)	/var/VRTSvcs/log/engine_A.log

Starting and Stopping the cluster

“-stale” instructs the engine to treat the local config as stale “-force” instructs the engine to treat a stale config as a valid one	hastart [-stale|-force]
Bring the cluster into running mode from a stale state using the configuration file from a particular server	hasys -force <server_name>
stop the cluster on the local server but leave the application/s running, do not failover the application/s	hastop -local
stop cluster on local server but evacuate (failover) the application/s to another node within the cluster	hastop -local -evacuate
stop the cluster on all nodes but leave the application/s running	hastop -all -force

Cluster Status

display cluster summary	hastatus -summary
continually monitor cluster	hastatus
verify the cluster is operating	hasys -display

Cluster Details

information about a cluster	haclus -display
value for a specific cluster attribute	haclus -value <attribute>
modify a cluster attribute	haclus -modify <attribute name> <new>
Enable LinkMonitoring	haclus -enable LinkMonitoring
Disable LinkMonitoring	haclus -disable LinkMonitoring

Users

add a user	hauser -add <username>
modify a user	hauser -update <username>
delete a user	hauser -delete <username>
display all users	hauser -display

System Operations

add a system to the cluster	hasys -add <sys>
delete a system from the cluster	hasys -delete <sys>
Modify a system attributes	hasys -modify <sys> <modify options>
list a system state	hasys -state
Force a system to start	hasys -force
Display the systems attributes	hasys -display [-sys]
List all the systems in the cluster	hasys -list
Change the load attribute of a system	hasys -load <system> <value>
Display the value of a systems nodeid (/etc/llthosts)	hasys -nodeid
Freeze a system (No	hasys -freeze [-persistent][-

offlining system, No groups onlining)	evacuate] Note: main.cf must be in write mode
Unfreeze a system (reenable groups and resource back online)	hasys -unfreeze [-persistent] Note: main.cf must be in write mode

Dynamic Configuration

The VCS configuration must be in read/write mode in order to make changes. When in read/write mode the configuration becomes stale, a .stale file is created in $VCS_CONF/conf/config. When the configuration is put back into read only mode the .stale file is removed.

Change configuration to read/write mode	haconf -makerw
Change configuration to read-only mode	haconf -dump -makero
Check what mode cluster is running in	haclus -display |grep -i 'readonly' 0 = write mode 1 = read only mode
Check the configuration file	hacf -verify /etc/VRTSvcs/conf/config Note: you can point to any directory as long as it has main.cf and types.cf
convert a main.cf file into cluster commands	hacf -cftocmd /etc/VRTSvcs/conf/config -dest /tmp
convert a command file into a main.cf file	hacf -cmdtocf /tmp -dest /etc/VRTSvcs/conf/config

Service Groups

add a service group	haconf -makerw hagrp -add groupw hagrp -modify groupw SystemList sun1 1 sun2 2 hagrp -autoenable groupw -sys sun1 haconf -dump -makero
delete a service group	haconf -makerw

	hagrp -delete groupw haconf -dump -makero
change a service group	haconf -makerw hagrp -modify groupw SystemList sun1 1 sun2 2 sun3 3 haconf -dump -makero Note: use the “hagrp -display <group>” to list attributes
list the service groups	hagrp -list
list the groups dependencies	hagrp -dep <group>
list the parameters of a group	hagrp -display <group>
display a service group’s resource	hagrp -resources <group>
display the current state of the service group	hagrp -state <group>
clear a faulted non-persistent resource in a specific grp	hagrp -clear <group> [-sys] <host> <sys>
Change the system list in a cluster	# remove the host hagrp -modify grp_zlnrssd SystemList -delete <hostname> # add the new host (don’t forget to state its position) hagrp -modify grp_zlnrssd SystemList -add <hostname> 1 # update the autostart list hagrp -modify grp_zlnrssd AutoStartList <host> <host>

Service Group Operations

Start a service group and bring its resources online	hagrp -online <group> -sys <sys>
Stop a service group and takes its resources offline	hagrp -offline <group> -sys <sys>
Switch a service group from system to another	hagrp -switch <group> to <sys>

Enable all the resources in a group	hagrp -enableresources <group>
Disable all the resources in a group	hagrp -disableresources <group>
Freeze a service group (disable onlining and offlining)	hagrp -freeze <group> [-persistent] note: use the following to check “hagrp -display <group> | grep TFrozen”
Unfreeze a service group (enable onlining and offlining)	hagrp -unfreeze <group> [-persistent] note: use the following to check “hagrp -display <group> | grep TFrozen”
Enable a service group. Enabled groups can only be brought online	haconf -makerw hagrp -enable <group> [-sys] haconf -dump -makero Note to check run the following command “hagrp -display | grep Enabled”
Disable a service group. Stop from bringing online	haconf -makerw hagrp -disable <group> [-sys] haconf -dump -makero Note to check run the following command “hagrp -display | grep Enabled”
Flush a service group and enable corrective action.	hagrp -flush <group> -sys <system>

Resources

add a resource	haconf -makerw hares -add appDG DiskGroup groupw hares -modify appDG Enabled 1 hares -modify appDG DiskGroup appdg hares -modify appDG StartVolumes 0 haconf -dump -makero
delete a resource	haconf -makerw hares -delete <resource> haconf -dump -makero
change a resource	haconf -makerw hares -modify appDG Enabled 1

	haconf -dump -makero Note: list parameters “hares -display <resource>”
change a resource attribute to be globally wide	hares -global <resource> <attribute> <value>
change a resource attribute to be locally wide	hares -local <resource> <attribute> <value>
list the parameters of a resource	hares -display <resource>
list the resources	hares -list
list the resource dependencies	hares -dep

Resource Operations

Online a resource	hares -online <resource> [-sys]
Offline a resource	hares -offline <resource> [-sys]
display the state of a resource(offline, online, etc)	hares -state
display the parameters of a resource	hares -display <resource>
Offline a resource and propagate the command to its children	hares -offprop <resource> -sys <sys>
Cause a resource agent to immediately monitor the resource	hares -probe <resource> -sys <sys>
Clearing a resource (automatically initiates the onlining)	hares -clear <resource> [-sys]

Resource Types

Add a resource type	hatype -add <type>
Remove a resource type	hatype -delete <type>
List all resource types	hatype -list

Display a resource type	hatype -display <type>
List a partitcular resource type	hatype -resources <type>
Change a particular resource types attributes	hatype -value <type> <attr>

Resource Agents

add a agent	pkgadd -d . <agent package>
remove a agent	pkgrm <agent package>
change a agent	n/a
list all ha agents	haagent -list
Display agents run-time information i.e has it started, is it running ?	haagent -display <agent_name>
Display agents faults	haagent -display |grep Faults

Resource Agent Operations

Start an agent	haagent -start <agent_name>[-sys]
Stop an agent	haagent -stop <agent_name>[-sys]

Veritas Cluster Tasks

Create a Service Group

hagrp -add groupw
hagrp -modify groupw SystemList sun1 1 sun2 2
hagrp -autoenable groupw -sys sun1

Create a disk group resource , volume and filesystem resource

We have to create a disk group resource, this will ensure that the disk group has been imported before we start any volumes
hares -add appDG DiskGroup groupw
hares -modify appDG Enabled 1
hares -modify appDG DiskGroup appdg
hares -modify appDG StartVolumes 0
Once the disk group resource has been created we can create the volume resource
hares -add appVOL Volume groupw
hares -modify appVOL Enabled 1
hares -modify appVOL Volume app01
hares -modify appVOL DiskGroup appdg

Now that the volume resource has been created we can create the filesystem mount resource

hares -add appMOUNT Mount groupw
hares -modify appMOUNT Enabled 1
hares -modify appMOUNT MountPoint /apps
hares -modify appMOUNT BlockDevice /dev/vx/dsk/appdg/app01
hares -modify appMOUNT FSType vxfs

To ensure that all resources are started in order, we create dependencies against each other

hares -link appVOL appDG
hares -link appMOUNT appVOL

Create a application resource

Once the filesystem resource has been created we cab add a application resource, this will start, stop and monitor the application.

hares -add sambaAPP Application groupw
hares -modify sambaAPP Enabled 1
hares -modify sambaAPP User root
hares -modify sambaAPP StartProgram "/etc/init.d/samba start"
hares -modify sambaAPP StopProgram "/etc/init.d/samba stop"
hares -modify sambaAPP CleanProgram "/etc/init.d/samba clean"
hares -modify sambaAPP PidFiles
"/usr/local/samba/var/locks/smbd.pid"
"/usr/local/samba/var/locks/nmbd.pid"
hares -modify sambaAPP MonitorProcesses "smbd -D" "nmdb -D"

Create a single virtual IP resource

create a single NIC resource

hares -add appNIC NIC groupw
hares -modify appNIC Enabled 1
hares -modify appNIC Device qfe0

Create the single application IP resource

hares -add appIP IP groupw
hres -modify appIP Enabled 1
hres -modify appIP Device qfe0
hres -modify appIP Address 192.168.0.3
hres -modify appIP NetMask 255.255.255.0
hres -modify appIP IfconfigTwice 1

Create a multi virtual IP resource

Create a multi NIC resource

hares -add appMultiNICA MultiNICA groupw
hares -local appMultiNICA Device
hares -modify appMulitNICA Enabled 1
hares -modify appMulitNICA Device qfe0 192.168.0.3 qfe1
192.168.0.3 -sys sun1 sun2
hares -modify appIPMultiNIC NetMask 255.255.255.0
hares -modify appIPMultiNIC ArpDelay 5
hares -modify appIPMultiNIC IfconfigTwice 1

Create the multi Ip address resource, this will monitor the virtual IP addresses.

```
hares -add appIPMultiNIC IPMultiNIC groupw
hares -modify appIPMultiNIC Enabled 1
hares -modify appIPMultiNIC Address 192.168.0.3
hares -modify appIPMultiNIC NetMask 255.255.255.0
hares -modify appIPMultiNIC MultiNICResName appMultiNICA
hares -modify appIPMultiNIC IfconfigTwice 1
```

Clear resource fault

```
# hastatus -sum
— SYSTEM STATE
— System     State           Frozen
A sun1       RUNNING    0
A sun2       RUNNING    0
— GROUP STATE
— Group      System   Probed   AutoDisabled    State
B  groupw   sun1      Y          N                 OFFLINE
B  groupw   sun2      Y          N                 STARTING|PARTIAL
— RESOURCES ONLINING
— Group    Type     Resource          System    IState
E groupw   Mount    app02MOUNT   sun2       W_ONLINE

# hares -clear app02MOUNT
```

Flush a group

```
# hastatus -sum
— SYSTEM STATE
— System     State           Frozen
A sun1       RUNNING    0
A sun2       RUNNING    0
— GROUP STATE
— Group      System   Probed   AutoDisabled    State
B  groupw   sun1      Y          N                 STOPPING|PARTIAL
B  groupw   sun2      Y          N                 OFFLINE|FAULTED
— RESOURCES FAILED
— Group     Type      Resource          System
C groupw    Mount    app02MOUNT    sun2
— RESOURCES ONLINING
— Group      Type      Resource          System     IState
E groupw     Mount    app02MOUNT    sun1
       W_ONLINE_REVERSE_PROPAGATE
— RESOURCES OFFLINING
— Group      Type          Resource    System     IState
F groupw     DiskGroup   appDG       sun1
      W_OFFLINE_PROPAGATE
# hagrp -flush groupw -sys sun1
```

References*
http://www.datadisk.co.uk/
http://sort.symantec.com/documents
http://www.veritashowto.com/
http://www.veritashowto.com/
http://sort.symantec.com/
http://vos.symantec.com/public/documents/sf/5.0/solaris/pdf/vcs_users.pdf
http://www.cheat-sheets.org/

Netbackup
Master Server Daemons/Processes

Request daemon	bprd
Scheduler	bpsched (started with bprd)
Netbackup database manager	bpdbm (started with bpsched)
Job Monitor	bpjobd (started with bpdbm)

Media Server Daemons/Processes

Communications daemon	bpcd
Backup and restore manager	bpbrm (started with bpcd)
Tape Manager	bptm (started with bpbrm)
Disk Manager	bpdm (started with bpbrm)
Media Manager	ltid
Bar code reader	avrd (started with ltid)
Remote device management/ controls volume database	vmd (started with ltid)
Roboticdaemon (one on each media server) talks to tldcd	tldd (started with ltid)
Robotic control daemon talks to the robot directl via scsi	tldcd (started with ltid)

Catalogs

Master Server	
Information about backed-up files	image – /opt/openv/netbackup/db
Storage Unit, Global Configuration, Catalog backup configuration.	config – /opt/openv/netbackup/db
Backup Policy information	class – /opt/openv/netbackup/db
Job status information	jobs – /opt/openv/netbackup/db
Netbackup logs with error and status information	error – /opt/openv/netbackup/db
Information on volumes, volume pools, scratch pool and volume groups	volume – /opt/openv/volmgr/database
Media Server	
Tracks assigned volumes (media that has data them)	media – /opt/openv/netbackup/db
Information about devices managed by the media server	device – /opt/openv/volmgr/database

Log and Information Files

Netbackup and Patch versions	/opt/openv/netbackup/bin/version
Media Version	/opt/openv/volmgr/version
Patch Level history	/opt/openv/netbackup/patch/patch.history
Buffer size	/opt/openv/netbackup/db/config/SIZE_DATA_BUF
Number of buffers	/opt/openv/netbackup/db/config/NUMBER_DATA
Network Buffer Size	/opt/openv/netbackup/NET_BUFFER_SZ (default =
Java GUI authorisation	/opt/openv/java/auth.conf
Catalog type	/opt/openv/netbackup/db/config/cat_format.cfg

(binary or ASCII)	
Netbackup and media manager parameter files	/opt/openv/netbackup/bp.conf /opt/openv/volmgr/vm.conf
Corrupt Database image files (5.0 and above)	/opt/openv/netbackup/db.corrupt

Server Commands

Check license details	/opt/openv/netbackup/bin/admincmd/get_license_
Start Netbackup	netbackup start /opt/openv/netbackup/bin/initbprd (master) /opt/openv/volmgr/bin/vmd (media)
Stop Netbackup (does not disconnect GUI sessions)	netbackup stop /opt/openv/netbackup/bin/admincmd/bprdreq -terminate (master) /opt/openv/netbackup/bin/bpdbm -terminate (master)
Stop Netbackup and kill all GUI sessions	/opt/openv/netbackup/bin/goodies/bp.kill_all
Start the GUI	/opt/openv/netbackup/bin/jnbSA
Scan for tape devices	sgscan (solaris) ioscan (HPUX)
Display all Netbackup processes	bpps -a

lists servers errors	bperror -U -problems -hoursago <number of hours bperror -U -backstat -by_statcode -hoursago <num of hours>
display information on a error code	bperror -statuscode <statuscode> [-recommendati
Reread bp.conf file without stopping Netbackup	bprdreq -rereadconfig
Check database consistency	bpdbm -consistency 1 bpdbm -consistency 2 Check for the below lines: Bad image header Does not exist
Netbackup Recovery	
Device catalog is intact	bprecover -l -m <media ID> -d dlt (listing) bprecover -r -m <media ID> -d dlt (recovering)
Device catalog is gone or corrupted	bprecover -l -tpath <tape_path> (listing) bprecover -r -tpath <tape_path> (recovering)
Disk backups	bprecover -l -dpath <disk_path> (listing) bprecover -r -dpath <disk_path> (recovering)

Volume Commands

Tape Drive and Inventory Commands	
List drive status, detail drive info and pending requests	vmoprcmd
List the tape drive status	vmoprcmd -d ds
List the pending requests	vmoprcmd -d pr
Control a tape device	vmoprcmd [-reset][-up][-down] <drive number>
List all changes in the	vmupdate -recommend -rt tld -rn 0

robot(but do not update)	vmcheckxxx -rt tld -rn 0 -recommend
Empty the robot and re-inventory (using barcodes)	vmupdate -rt tld -rn <robot number> -rh <silo slave> -vh <host> -nostderr -use_barcode_rules -use_seed -empty_ie
Tape Media Commands	
List all pools	vmpool -listall -bx
List tapes in pool	vmquery -pn <pool name> -bx
List all tapes in the robot	vmquery -rn 0 -bx |grep 'TLD' | sort +4
List cleaning tapes	vmquery -mt dlt_clean -bx
List tape volume details	vmquery -m <media ID>
Delete a volume from the catalog	vmdelete -m <media ID>
Change a tapes expiry date	vmchange -exp 12/31/06 23:59:58 -m <media ID>
Change a tape's media pool	vmchange -p <pool number> -m <media ID>

Media commands

List the storage units	bpstulist -U
Freeze or unfreeze media	bpmedia [-freeze][-unfreeze] -ev <media ID>
List media details	bpmedialist -ev <media ID>
List media contents	bpmedialist -U mcontents -m <media ID>
List backup Image Information	bpimagelist -backupid <image ID>
Expire client images	bpimage -cleanup -allclients
Expire a tape	bpexpdate -d 0 -ev <media ID> -force
List all netbackups jobs	bpdbjobs -report [-hoursago]
Move media from one media server to another	bpmedia -movedb -newserver <media server> -oldserver <media server>

Tape/Robot commands

List tape drives	tpconfig -d
List cleaning times on drives	tpclean -L
clean a drive	tpclean -C <drive number>
change a drives cleaning frequency	tpclean -F <drive> <frequency>
set a drives cleaning time to zero	tpclean -M <drive>
Move tapes within robot using robtest	robtest commands that can be used are as follows: s s (show slots) s d (show drives) s i (show load port) m s250 d5 (move tape from slot 250 into drive 5) uload d5 (unload tape from drive 5) m d5 s250 (move tape from drive 5 to slot 250) m s250 i1 (mov tape from slot 250 to load port 1)
List load port tapes	echo “s i q” | tldtest -r /dev/sg/c0t4l0
List all slot contents	echo “s s q” | tldtest -r /dev/sg/c0t4l0
List tape drive contents	echo “s d q” | tldtest -r /dev/sg/c0t4l0
Move a tape in s100 to drive 1	echo “m s100 d1” | tldtest -r /dev/sg/c0t4l0
Move a tape to load port 1	echo “m s100 i1” | tldtest -r /dev/sg/c0t4l0

Archiving Commands

list archive info	bpcatlist -client all -before Jul 01 2006 bpcatlist -client all -before Aug 01 2006
archive and remove images	bpcatlist -before Jul 01 2006 | bpcatarc | bpcatrm
restore archive files	bpcatlist -before Jul 01 2006 | bpcatres

Client commands

test client connectivity	bpclntcmd [-ip <ip addres>] bpclntcmd [-hn <hostname>] bpclntcmd [-pn] bpclntcmd [-sv]

Basic Veritas Cluster Server Troubleshooting

Troubleshooting VCS startup

http://sfdoccentral.symantec.com/sf/5.0/hpux/html/vcs_users/ch_vcs_troubleshooting9.html
The setup: Your site is down. It's a small cluster configuration with only two nodes and redundant nic's, attached network disk, etc. All you know is that the problem is with VCS (although it's probably indirectly due to a hardware issue). Something has gone wrong with VCS and it's, obviously, not responding correctly to whatever terrible accident of nature has occurred. You don't have much more to go on than that. The person you receive your briefing from thinks the entire clustered server set up (hardware, software, cabling, power, etc) is a bookmark in IE 😉

1. Check if the cluster is working at all.

Log into one of the cluster nodes as root (or a user with equivalent privilege – who shouldn't exist 😉 and run

host1 # hastatus –summary

or

host1 # hasum <– both do the same thing, basically

Ex:

host1 # hastatus -summary

— SYSTEM STATE
— System State Frozen

A host1 RUNNING 0
A host2 RUNNING 0

— GROUP STATE
— Group System Probed AutoDisabled State

B ClusterService host1 Y N OFFLINE
B ClusterService host2 Y N ONLINE
B SG_NIC host1 Y N ONLINE
B SG_NIC host2 Y N OFFLINE
B SG_ONE host1 Y N ONLINE
B SG_ONE host2 Y N OFFLINE
B SG_TWO host1 Y N OFFLINE
B SG_TWO host2 Y N OFFLINE

Clearly, your situation is bad: A normal VCS status should indicate that all nodes in the cluster are “RUNNING” (which these are). However, it should also show all service groups as being ONLINE on at least one of the nodes, which isn’t the case above with SG_TWO (Service Group 2).

2. Check for cluster communication problems. Here we want to determine if a service group is failing because of any heartbeat failure (The VCS cluster, that is, not another administrator 😉

Check on GAB first, by running:

host1 # gabconfig -a

Ex:

host1 # gabconfig -a
GAB Port Memberships
==========
==========
Port a gen 3a1501 membership 01
Port h gen 3a1505 membership 01

This output is okay. You would know you had a problem at this point if any of the following conditions were true:

if no port “a” memberships were present (0 and 1 above), this could indicate a problem with gab or llt (Looked at next)

If no port “h” memberships were present (0 and 1 above), this could indicate a problem with had.

If starting llt causes it to stop immediately, check your heartbeat cabling and llt setup.

Try starting gab, if it’s down, with:

host1 # /etc/init.d/gab start

If you’re running the command on a node that isn’t operational, gab won’t be seeded, which means you’ll need to force it, like so:

host1 # /sbin/gabconfig -x

3. Check on LLT, now, since there may be something wrong there (even though it wasn’t indicated above)

LLT will most obviously present as a crucial part of the problem if your “hastatus -summary” gives you a message that it “can’t connect to the server.” This will prompt you to check all cluster communication mechanisms (some of which we’ve already covered).

First, bang out a quick:

host1 # lltconfig

on the command line to see if llt is running at all.

If llt isn’t running, be sure to check your console, system messages file (syslog, possibly messages and any logs in /var/log/VRTSvcs/… – usually the “engine log” is worth a quick look) As a rule, I usually do

host1 # ls -tr

when I’m in the VCS log directory to see which log got written to last, and work backward from there. This puts the most recently updated file last in the listing. My assumption is that any pertinent errors got written to one of the fresher log files 🙂 Look in these logs for any messages about bad llt configurations or files, such as /etc/llttab, /etc/llthost and /etc/VRTSvcs/conf/sysname. Also, make

sure those three files contain valid entries that “match” <– This is very important. If you refer to the same facility by 3 different names, even though they all point back to the same IP, VCS can become addled and drop-the-ball.

Examples of invalid entries in LLT config files would include “node numbers” outside the range of 0 to 31 and “cluster numbers” outside the range of 0 to 255.

Now, if LLT “is” running, check its status, like so:

host # lltstat -wn <– This will let you know if llt on the separate nodes within the cluster can communicate with one another.

Of course, verify physical connections, as well. Also, see our previous post on dlpiping for more low-level-connection VCS troubleshooting tips.

Ex:

host1 # lltstat -vvn
LLT node information:
Node State Link Status Address
0 prsbn012 OPEN
ce0 DOWN
ce1 DOWN
HB172.1 UP 00:03:BA:9D:57:91
HB172.2 UP 00:03:BA:0E:F1:DE
HB173.1 UP 00:03:BA:9D:57:92
HB173.2 UP 00:03:BA:0E:D0:BE
1 prsbn015 OPEN
ce3 UP 00:03:BA:0E:CE:09
ce5 UP 00:03:BA:0E:F4:6B
HB172.1 UP 00:03:BA:9D:5C:69
HB172.2 UP 00:03:BA:0E:CE:08
HB173.1 UP 00:03:BA:0E:F4:6A
HB173.2 UP 00:03:BA:9D:5C:6A

host1 # cat /etc/llttab <– pardon the lack of low-pri links. We had to build this cluster on the cheap 😉

set-node /etc/VRTSvcs/conf/sysname
set-cluster 100
link ce0 /dev/ce:0 – ether 0x1051 –
link ce1 /dev/ce:1 – ether 0x1052 –
exclude 7-31
host1 # cat /etc/llthosts
0 host1

1 host2
host1 # cat /etc/VRTSvcs/conf/sysname
host1

If llt is down, or you think it might be the problem, either start it or restart it with:

host1 # /etc/init.d/llt.rc start

or

host1 # /etc/init.d/llt.rc stop
host1 # /etc/init.d/llt.rc start

And, that's where we'll end it today. There's still a lot more to cover (we haven't even given the logs more than their minimum due), but that's for next week.

« SF DocCentral

Product Guides: Veritas Cluster Server

Platform: Linux

Release: 6.0

Release Notes			
			Veritas Cluster Server Release Notes
Cluster Server Guides			
			Veritas Cluster Server Installation Guide
			Veritas Cluster Server Administrator’s Guide
			Veritas Cluster Server Bundled Agents Reference Guide
			Veritas Cluster Server Agent Developer’s Guide
			Virtual Business Service–Availability User’s Guide
Cluster Server Agent Guides			
			Veritas Cluster Server Agent for DB2 Installation and Configuration Guide
			Veritas Cluster Server Agent for Oracle Installation and Configuration Guide
			Veritas Cluster Server Agent for Sybase Installation and Configuration Guide

Reference:
http://sfdoccentral.symantec.com/sf/5.0/hpux/html/vcs_users/vcs_usersTOC.html
http://linuxshellaccount.blogspot.in/2008/11/basic-veritas-cluster-server.html
http://linuxshellaccount.blogspot.in/search?q=vcs

« PREVIOUS
ILOM setup on Sparc T4 server

NEXT »
ZFS Cheat sheet

BE THE FIRST TO COMMENT

Leave a Reply

Your email address will not be published.

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

Copyright © 2020 | WordPress Theme by MH Themes