

Veritas Cluster Cheat sheet

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

<code>/etc/llthosts</code>	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.
<code>/etc/llttab</code>	The file contains information that is derived during installation and is used by the utility lltconfig.
<code>/etc/gabtab</code>	The file contains the information needed to configure the GAB driver. This file is used by the gabconfig utility.
<code>/etc/VRTSvcs/conf/config/main.cf</code>	The VCS configuration file. The file contains the information that defines the cluster and its systems.

Gabtab Entries

```
/sbin/gabdiskconf -i /dev/dsk/clt2d0s2 -s 16 -S 1123
/sbin/gabdiskconf -i /dev/dsk/clt2d0s2 -s 144 -S 1124
/sbin/gabdiskhb -a /dev/dsk/clt2d0s2 -s 16 -p a -s 1123
/sbin/gabdiskhb -a /dev/dsk/clt2d0s2 -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 command	lltstat -nvv more
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 Membership

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 Daemon	had
Companion Daemon	hashadow
Resource Agent daemon	<resource>Agent
Web Console cluster management 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 offlining system, No groups online)	hasys -freeze [-persistent][-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	<pre>hacf -verify /etc/VRTS/conf/config</pre> <p>Note: you can point to any directory as long as it has main.cf and types.cf</p>
convert a main.cf file into cluster commands	<pre>hacf -cftocmd /etc/VRTS/conf/config -dest /tmp</pre>
convert a command file into a main.cf file	<pre>hacf -cmdtocf /tmp -dest /etc/VRTS/conf/config</pre>

Service Groups

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

Service Group Operations

Start a service group and bring its resources online	<pre>hagrp -online <group> -sys <sys></pre>
Stop a service group and takes its resources offline	<pre>hagrp -offline <group> -sys <sys></pre>
Switch a service group from system to another	<pre>hagrp -switch <group> to <sys></pre>
Enable all the resources in a group	<pre>hagrp -enableresources <group></pre>
Disable all the resources in a group	<pre>hagrp -disableresources <group></pre>
Freeze a service group (disable onlining and offlining)	<pre>hagrp -freeze <group> [-persistent]</pre> <p>note: use the following to check "hagrp -display <group> grep TFrozen"</p>
Unfreeze a service group (enable onlining and	<pre>hagrp -unfreeze <group> [-persistent]</pre>

offlining)	note: use the following to check "hagrp -display <group> grep TFrozen"
Enable a service group. Enabled groups can only be brought online	<pre>haconf -makerw hagrp -enable <group> [-sys] haconf -dump -makero</pre> <p>Note to check run the following command "hagrp -display grep Enabled"</p>
Disable a service group. Stop from bringing online	<pre>haconf -makerw hagrp -disable <group> [-sys] haconf -dump -makero</pre> <p>Note to check run the following command "hagrp -display grep Enabled"</p>
Flush a service group and enable corrective action.	<pre>hagrp -flush <group> -sys <system></pre>

Resources

add a resource	<pre>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</pre>
delete a resource	<pre>haconf -makerw hares -delete <resource> haconf -dump -makero</pre>
change a resource	<pre>haconf -makerw hares -modify appDG Enabled 1 haconf -dump -makero</pre> <p>Note: list parameters "hares -display <resource>"</p>
change a resource attribute to be globally wide	<pre>hares -global <resource> <attribute> <value></pre>
change a resource attribute to be locally wide	<pre>hares -local <resource> <attribute> <value></pre>
list the parameters of a resource	<pre>hares -display <resource></pre>
list the resources	<pre>hares -list</pre>
list the resource dependencies	<pre>hares -dep</pre>

Resource Operations

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

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 particular 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]