WHITE PAPER:
IMPLEMENTING SOLARIS ZONES WITH
VERITAS™ CLUSTER SERVER FROM SYMANTEC

Implementing Solaris Zones with Veritas™ Cluster Server from Symantec and Veritas™ Cluster file System High Availability 6.0 from Symantec

Who should read this paper

The audience for this document is system administrators who need best practices for configuring local zones in Veritas Cluster Server and for systems maintenance when local zones are placed under Veritas Cluster Server control. Moreover, provided within this guide will be the steps necessary to establish a Cluster File System High Availability-based cluster topology for both zone root and application data file systems.

Content

ACKNOWLEDGEMENTS

I would like to take this opportunity to acknowledge the contributing members of the SAMG team, specifically Eric Hennessey and James "Jax" Jackson. Both of whom without I would have not been able to complete the refresh of this document.

This content provided and the best practices put forth here were not exclusively developed in the vacuum of a lab environment. I would like to extend my sincere appreciation to not only the customer contributors but the Symantec Product Management, Engineering and Support teams, all of whose willingness to indulge my requests made this effort possible. You know who you are.

DOCUMENT CONSIDERATIONS

The comments and best practices included within this document assume a certain set of prerequisites to support the published capabilities and functionality. Along with those environmental factors this guide contains additional content specific considerations. These include but are not limited to the following:

- The operating environment is Solaris® 10 Update 8 or higher Scalable Processor Architecture (SPARC).
- All Cluster Nodes are physical hosts and not Oracle Virtual Machine and Logical Domain (VM's/LDoms). That said, most of the same considerations will also apply to configuring Zones relative to LDoms as opposed to physical servers.
- All Zone examples in this document will be of the "Whole Root" variety. For details on Whole Root vs. Sparse Zones, please consult Oracle® Zone Administration Guide. http://docs.oracle.com/cd/E19455-01/817-1592/fgotf/index.html
- Although both are supported with Veritas™ Cluster Server from Symantec, it is the assumption of this document that the "Shared-IP" as opposed to the "Exclusive-IP" model will be configured for network connectivity in each example.
- Zetabyte File System (ZFS) specific considerations will be addresses exclusively in the Appendix: Veritas Cluster Server, Local Zones and Native Volume Management.
- Although not explicitly covered, information regarding Solaris 11 will be provided only within the context of the ZFS appendix. Further elaboration will be incorporated at a later date.
- With Veritas Storage Foundation™ 6.0 certain VxFS commands are supported within the context of a non-global zone. However for the purposes of this document, that configuration option will not be covered.

INTRODUCTION

With the release of Solaris® 10, Sun Microsystems™ introduced the concept of a "Local Zone." Zones are an isolation mechanism for applications in which the application is executed within the confines of a zone. From an application view, the zone appears to be an independent system, where the application gets exclusive use of system resources, including processor and memory, as well as access to specific file systems without risk of interfering with other applications. From an implementation view, a local zone does not instantiate a separate OS kernel as is done with Virtual Machines or para-virtualization; but rather zones operate as resource "containers" with independent user control and file systems. In effect, zones themselves are an instance of the user space portion of the OS. Zones allow the system administrator to isolate an application and manage system resource allocation between applications running in other zones. Zones extend the concepts of resource management from simply controlling resource allocations between applications to more robust isolation, where one application cannot effect the operation of another. One feature that Solaris zones do share with the Virtual Machine concept is the notion of representing not only multiple instances but versions of the same operating system. This however is achieved through the use of branded Zones as opposed to individual kernel isolation.

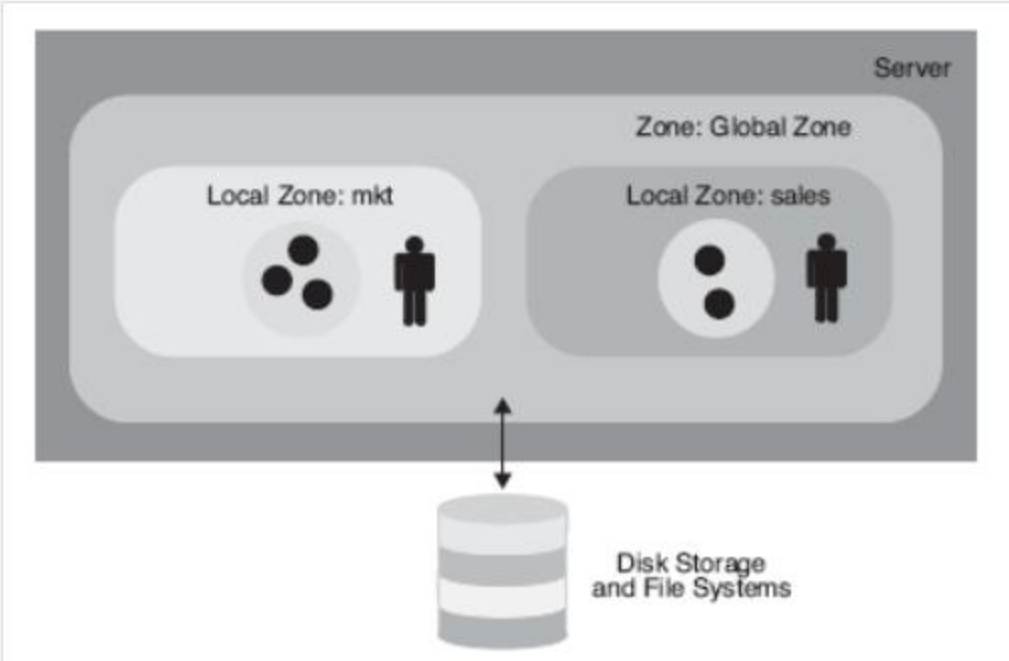


Figure 1 – Relationship of local zones to the global zone

For more information on zones and resource management, refer to the Oracle reference guide "System Administration Guide: Oracle® Solaris Containers-Resource Management and Oracle Solaris Zones" (September 2010, no. 817-1592).

http://docs.oracle.com/cd/E19253-01/817-1592/817-1592.pdf

Beginning with Veritas Cluster Server version 4.1, Veritas added support for Solaris zones in a clustered environment. This allows a system administrator to start, stop and monitor an application within the confines of a local zone, and failover zones between systems in a cluster.

The intent of this document is to provide systems administrators the information needed to correctly configure local zones in a Veritas Cluster Server cluster and to provide best practices for systems maintenance when local zones are placed under Veritas Cluster Server control. Moreover, provided within this guide will be the steps necessary to establish a Cluster File System High Availability-based cluster topology for both zone root and application data file systems. Best practices noted in the document will be preceded by this symbol: ♦ By the same token, it will attempt to address configurations choices that should be carefully taken into consideration. Such topics will be denoted with the following symbol: ♣

What this document will not cover however is implementing Veritas Storage Foundation™ for Oracle® RAC from Symantec within non-global zones. For more information on Storage Foundation for Oracle RAC support for non-global zones please refer to the following README documentation:

https://www-secure.symantec.com/connect/sites/default/files/sfrac_appnote_zones_51sp1rp2_sol.pdf

It is assumed that the audience for this document holds prior knowledge of, and experience with, managing Veritas Cluster Server as well as Solaris zones.

INTERACTION BETWEEN VERITAS CLUSTER SERVER AND SOLARIS LOCAL ZONES

In a Solaris environment, the Veritas Cluster Server daemon / High Availability Daemon (HAD) will always run within the context of global/ root zone of the Solaris 10 OS. Traditionally Veritas Cluster Server has the capability to manage applications running in the global zone in what can be considered a "classic" way of controlling cluster resources.

By introducing the Local Zone/Container framework, Veritas Cluster Server now has the ability to manage the start and stop of the local zones themselves as well as the constituent applications within the zones. Effectively treating the zone as it would any other application resource.

After introducing the Zone agent to Veritas Cluster Server, it became apparent that support for physical to virtual failovers would be a valuable configuration option. Starting in version 5.1 a complete redesign of the Zone framework for Veritas Cluster Server was introduced to support this particular use case as well as simplify enablement.

VERITAS CLUSTER SERVER AGENT FRAMEWORK CHANGES FOR 5.1

The Veritas Cluster Server Agent Framework is a core set of functions that is compiled into every agent. The agent is responsible for connecting with the Veritas Cluster Server engine (HAD) and carrying out core agent logic. The Veritas Cluster Server agent framework first introduced the concept of Container Name with the release of 4.1. In subsequent releases, specifically 5.1, Symantec redesigned how Cluster Server is configured to support virtual environments for both Solaris and Advanced Interactive Executive (AIX). With the release of version 5.1, the defining attributes for Container Name and Container Type (i.e. Zone or WPar for AIX) have been either modified or moved entirely to the Service Group layer. By doing so, this now allows for the configuration of application resources that can failover between hosts irrespective of whether the environment is virtualized or not. This by definition enables P-2-V (Physical to Virtual) cluster topologies that otherwise would be in many cases unnecessarily complex to deploy.

To effectively support this modification, the Veritas Cluster Server 5.1 agent framework has introduced a new service group attribute entitled "ContainerInfo." This attribute, when configured, defines the *Type* (Zone or Wpar), *Name* and *Enabled* (0, 1 or 2). Designed to work in conjunction with the newly added resource type attribute "ContainerOpts," these two configurable attributes provide a comprehensive and flexible framework to support all manner of application and zone monitoring requirements.

For specific details on the various attribute settings, please refer to the Storage FoundationHigh Availability Virtualization Guide for Solaris.

http://www.symantec.com/business/support/resources/sites/BUSINESS/content/live/DOCUMENTATION/5000/DOC5405/en_US/sfha_virtualization_60_sol.pdf

It is also worth noting that with the release of Veritas Cluster Server 5.1 SP1, Symantec has introduced the concept of an Intelligent Monitoring Framework (IMF) or Kernel Level Monitoring. The purpose of this evolution past the traditional polling agent methodology is intended to first provide for instantaneous notification of application state changes and subsequently reduce the overall demands placed upon the resources of an individual host by the multiple instances of our traditional monitoring agents. When coupled with the use of Cluster File System, you are able to dramatically reduce not only the overhead on the system but the time required to failover an application or zone between physical hosts. With version 6.0, support for the Zone agent was added to the list of IMF aware resource types. Please note, whereas with 5.1SP1, enabling IMF was a manual procedure, starting in 6.0 all IMF supported resource types will have this setting enabled by default. With the enabling of the IMF attribute for the Zone agent, a nearly 80 percent performance improvement with regard to CPU load was achieved (for online zones). To see a list of the agents supported by IMF, you can run the following command: **#> haimfconfig -display**

VERITAS CLUSTER SERVER RESOURCE & RESOURCE TYPE CHANGES

Along with agent framework modifications, several core agents have been modified or created that supports the ContainerOpts attribute settings. These are described below.

Zone Agent

With the release of versions 5.1 of Veritas Cluster Server, the bundled Zone agent has been modified to support the ContainerInfo Service Group attribute. Whereas previous releases of Veritas Cluster Server required that the Zone or "Container" name be defined as part of the Zone agent itself, users are no longer required to define that attribute. Rather, the Zone agent now simply references the string assigned to the "Name" key within the ContainterInfo service group attribute.

♣ **Important note:** With the release of Solaris 10 8/07 (Update 4), Sun introduced two new functions to the online and offline operation associated with local zones. Users can now choose to either attach or detach a zone as part of the process for moving (or decoupling) a zone between physical global zones. This plays a significant role when determining the patching strategy particularly if zones are the predominant virtual environment being deployed.

When a zone is shutdown, it is the default behavior for packages that contain the following attribute to boot all offline Zones prior to adding the package: **SUNW_ALL_ZONES=true**. This setting is defined within the package's pkginfo file. However, should the zone be detached, the pkgadd operation will NOT consider the zone eligible for booting. The following output of the zoneadm command identifies whether a zone is attached, detached as well as online or offline.

#> zoneadm list –civ

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	native	shared
1	calzone	running	/zones/calzone/base	native	shared

When a zone is fully booted zoneadm will report its status as ***"running"*** while a zone that is attached but not booted will report its status as being ***"installed."*** For those local zones that are shutdown and detached the zoneadm command with the –civ flags will indicate which zones are in the ***"configured***" state.

The entry points for the Zone agent (online, offline, monitor & clean) are executed using a variation of the zoneadm boot, zoneadm halt and zoneadm list commands. Additionally, the options for detach and attach (-f) have been incorporated to the online and offline procedures.

Veritas Cluster Server by default will attach the zone on boot using the –f force option and detach the zone when taken offline. These setting however can be modified with the "DetachZonePath" and " ForceAttach" attributes. If overridden, this setting will cause the offline operation to leave the zone in the attached state. Each of these settings is resource specific and therefore unique to each instance of the Zone Agent.

Networking Agents

If you chose, you can enable the attribute "ExclusiveIPZone" for resources of type IP and NIC when these resources are configured to manage the IP and the NIC inside an exclusive-IP zone. This attribute is disabled by default. The IP agent and the NIC agent assume the native zone behavior (shared-IP). This however is no longer the case with Solaris 11 as exclusive IP (vnic) is the default setting.

Veritas Cluster Server brings IP resources online in the global zone by default. If you want to bring these resources online inside the exclusive-IP zone, perform the following tasks:

- Make sure that the resource is in a service group that has valid ContainerInfo attribute value configured.
- Set the value of the ExclusiveIPZone attribute to 1.

Note: The exclusive-IP zone supports the IP and NIC networking agents. For more information about these agents, see the *Veritas Cluster Server Bundled Agents Reference Guide*
http://www.symantec.com/business/support/resources/sites/BUSINESS/content/live/DOCUMENTATION/5000/DOC5233/en_US/vcs_bundled_agents_60_sol.pdf

Other bundled agents

Along with the bundled Zone agent, all remaining bundled agents have been modified to support the use of the ContainerOpts attribute: Additionally the following replication agents have all been made container aware:

- Oracle Data Guard

CLUSTER FILE SYSTEM HIGH AVAILABILITY OVERVIEW

Cluster File System High Availability combines the industry leading Storage Foundation High Availability product set with the extended capability of Clustered Volume Manager and Cluster File System. Together these tools allow for users to mount a Veritas File System (VxFS) on up to 64 Nodes concurrently for the supported UNIX and Linux operating systems (Solaris, AIX, RedHat® Enterprise Linux (RHEL), SUSE® Linux and HP-UX®).

Veritas Storage Foundation Cluster File System is the cornerstone of a highly available environment, delivering faster recovery for business-critical applications –– with reduced complexity and costs. Compared to traditional single-instance file system implementations, Cluster File System significantly reduces application downtime and improves data access. Because multiple servers share the same storage resources, the length of time it takes to bring storage online is drastically reduced when an application fails. Its high performance file system spans heterogeneous servers to provide concurrent access to data and enables faster failover of applications and databases, such as Oracle, SAP or Tibco.

Additionally Cluster File System provides the parallel data access necessary to allow for nearly instantaneous provisioning of new virtual environments. These can include Solaris Zones, Oracle VM's for SPARC (LDoms) as well as Red Hat KVM Virtual Machines. Further elaboration of this topic will be covered later on in this document.

FILE SYSTEM SUPPORT FOR ZONES

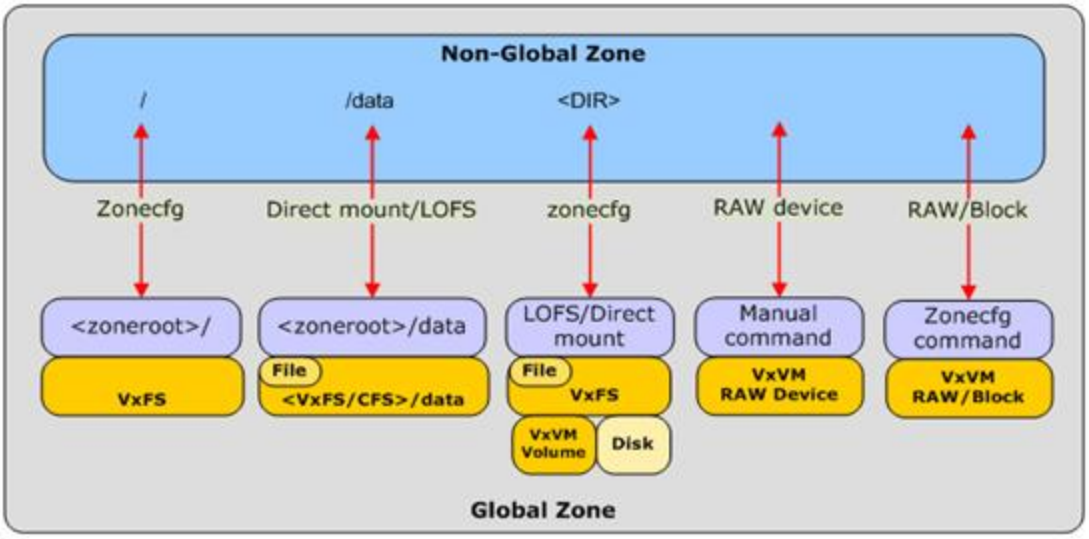


Figure 2 - File System Options for Solaris Zones

Depending on the requirement for the application being made highly available within the zone, or simply just the zone itself, Solaris 10 supports a wide variety of file system configuration options (as shown above in Figure 2) Elaboration of the various VxFS, Cluster File System and ZFS combinations are detailed in the following sections.

Figure 3 outlines the available combinations of the zone root path, application data and the unit of failover (App or Zone). For those configurations where the unit of failover is the application, the examples provided call for 2 independent zones, neither of which will be decoupled or moved from their corresponding global zone.

Zone Root	Zone Root Storage Location	Application Data	LOFS or Direct Mount	Unit of Failover
ZFS	Local Storage (SAN or DAS)	ZFS	Direct Mount	Application
ZFS	Shared/SAN	ZFS	Either	Zone
ZFS or VxFS	Local Storage (SAN or DAS)	VxFS	Direct Mount	Application
ZFS or VxFS	Shared/SAN	VxFS	Either	Zone
ZFS or VxFS	Local Storage (SAN or DAS)	Cluster File System	Either	Application
Cluster File System	Shared/SAN	Cluster File System	Either	Zone
Cluster File System*	Shared/SAN	Cluster File System	Either	Application

Figure 3 – Available combinations of Zone Root and Application Data Storage:

*This option although allowing for the application to failover rather than the Zone, calls for a single Zone Root to be concurrently booted on multiple Physical Nodes/Global Zones and will not be covered in this guide.

Although the number of combinations for file system layout for both Zone root and data volumes is numerous, the decision on which method to use –in most cases- is based on one or more of the following considerations:

- Required Unit of Failover (Zone or Application)
- Tolerance for Patching Outages
- User Account Control
- Is there a need to reduce the number of Individual OS environments?
- SAN Impact for increase in virtual environments.

Sample Cluster Topology

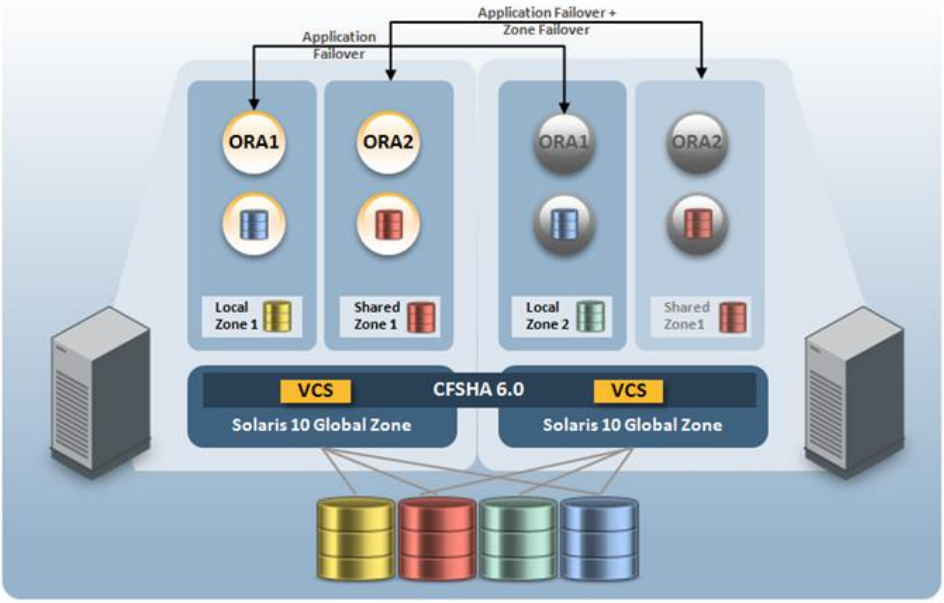


Figure 4 - Cluster Topology when using CFSHA with Solaris Zones

Best Practices For Local Zone Configuration in Veritas Cluster Server

Choosing the Location for the Zone Root File System

Veritas Cluster Server supports placement of the zone root file system on either shared or local storage. The advantage of placing the zone root file system on shared storage is that the zone installation must be performed only once. This does have a disadvantage however when it comes time to apply system patches. This topic is described in more detail in Appendix C.

There are number of additional considerations when deciding the placement of zone root file systems. For the purposes of this document however, the most important scenarios to consider will be the following:

1. Shared Zone Root with Zone and Application Failover
2. Local Zone Root with Application Failover Only

In the first example, a non-global zone will be essentially portable and subject to patching considerations when detached and attached to a different global zone/physical host. This can be advantageous if your concern is in the number of operating/user environments and keeping them to a minimum. Also with the use of Cluster File System, you can completely remove the delay associated with deporting and importing the zone root and application storage. One additional advantage here is in the ability to create a "Golden Zone Image" (one that has not been

configured) and use Flashsnap to provision new environments nearly instantaneously (This is covered in Appendix B). The disadvantage however is that you must be cognizant of the patch level on each global zone and avoid patch drift between the global and non-global zones. To some degree this has been addressed with the addition of the Update on Attach function from Solaris 10. It is important though to note that this feature is not supported by the Veritas Cluster Server Zone agent and will not covered in this document.

The second scenario will mitigate all patching concerns as the zone root itself is not "portable" but rather localized to the physical host. This can still be a SAN attached VxFS file system but not one that would be deported on imported on a regular basis. The Veritas Cluster Server configuration here would simply localize the Zone Name attribute in the ContainerInfo property for the service group to each respective global zone and thus only the application would failover. This option does however require an independent zone for each physical host and as such you would be required to maintain user accounts, application installs and DNS entries accordingly. You can consider this scenario more analogous to traditional physical host clustering.

Application IP Address Configuration

Solaris supports the ability to configure an application's virtual IP address within the zone configuration. At first glance, this appears to be an effective method of managing the virtual IP, since the IP will be brought up and down along with the local zone. However, doing this has the following disadvantages:

- Since the IP isn't being monitored, IP-related faults won't be detected.

It also worth noting that when cloning a zone root, having the configuration as generic as possible makes for simpler provisioning.

♦ Veritas Cluster Server supports local zones configured with or without the virtual IP, but given the disadvantages mentioned above, best practices dictate leaving the network information out of the zone configuration, and using the IP agents in Veritas Cluster Server to control the virtual IP. This topic is covered in more detail in the section "**Defining the Non-Global Zone**."

CONFIGURING A LOCAL ZONE TO WORK WITH CLUSTER FILE SYSTEM HIGH AVAILABILITY

♣ **Important note:** When configuring a local zone which will be under Veritas Cluster Server control, there is only one required deviation from the default build: the zone must be configured so that it will not boot when the system is started, as Veritas Cluster Server will be responsible for controlling the boot execution for the zone.

Installing Non-Global Zones

The available options for when initially configuring a Zone are too numerous to cover for scope of this document. However, for the sake of context and understanding the necessary steps, the basic zonecfg and zoneadm commands required are detailed below. Additionally, it is assumed that all LUNS have been presented to the corresponding nodes as well as formatted and initialized for use with Volume Manager. Please consult the Storage Foundation Administrators guide for details on preparing storage for use with VxVM.

https://sort.symantec.com/documents/doc_details/sfha/6.0/Solaris/ProductGuides/

Example Zone Configuration Overview

Throughout this document, the example of a two-node Veritas Cluster Server cluster will be used. The basic configuration is as follows:

Cluster Nodes: **node1** & **node2**
Local zones with VxFS/Cluster File System root: **dangerzone** and **redzone**
Shared Zone with Cluster File System Root: **twilightzone**
Local zones with ZFS root: **calzone** and **endzone**
VxFS Zone root file system: **/zones/"zonename"/base**
ZFS Zone root file system: **/zones/"zonename"/base**
Cluster File System/LOFS Application volumes mounted at: **/zones/"zonename"/data_mnt**

The Examples given in this document will cover the following 5 scenarios:

- VxFS Root for Local Zone with Cluster File System/Direct Mount for Application Data (App Failover)
- VxFS Root for Local Zone with Cluster File System/LOFS for Application Data (App Failover)
- Cluster File System Root for Shared Zone with Cluster File System/LOFS for Application Data (Zone/App Failover)

(Appendix A)

- ZFS Root for Local Zone with ZFS Direct Mount for Application Data (Zone and App Failover)
- ZFS Root for Local Zone with Cluster File System/Direct Mount for Application Data (Application Failover)

In order to support the scenarios above for application and Zone failover, the following considerations must be noted:

- The application service group (failover) must be dependent upon the Zone, root and data storage service group using online-local-firm
- When configuring Zones for application failover you must localize the ContainerInfo Service Group attributes (Name, Type and Enabled)
- The Mountpoint attribute for the Cluster File SystemMount data resource must also be localized
  - Node 1 → /zones/ZoneA//root/mydata
  - Node 2 → /zones/ZoneB/root/mydata
- Once the zone is brought online with Direct Mount file systems, this mount point will be visible only from within the Zone itself using "df" or by using the "mount" command from the global zone.

Defining The Non-Global-Zone

While it's beyond the scope of this document to cover details on configuring a local zone, some review of a zone's configuration is in order.

Local zone configurations are maintained in /etc/zones. For each local zone configured on a host, an entry exists in /etc/zones/index and appears as follows:

calzone:installed:/zones/calzone/base

The components of the entry are zone name, status and path to zone root, separated by colons. In the /etc/zones directory, each local zone's configuration is stored in a file in XML format as <zonename>.xml. Figure 4 contains the entire zone configuration for our sample zone.

```
<!DOCTYPE zone PUBLIC "-//Sun Microsystems Inc//DTD Zones//EN" "file:///usr/share/lib/xml/dtd/zonecfg.dtd.1">
<!--
   DO NOT EDIT THIS FILE.  Use zonecfg(1M) instead.
-->
<zone name="calzone" zonepath="/zones/dangerzone/base" autoboot="false"/>
<filesystem special="/zones/dangerzone/mydata"directory="/mydata" type="lofs"/> </zone>
```

Figure 5 – Local zone configuration file in XML format<?xml version="1.0" encoding="UTF-8"?>

There are several areas worth noting in the above configuration.

In order for Veritas Cluster Server to be effectively responsible for controlling the online and offline operations of the zone the auto-boot attribute must be set to false. The second area of note is the lack of any specific network information. Veritas Cluster Server will make the appropriate IP address(s) available to the zone after it's booted using the IP Agent. This is particularly useful when dealing with disaster recovery and the disaster recovery personalization features included with Veritas Cluster Server. Keeping the resident configuration of the zone as generic as possible will allow for increased portability as well as simpler migrations.

There are, however, circumstances where as having the IP address available as part of the zonecfg process will be required. One example would be the presence of NFS mounted file systems in the zone's /etc/vfstab. You can however choose (optionally) to implement the Veritas Cluster Server Mount agent to manage the NFS file systems rather than the zone boot sequence itself.

It is also worth noting that should you choose to configure an IP address as part of the zonecfg process, you will still want to use the IP Agent to monitor the status of the network address. The is due to the default behavior that a Solaris Zones will remain booted even in the event that it's IP address is taken offline.

♣ **Important Note:** When you omit network information during the Zone build process, you may encounter error messages when attempting to configure DNS during the network configuration steps. To mitigate this issue you can simply copy the /etc/resolv.conf and nsswitch.conf from the global zone to the local zone after the configuration is complete.

The final area of mention is the "zonepath" attribute. The purpose for creating the additional "base" directory is to allow for a consistent mount point path for any subsequent LOFS or Cluster File System Direct Mount file systems that may be necessary. This method is entirely optional and is done so purely for consistency purposes only.

Sample Zone Configuration

Creating the Local Root Zone Configuration (for use with LOFS data Mounts)

bash-3.2# **zonecfg -z zonename**
calzone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zonename > **create -b**
zonecfg:zonename > **set zonepath=/zones/zonename/base**
zonecfg:zonename > **set autoboot=false**
zonecfg:zonename> **add fs**
zonecfg:zonename:fs> **set dir=/data**
zonecfg:zonename:fs> **set special=/zones/zonename/data**
zonecfg:zonename:fs> **set type=lofs**
zonecfg:zonename:fs> **end**
zonecfg:zonename > **commit**
zonecfg:zonename > **verify**
zonecfg:zonename > **exit**

Creating the Local Root Zone Configuration (for use with Direct Mount File Systems)

bash-3.2# **zonecfg -z zonename**
calzone: No such zone configured
se 'create' to begin configuring a new zone.
zonecfg:zonename > **create -b**
zonecfg:zonename > **set zonepath=/zones/zonename/base**
zonecfg:zonename > **set autoboot=false**
zonecfg:zonename > **commit**
zonecfg:zonename > **verify**
zonecfg:zonename > **exit**

Note that these steps must be performed on each node in the cluster where the local zone is configured to run. You can optionally export the Zone configuration to a file and configure any subsequent nodes by copying the configuration file to remaining nodes and using the following commands:

Node1:
#> **zonecfg –z zonenmame export –f zonename.cfg**
#> scp zone.cfg node2:/tmp

Node2:
#> zonecfg –z zonename –f /tmp/zonename.cfg

SUMMARY

Controlling applications running in Solaris local zones using Veritas Cluster Server is a relatively straight-forward process. There are however a number of configuration options that can be chosen to achieve a variety of different results. Whether your goal is to reduce physical server footprint or simply increase utilization of the available host-side resources, Solaris zones are very effective means to achieve these milestones.

Combining however, Cluster File System High Availability and Solaris local zones provide a robust, flexible and scalable solution that addresses a variety of use cases. These include reducing the downtime associated with a number of dally operational tasks, providing increased levels of availability as well as allowing for enterprise wide visibility to all Solaris (IBM, VMware Red Hat and Microsoft included) virtual environments through the use of VERITAS Operations Manager. Furthermore, Cluster File System High Availability will dramatically reduce the recovery times associated with planned as well as unplanned outages.

No true cluster solution would be complete unless it addresses both Local High Availability as well as DR considerations. Veritas Cluster Server does precisely that by allowing for DR personalization for use with replication of the zone root. IT Enterprises can now confidently replicate their Solaris virtual environments without concern for the often tedious and manual process of preparing a Zone for use in a new Data center. For more information about this feature please refer to the *Veritas Cluster Server Bundled Agents Reference Guide* on Symantec™ Operation Readiness Tools.

Cluster File System High Availability is comprehensive solution which improves upon that native Solaris zone framework. This is done so by providing the same High Availability/Disaster Recovery feature set that users have been implementing for years with Veritas Cluster Server. Having adapted them in to the virtual context, this inevitably makes the adoption of containers a much easier process.

Option 1: VxFS Zone Root (Local) and Cluster File System for Application Data (Direct Mount)

In this configuration, each zone root will reside on local storage while the application data will reside on Cluster File System. The unit of failover will be the application and as such two separate service groups will be required. One parallel SG for the zone and Cluster File System file system and one failover SG for the applications and virtual IP's.

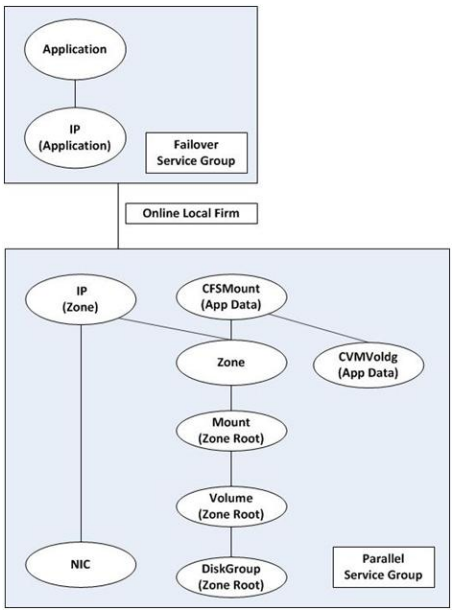


Figure 6 - Service Group Dependency for Direct Mount Cluster File System

Step 1: Configure Zone Root and Application Disk Groups, Volumes and File Systems:

It is the assumption of this document that all LUN's have been formatted/labeled and initialized for use with Volume Manager prior to executing the following steps.

A. Create Disk Groups on each host for the Zone Root (Node1 & Node2)

From Node1

#> vxdg init dangerzone_zroot_dg dangerzone_zroot_dg01=hitachi_usp-vm0_083f

From Node2

#> vxdg init redzone_zroot_dg redzone_zroot_dg01=hitachi_usp-vm0_0840

B. Create Volumes and File Systems for each Zone Root (Node1 & Node2)

Node 1:

#> vxassist –g dangerzone_zroot_dg make dangerzone_zroot_vol 5g dangerzone_zroot_dg01

#> mkfs –F vxfs /dev/vx/rdsk/dangerzone_zroot_dg/dangerzone_zroot_vol

#> mkdir –p /zones/dangerzone/base

#> mount –f vxfs /dev/vx/dsk/dangerzone_zroot_dg/dangerzone_zroot_vol /zones/dangerzone/base #> chmod 700 /zones/dangerzone/base

Node 2:

#> vxassist –g redzone_zroot_dg make redzone_zroot_vol 5g redzone_zroot_dg01 #> mkfs –F vxfs /dev/vx/rdsk/endzone_zroot_dg/ endzone_zroot_vol

#> mkdir –p /zones/endzone/base #> mount –f vxfs /dev/vx/dsk/endzone_zroot_dg/endzonezone_zroot_vol /zones/endzone/base #> chmod 700 /zones/endzone/base

C. Create Service Group for Zone and Storage Resources

From Either Node:

Create Service Group for Zone and Storage Resources

#> haconf –makerw #> hagrp -add local_vxfs_zone_SG

#> hagrp -modify local_vxfs_zone_SG SystemList node1 0 node2 1

#> hagrp -modify local_vxfs_zone_SG Parallel 1

Add DiskGroup Resource

#> hares -add zoneroot_DG DiskGroup local_vxfs_zone_SG

#> hares -modify zoneroot_DG Critical 0

#> hares -modify zoneroot_DG StartVolumes 0

#> hares -modify zoneroot_DG StopVolumes 1

#> hares -local zoneroot_DG DiskGroup

#> hares -modify zoneroot_DG DiskGroup dangerzone_zroot_dg -sys node1

#> hares -modify zoneroot_DG DiskGroup redzone_zroot_dg -sys node2

#> hares -modify zoneroot_DG Enabled 1

Add Volume Resource

#> hares -add zoneroot_DG DiskGroup local_vxfs_zone_SG

#> hares -modify zoneroot_VOL Critical 0

#> hares -local zoneroot_VOL Volume

#> hares -modify zoneroot_VOL Volume dangerzone_zroot_vol -sys node1

#> hares -modify zoneroot_VOL Volume endzone_zroot_vol -sys node2

#> hares -local zoneroot_VOL DiskGroup

#> hares -modify zoneroot_VOL DiskGroup dangerzone_zroot_dg -sys node1

#> hares -modify zoneroot_VOL DiskGroup endzone_zroot_dg -sys node2

#> hares -modify zoneroot_VOL Enabled 1

#> hares –link zoneroot_VOL zoneroot_DG

Add Mount Resource

#> hares -add zoneroot_MNT Mount local_vxfs_zone_SG

#> hares -modify zoneroot_MNT Critical 0

#> hares -local zoneroot_MNT MountPoint

#> hares -modify zoneroot_MNT MountPoint /zones/dangerzone/base -sys node1

#> hares -modify zoneroot_MNT MountPoint /zones/redzone/base -sys node2

#> hares -local zoneroot_MNT BlockDevice

#> hares -modify zoneroot_MNT BlockDevice /dev/vx/dsk/dangerzone_zroot_dg/dangerzone_zroot_vol -sys node1

#> hares -modify zoneroot_MNT BlockDevice /dev/vx/dsk/redzone_zroot_dg/redzone_zroot_vol -sys node2

#> hares -modify zoneroot_MNT FSType vxfs

#> hares -modify zoneroot_MNT FsckOpt %-n

#> hares -modify zoneroot_MNT Enabled 1

#> hares –link zoneroot_MNT zoneroot_VOL

D. Create CVM Disk Groups, Volumes and File Systems for Application Data

From either Cluster File System node:

#> vxdg -s init mydata_dg mydata_dg01=hitachi_usp-vm0_083e

#> vxassist –g mydata_dg make mydata_vol 1g mydata_dg01

#> mkfs –F vxfs /dev/vx/rdsk/mydata_dg/mydata_vol

From Node1:

#> mkdir /zones/dangerzone/mydata

#> mount –f vxfs –o cluster,suid,rw /dev/vx/dsk/mydata_dg/mydata_vol /zones/dangerzone/mydata

From Node2:

#> mkdir /zones/redzone/mydata

#> mount –f vxfs –o cluster,suid,rw /dev/vx/dsk/mydata_dg/mydata_vol /zones/redzone/mydata

From Either Node:

♣ **Important Note:** For this scenario the commands for adding a cluster mount to your Veritas Cluster Server configuration will require that you first choose the same entry for the MountPoint argument for the **cfsmntadm** command followed by modifying the Mount Point attribute using the **hares –modify** flag so that it is localized to each Host.

#> cfsmntadm add mydata_dg mydata_vol /zones/mydata local_vxfs_zone_SG node1=suid,rw node2=suid,rw →This will add a CVMVoldg and CFSMount resource to the local_vxfs_zone_SG service group

#> hares -local cfsmount1 MountPoint → The naming convention of cfsmount# is the default naming scheme for adding Cluster File System Mount resources. You may choose to modify the resource name offline by editing the main.cf or via the copy/paste function on the Veritas Cluster Server Java GUI.

#> hares -modify cfsmount1 MountPoint /zones/dangerzone/mydata -sys node1
#> hares -modify cfsmount1 MountPoint /zones/redzone/mydata -sys node2
#> hares -modify cfsmount1 NodeList node1 node2
#> hares -modify cfsmount1 Primary node1
#> haconf –dump –makero
#> hagrp –online local_vxfs_zone_SG –any

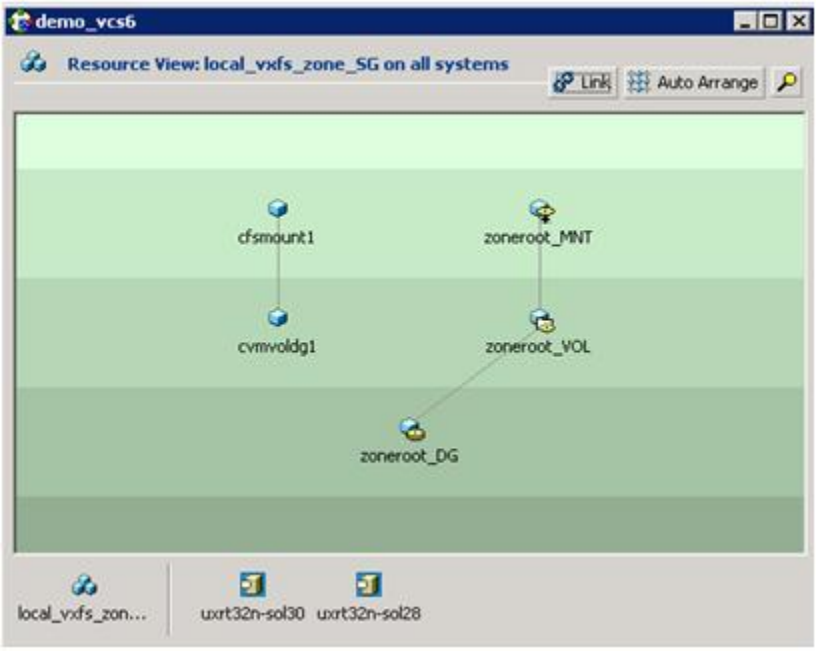


Figure 7: Resource Dependency View

Step 2: Configure & Install each Zone

A. Define each individual Zone on its corresponding Physical Node:

From Node1:

#> zonecfg -z dangerzone
create -b
set zonepath=/zones/dangerzone/base
set autoboot=false
commit
verify
exit

#> zoneadm –z dangerzone install
#> zoneadm –z dangerzone boot
#> zlogin –C dangerzone

Follow Prompts for configuring zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

#> cp /etc/resolv.conf /zones/dangerzone/base/root/etc
#> cp /etc/nsswtich* /zones/dangerzone/base/root/etc/

This procedure will only work consistently for shared-IP zones.

From Node2:

#> zonecfg -z redzone
create -b
set zonepath=/zones/redzone/base
set autoboot=false
commit
verify
exit

#> zoneadm –z redzone install
#> zoneadm –z redzone boot
#> zlogin –C redzone

Follow Prompts for configuring the zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

#> cp /etc/resolv.conf /zones/redzone/base/root/etc
#> cp /etc/nsswtich* /zones/redzone/base/root/etc/

This procedure will only work consistently for shared-IP zones.

Step 3: Configure Service Groups to support Zones and for Application Failover

For Veritas Cluster Server to effectively monitor applications that reside inside of a local zone, Veritas Cluster Server must be configured to allow communications to occur from the global zone to the local zone and vice versa. This is established using a combination of the "halogin" and "hazonesetup" commands. Please note that any resources configured with the Critical flag set to "0" are done so for initial deployment purposes only.

HAZONESETUP: This utility establishes the following configuration settings:

- Creates a new or modifies an existing Service Group to support the ContainerInfo Attributes
- Defines whether the SG is of the parallel or failover variety.
- Creates a new resource of Type Zone in the aforementioned SG.
- Optionally creates new Individual users for Veritas Cluster Server authentication from the non-global zone (Otherwise a predefined user can be chosen).
- Established Authentication between the non-global zone and Global Zone (halogin)

From the Global Zone

#> hazonesetup [-t] sg_name -r res_name -z zone_name [-u] user_name -p password [-a] [-l] -s sys1,sys2

Where the Values are:

-t	Updates the password for the Veritas Cluster Server zone user.
-g *sg_name*	Name of the zone service group to be created in Veritas Cluster Server configuration
-r *res_name*	Name of the zone resource to be created in Veritas Cluster Server configuration
-z *zone_name*	Name of the zone that is configured on the system.
-u *user_name*	Name of the Veritas Cluster Server user used for password less communication between the local zone and the global zone. If no username is specified the default username is used.
-p *password*	Password for the Veritas Cluster Server user used for password less communication
-a	Populate AutoStartList for the group.
-l	Configure a parallel service group. If you do not specify the -l option a failover service group is created by default.
-s *systems*	A comma separated list of systems where the zone service group need to be configured, for example: sys1,sys2,sys3.

A. Configure Individual Zone IP Addresses

Node1 & Node2

#> **ifconfig** *interface:#* **plumb**

#> **ifconfig** *interface:1 Address* **netmask** *netmask* **zone** *zonename* **up**

Example: (node1)

#> **ifconfig bge0:1 plumb**

#> **ifconfig bge0:1 10.10.10.1 netmask 255.255.255.0 zone dangerzone up**

Example: (node2)

#> **ifconfig bge0:1 plumb**

#> **ifconfig bge0:1 10.10.10.2 netmask 255.255.255.0 zone redzone up**

B. Run the hazonesetup script on each cluster node.

Node1:

#> hazonesetup -t -g local_vxfs_zone_SG -r localzone_ZONE -z dangerzone -u z_Veritas Cluster Server_dangerzone -p password -l -s node1

Node2:

#> hazonesetup -t -g local_vxfs_zone_SG -r localzone_ZONE -z redzone -u z_Veritas Cluster Server_redzone -p password -l -s node2

C. Add Zone IP and NIC Resource to Service Group.

Add NIC Resource

From Either Node:

#> hares -add zone_NIC NIC local_vxfs_zone_SG

#> hares -modify zone_NIC Device bge0

#> hares -modify zone_NIC Critical 0

#> hares -modify zone_NIC Enabled 1

Add IP Resource

From Either Node:

#> hares -add zone_IP IP local_vxfs_zone_SG

#> hares -modify IP Critical 0

#> hares -modify IP Device bge0

#> hares -local IP Address

#> hares -modify IP Address 10.10.10.1 -sys node1

#> hares -modify IP Address 10.10.10.2 -sys node2

#> hares -modify IP NetMask 255.255.240.0

#> hares -modify IP Enabled 1

Configure Resource Dependencies

From Either Node:

#> hares -link cfsmount1 localzone_ZONE

#> hares -link localzone_ZONE zoneroot_MNT

#> hares -link zone_IP localzone_ZONE

#> hares –link zone_IP zone_NIC

#> haconf –dump -makero

Figure 8: Completed Resource Dependency View

Once the Zone service group is completed, you will want establish a separate failover service group to support your application(s). The dependency between the Application/Parent service group and the Zone/Child service group should be Online Local Firm. Please follow the Veritas Cluster Server solutions guides for specifics on how to configure the support applications for failover.

#> hagrp –link my_application_SG local_vxfs_zone_SG online local firm

https://sort.symantec.com/documents/doc_details/sfha/6.0/Solaris/ProductGuides/

Option 2: VxFS Zone Root (Local) and Cluster File System for Application Data (LOFS)

In this configuration, each zone root will reside on local storage while the application data will reside on Cluster File System. The unit of failover will be the application and as such two separate service groups will be required. One parallel SG for the zone and Cluster File System file system and one SG for the applications and virtual IP's.

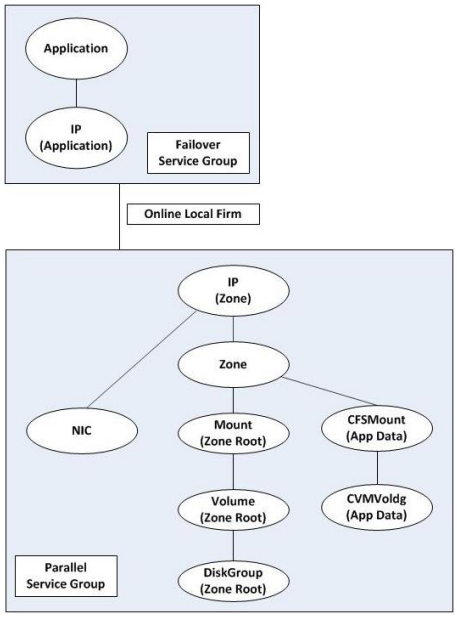


Figure 9 - Service Group Dependency for LOFS Mount Cluster File System

Step 1: Configure Zone Root and Application Disk Groups, Volumes and File Systems:

A. Create Disk Groups on each host for the Zone Root (Node1 & Node2)

From Node1

#> vxdg init dangerzone_zroot_dg dangerzone_zroot_dg01=hitachi_usp-vm0_083f

From Node2

#> vxdg init redzone_zroot_dg redzone_zroot_dg01=hitachi_usp-vm0_0840

B. Create Volumes and File Systems for each Zone Root (Node1 & Node2)

Node 1:

#> vxassist –g dangerzone_zroot_dg make dangerzone_zroot_vol 5g dangerzone_zroot_dg01

#> mkfs –F vxfs /dev/vx/rdsk/dangerzone_zroot_dg/dangerzone_zroot_vol

#> mkdir –p /zones/dangerzone/base

#> mount –f vxfs /dev/vx/dsk/dangerzone_zroot_dg/dangerzone_zroot_vol /zones/dangerzone/base

#> chmod 700 /zones/dangerzone/base

Node 2:

#> vxassist –g redzone_zroot_dg make redzone_zroot_vol 5g redzone_zroot_dg01

#> mkfs –F vxfs /dev/vx/rdsk/endzone_zroot_dg/endzone_zroot_vol

#> mkdir –p /zones/endzone/base

#> mount –f vxfs /dev/vx/dsk/endzone_zroot_dg/endzonezone_zroot_vol /zones/endzone/base

#> chmod 700 /zones/endzone/base

C. Create Service Group and Resources for Zone Root

From Either Node

Create Service Group for Zone and Storage Resources

#> haconf –makerw

#> hagrp -add local_vxfs_zone_SG

#> hagrp -modify local_vxfs_zone_SG SystemList node1 0 node2 1

#> hagrp -modify local_vxfs_zone_SG Parallel 1

Add DIskGroup Resource

#> hares -add zoneroot_DG DiskGroup local_vxfs_zone_SG

#> hares -modify zoneroot_DG Critical 0

#> hares -modify zoneroot_DG StartVolumes 0

#> hares -modify zoneroot_DG StopVolumes 1

#> hares -local zoneroot_DG DiskGroup

#> hares -modify zoneroot_DG DiskGroup dangerzone_zroot_dg -sys node1

#> hares -modify zoneroot_DG DiskGroup redzone_zroot_dg -sys node2

#> hares -modify zoneroot_DG Enabled 1

Add Volume Resource

#> hares -add zoneroot_DG DiskGroup local_vxfs_zone_SG

#> hares -modify zoneroot_VOL Critical 0

#> hares -local zoneroot_VOL Volume

#> hares -modify zoneroot_VOL Volume dangerzone_zroot_vol -sys node1

#> hares -modify zoneroot_VOL Volume endzone_zroot_vol -sys node2

#> hares -local zoneroot_VOL DiskGroup

#> hares -modify zoneroot_VOL DiskGroup dangerzone_zroot_dg -sys node1

#> hares -modify zoneroot_VOL DiskGroup endzone_zroot_dg -sys node2

#> hares -modify zoneroot_VOL Enabled 1

#> hares –link zoneroot_VOL zoneroot_DG

Add Mount Resource

#> hares -add zoneroot_MNT Mount local_vxfs_zone_SG

#> hares -modify zoneroot_MNT Critical 0

#> hares -local zoneroot_MNT MountPoint

#> hares -modify zoneroot_MNT MountPoint /zones/dangerzone/base -sys node1

#> hares -modify zoneroot_MNT MountPoint /zones/redzone/base -sys node2

#> hares -local zoneroot_MNT BlockDevice

#> hares -modify zoneroot_MNT BlockDevice
/dev/vx/dsk/dangerzone_zroot_dg/dangerzone_zroot_vol -sys node1
#> hares -modify zoneroot_MNT BlockDevice
/dev/vx/dsk/redzone_zroot_dg/redzone_zroot_vol -sys node2
#> hares -modify zoneroot_MNT FSType vxfs
#> hares -modify zoneroot_MNT FsckOpt %-n
#> hares -modify zoneroot_MNT Enabled 1
#> hares –link zoneroot_MNT zoneroot_VOL

Bring Service Group Online

#> hagrp –online local_vxfs_zone_SG -any

D. Create CVM Disk Groups, Volumes and File Systems for Application Data

From CVM Master node: vxdctl –c mode

#> **vxdg -s init mydata_dg mydata_dg01=hitachi_usp-vm0_083e**
#> **vxassist –g mydata_dg make mydata_vol 1g mydata_dg01**
#> **mkfs –F vxfs /dev/vx/rdsk/mydata_dg/mydata_vol**

From Node1:
#> mkdir /zones/dangerzone/mydata

From Node2:
#> mkdir /zones/redzone/mydata

From CVM Master:

♣ **Important Note:** For this scenario the commands for adding a cluster mount to your Veritas Cluster Server configuration will require that you first choose the same entry for the MountPoint argument for the **cfsmntadm** command followed by modifying the Mount Point Attribute using the **hares –modify** flag so that it is localized to each host.

#> cfsmntadm add mydata_dg mydata_vol /zones/mydata local_vxfs_zone_SG node1=suid,rw node2=suid,rw → This will add a CVMVoldg and CFSMount Resource to the local_vxfs_zone_SG service group

#> hares -local cfsmount1 MountPoint→ The naming convention of cfsmount# is the default naming scheme for adding CFSMount resources. You may choose to modify the resource name offline by editing the main.cf or via the copy/paste function on the Veritas Cluster Server Java GUI.

#> **hares -modify cfsmount1 MountPoint /zones/dangerzone/mydata -sys node1**
#> **hares -modify cfsmount1 MountPoint /zones/redzone/mydata -sys node2**
#> **hares -modify cfsmount1 NodeList node1 node2**
#> **hares -modify cfsmount1 Primary node1**
#> **haconf –dump –makero**
#> **hagrp –online local_vxfs_zone_SG –any**

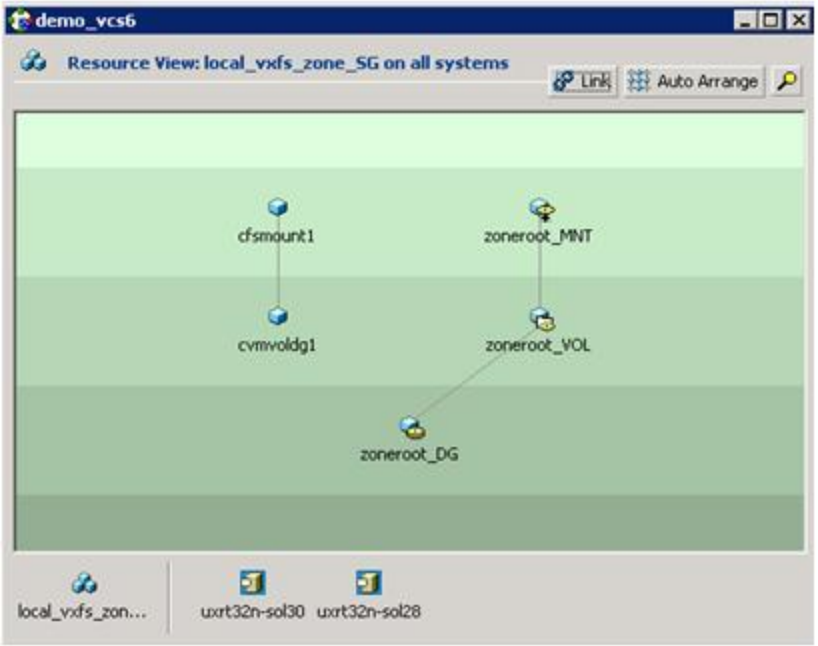


Figure 10 Storage Resource Dependency View

Step 2: Define and configure each individual Zone on its corresponding Physical Node:

From Node1:

#> zonecfg -z dangerzone
create -b
set zonepath=/zones/dangerzone/base
set autoboot=false
add fs
set dir=/mydata
set special=/zones/dangerozne/mydata
set type=lofs
end
commit
verify
exit

#> zoneadm –z dangerzone install
#> zoneadm –z dangerzone boot
#> zlogin –C dangerzone

Follow Prompts for configuring zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

#> cp /etc/resolv.conf /zones/dangerzone/base/root/etc/resolv.conf

#> cp /etc/nsswtich* /zones/dangerzone/base/root/etc/

This procedure will only work consistently for shared-IP zones.

From Node2:

#> zonecfg -z redzone
create -b
set zonepath=/zones/redzone/base
set autoboot=false
add fs
set dir=/m
data
set special=/zones/redzone/mydata
set type=lofs
end
commit
verify
exit

#> zoneadm –z redzone install
#> zoneadm –z redzone boot
#> zlogin –C redzone

Follow Prompts for configuring the zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

#> cp /etc/resolv.conf /zones/redzone/base/root/etc

#> cp /etc/nsswtich* /zones/redzone/base/root/etc/

This procedure will only work consistently for shared-IP zones.

Step 3: Configure Service Groups to support Zones and for Application Failover

For Veritas Cluster Server to effectively monitor applications that reside inside of a local zone, Veritas Cluster Server must be configured to allow communications to occur from the global zone to the local zone and vice versa. This is established using a combination of the "halogin" and "hazonesetup" commands. Please note that any resources configured with the Critical flag set to "0" are done so for initial deployment purposes only.

HAZONESETUP: This utility establishes the following configuration settings:

- Creates a new or modifies an existing Service Group to support the ContainerInfo Attributes
- Defines whether the SG is of the parallel or failover variety.
- Creates a new resource of Type Zone in the aforementioned SG.

- Optionally creates new Individual users for Veritas Cluster Server authentication from the non-global zone (Otherwise a predefined user can be chosen).
- Established Authentication between the non-global zone and Global Zone (halogin)

From the Global Zone

#> hazonesetup [-t] sg_name -r res_name -z zone_name [-u] user_name -p password [-a] [-l] -s sys1,sys2

Where the Values are:

-t	Updates the password for the VCS zone user.
-g *sg_name*	Name of the zone service group to be created in VCS configuration
-r *res_name*	Name of the zone resource to be created in VCS configuration
-z *zone_name*	Name of the zone that is configured on the system.
-u *user_name*	Name of the VCS user used for password less communication between the local zone and the global zone. If no username is specified the default username is used.
-p *password*	Password for the VCS user used for password less communication
-a	Populate AutoStartList for the group.
-l	Configure a parallel service group. If you do not specify the -l option a failover service group is created by default.
-s *systems*	A comma separated list of systems where the zone service group need to be configured, for example: sys1,sys2,sys3.

D. Configure Individual Zone IP Addresses

Node1 & Node2

#> ifconfig interface:# plumb

#> ifconfig interface:1 Address netmask netmask zone zonename up

Example: (node1)

#> ifconfig bge0:1 plumb

#> ifconfig bge0:1 10.10.10.1 netmask 255.255.255.0 zone dangerzone up

Example: (node2)

#> ifconfig bge0:1 plumb

#> ifconfig bge0:1 10.10.10.2 netmask 255.255.255.0 zone redzone up

E. Run the hazonesetup script on each cluster node.

Node1:

#> hazonesetup -t -g local_vxfs_zone_SG -r localzone_ZONE -z dangerzone -u z_vcs_dangerzone -p password -l -s node1

Node2:

#> hazonesetup -t -g local_vxfs_zone_SG -r localzone_ZONE -z redzone -u z_vcs_redzone -p password -l -s node2

F. Add Zone IP and NIC Resource to Service Group.

Add NIC Resource

From Either Node:

#> hares -add zone_NIC NIC local_vxfs_zone_SG

#> hares -modify zone_NIC Device bge0

#> hares -modify zone_NIC Critical 0

#> hares -modify zone_NIC Enabled 1

Add IP Resource

From Either Node:

#> hares -add zone_IP IP local_vxfs_zone_SG

#> hares -modify IP Critical 0

#> hares -modify IP Device bge0

#> hares -local IP Address

#> hares -modify IP Address 10.10.10.1 -sys node1

#> hares -modify IP Address 10.10.10.2 -sys node2

#> hares -modify IP NetMask 255.255.240.0

#> hares -modify IP Enabled 1

Configure Resource Dependencies

From Either Node:

#> hares -link localzone_ZONE cfsmount1

#> hares -link localzone_ZONE zoneroot_MNT

#> hares -link zone_IP localzone_ZONE

#> hares –link zone_IP zone_NIC

#> haconf –dump -makero

Figure 11 Completed Resource Dependency View

Once the Zone service group is completed, you will want establish a separate failover service group to support your application(s). The dependency between the Application/Parent service group and the Zone/Child service group should be Online Local Firm. Please follow the VCS solutions guides for specifics on how to configure the supported applications for failover.

#> hagrp –link my_application_SG local_vxfs_zone_SG online local firm

https://sort.symantec.com/documents/doc_details/sfha/6.0/Solaris/ProductGuides/

Option 3: Cluster File System Zone Root (Shared) and Cluster File System for Application Data (LOFS)

In this scenario, a single zone will be configured with its root file system on shared storage. The application data will also reside on Cluster File System which can be presented as LOFS or direct mount. The unit of failover will be the entire zone and as such the Zone and Application resources will be configured in a same service group. The Zone Root and application storage resources will be configured as a parallel service group.

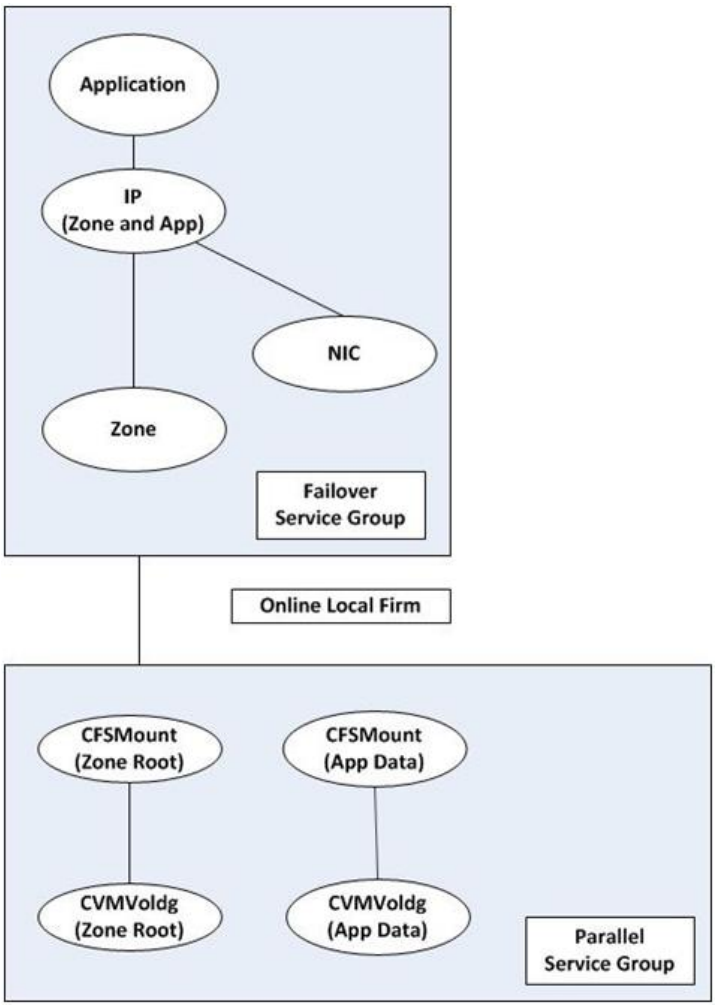


Figure 12 - Service Group Dependency for LOFS/CFS

Step 1: Configure Zone Root and Application Disk Groups, Volumes and File Systems:

A. Create Shared Disk Group, Volume and File System for the Zone Root

From Master Node1

#> vxdg –s init twilightzone_zroot_dg dangerzone_zroot_dg01=hitachi_usp-vm0_083f

Example:

Master Node:

#> vxassist –g twilightzone_zroot_dg make twilightzone _zroot_vol 5g twilightzone _zroot_dg01

#> mkfs –F vxfs /dev/vx/rdsk/twilightzone _zroot_dg/twilightzone _zroot_vol

#> mkdir –p /zones/twilightzone/base#> mount –f vxfs –o cluster,suid,rw /dev/vx/dsk/twilightzone _zroot_dg/twilightzone _zroot_vol /zones/twilightzone/base

#> chmod 700 /zones/twilightzone/base

Node 2:

#> mkdir –p /zones/twilightzone/base

#> mount –f vxfs –o cluster,suid,rw /dev/vx/dsk/twilightzone _zroot_dg/twilightzone _zroot_vol/zones/ twilightzone/base#

> chmod 700 /zones/twilightzone/base

#> haconf –makerw

#> cfsmntadm add twilightzone_zroot_dg twilightzone_zroot_vol /zones/twilightzone/base shared_vxfs_zone_SG node1=suid,rw node2=suid,rw → This will create the parallel service group "local_vxfs_zone_SG" with a CVMVoldg and CFSMount Resource

B. Create Shared Disk Group and CFS File Systems for Application Data

Example: From Master Node

#> vxdg -s init mydata_dg mydata_dg01=hitachi_usp-vm0_0842

#> vxassist –g mydata_dg make mydata_vol 1g mydata_dg01

#> mkfs –F vxfs /dev/vx/rdsk/mydata_dg/mydata_vol

From Node1:

#> mkdir /zones/twilightzone/mydata

#> mount –f vxfs –o cluster,suid,rw /dev/vx/dsk/mydata_dg/mydata_vol /zones/twilightzone/mydata

From Node2:#

> mkdir /zones/twilightzone/mydata

#> mount –f vxfs –o cluster,suid,rw /dev/vx/dsk/mydata_dg/mydata_vol /zones/twilightzone/mydata

From Either Node:

#> cfsmntadm add mydata_dg mydata_vol /zones/twilightzone/mydata shared_vxfs_zone_SG node1=suid,rw node2=suid,rw → This will create the parallel service group "local_vxfs_zone_SG" with a CVMVoldg and CFSMount Resource

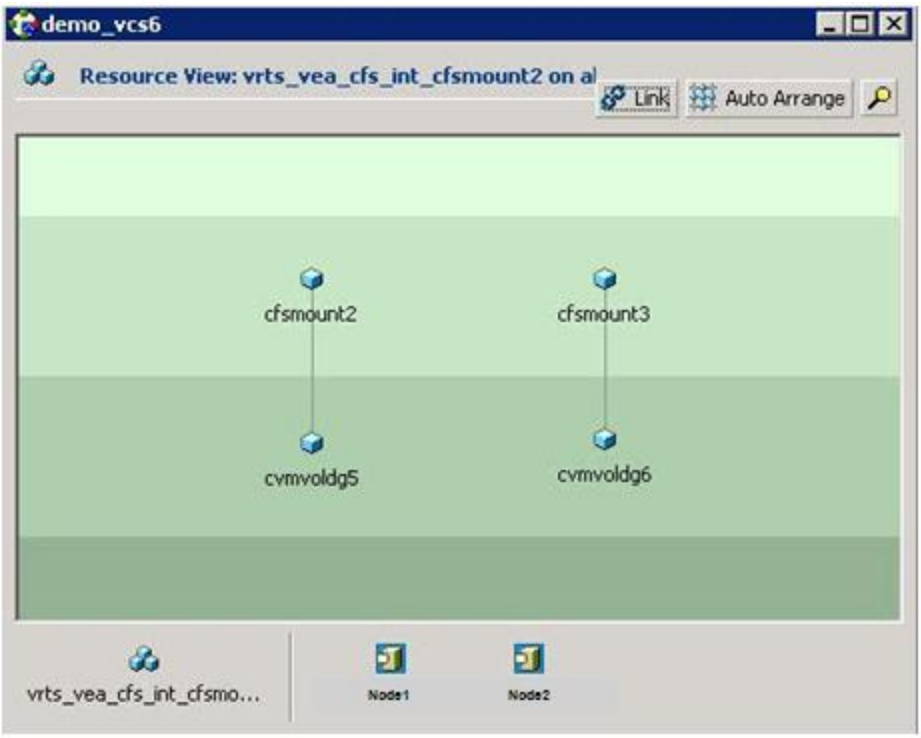


Figure 13 - Application Storage and Zone Root Resources

Step 2: Configure & Install each Zone and Incorporate CFS/LOFS File Systems

C. Define the Zone on each corresponding Physical Node:

Example:

From Node1:

#> zonecfg -z twilightzone
create -b
set zonepath=/zones/twilightzone/base
set autoboot=false
add fs
set dir=/mydata
set special=/zones/twilightzone/mydata
set type=lofs
end
commit
verify
exit

#> zoneadm –z twilightzone install
#> zoneadm –z twilightzone boot
#> zlogin –C twilightzone

Follow Prompts for configuring zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

#> cp /etc/resolv.conf /zones/dangerzone/base/root/etc
#> cp /etc/nsswtich* /zones/dangerzone/base/root/etc

This procedure will only work consistently for shared-IP zones.

D. Export Zone Configuration to Node2

From Node1:
#> zonecfg –z twilightzone export –f /tmp/twilightzone.cfg
#> scp /tmp/twilightzone node2:/tmp

From Node2:
#> zonecfg –z twlightzone –f /tmp/twilightzone.cfg

Step 3: Configure Service Groups to support Zones and for Application Failover

For Veritas Cluster Server to effectively monitor applications that reside inside of a local zone, Veritas Cluster Server must be configured to allow communications to occur from the global zone to the local zone and vice versa. This is established using a combination of the "halogin" and "hazonesetup" commands. Please note that any resources configured with the Critical flag set to "0" are done so for initial deployment purposes only.

HAZONESETUP: This utility establishes the following configuration settings:

- Creates a new or modifies an existing Service Group to support the ContainerInfo Attributes
- Defines whether the SG is of the parallel or failover variety.
- Creates a new resource of Type Zone in the aforementioned SG.
- Optionally creates new Individual users for Veritas Cluster Server authentication from the non-global zone (Otherwise a predefined user can be chosen). Established Authentication between the non-global zone and Global Zone (halogin)

From the Global Zone
#> hazonesetup [-t] sg_name -r res_name -z zone_name [-u] user_name -p password [-a] [-l] -s sys1,sys2

Where the Values are:

-t	Updates the password for the VCS zone user.
-g *sg_name*	Name of the zone service group to be created in VCS configuration
-r *res_name*	Name of the zone resource to be created in VCS configuration
-z *zone_name*	Name of the zone that is configured on the system.
-u *user_name*	Name of the VCS user used for password less communication between the local zone and the global zone. If no username is specified the default username is used.
-p *password*	Password for the VCS user used for password less communication
-a	Populate AutoStartList for the group.
-l	Configure a parallel service group. If you do not specify the -l option a failover service group is created by default.
-s *systems*	A comma separated list of systems where the zone service group need to be configured, for example: sys1,sys2,sys3.

E. Configure Individual Zone IP Addresses

Node1 & Node2

#> ifconfig interface:# plumb

#> ifconfig interface:1 Address netmask netmask zone zonename up

Example: (node1)

#> ifconfig bge0:1 plumb

#> ifconfig bge0:1 10.10.10.1 netmask 255.255.255.0 zone dangerzone up

Example: (node2)

#> ifconfig bge0:1 plumb

#> ifconfig bge0:1 10.10.10.2 netmask 255.255.255.0 zone redzone up

F. Run the hazonesetup script on each cluster node.

Node1:

#> hazonesetup -t -g local_vxfs_zone_SG -r localzone_ZONE -z dangerzone -u z_vcs_dangerzone -p password -l -s node1

Node2:

#> hazonesetup -t -g local_vxfs_zone_SG -r localzone_ZONE -z redzone -u z_vcs_redzone -p password -l -s node2

G. Add Zone IP and NIC Resource to Service Group.

Add NIC Resource

From Either Node:

#> hares -add zone_NIC NIC local_vxfs_zone_SG

#> hares -modify zone_NIC Device bge0

#> hares -modify zone_NIC Critical 0

#> hares -modify zone_NIC Enabled 1

Add IP Resource

From Either Node:

#> hares -add zone_IP IP local_vxfs_zone_SG

#> hares -modify IP Critical 0

#> hares -modify IP Device bge0

#> hares -local IP Address

#> hares -modify IP Address 10.10.10.1 -sys node1

#> hares -modify IP Address 10.10.10.2 -sys node2

#> hares -modify IP NetMask 255.255.240.0

#> hares -modify IP Enabled 1

Configure Resource Dependencies

From Either Node:

#> hares -link localzone_ZONE cfsmount1

#> hares -link localzone_ZONE zoneroot_MNT

#> hares -link zone_IP localzone_ZONE

#> hares –link zone_IP zone_NIC

#> haconf –dump -makero

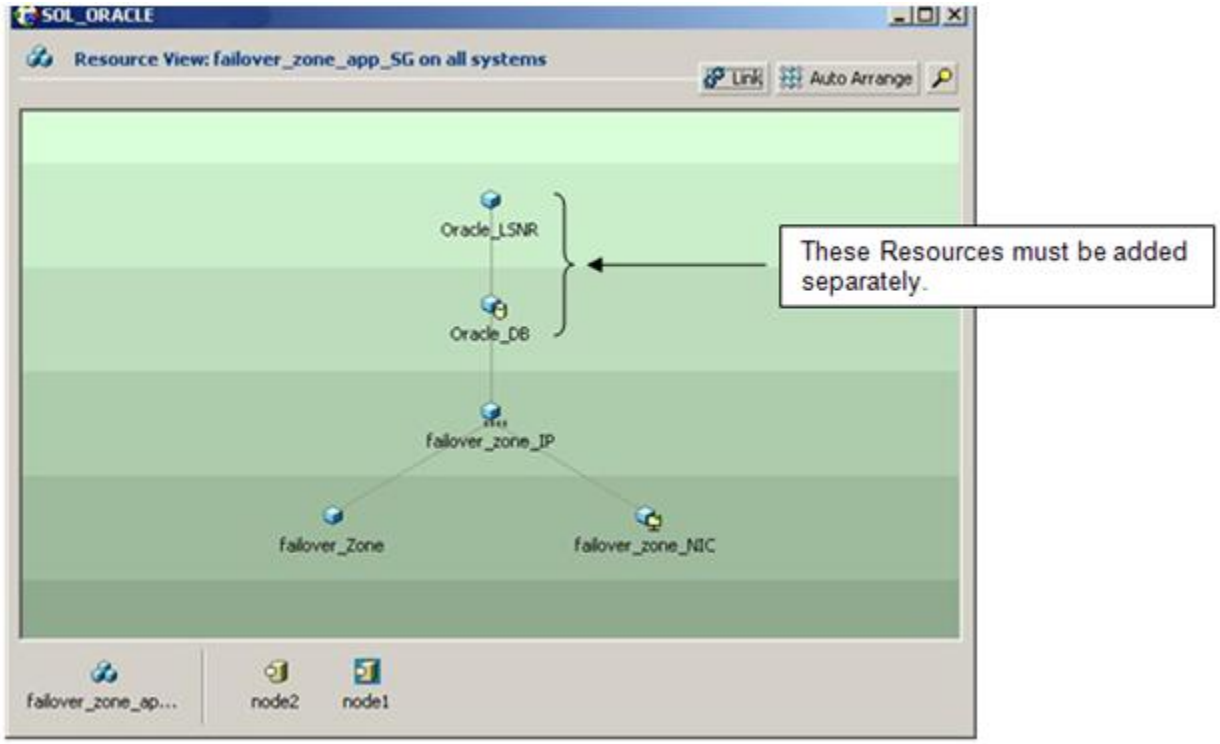


Figure 14 - Example Failover Application Service Group (Application Resources Shown)

Once the Zone service group is completed, you will want establish a separate failover service group to support your application(s). The dependency between the Application/Parent service group and the Zone/Child service group should be Online Local Firm. Please follow the VCS solutions guides for specifics on how to configure the supported applications for failover.

#> hagrp –link my_application_SG local_vxfs_zone_SG online local firm

https://sort.symantec.com/documents/doc_details/sfha/6.0/Solaris/ProductGuides/

Appendix A: Veritas Cluster Server, Local Zones and Native Volume Management

In this section the following topics will be covered:

- Overview of the Veritas Cluster System support for ZFS
- Configuring ZFS pools and File Systems
- Incorporating ZFS resources into failover and Parallel service groups
- Sample configurations
- Configuring ZFS and Cluster File System coexistence

Please note that although minimal CLI syntax will be provided to show the best practice for integrating ZFS into Veritas Cluster Server, this section will not cover in any great detail ZFS operations or administration. Please consult Oracle documentation for and additional ZFS related inquiries.

Veritas Cluster Server Support for ZFS

Bundled with Veritas Cluster Server are two agents that support native ZFS components, the ZPool and Mount agents. The ZPool agent was introduced to provide control for the import and export operations associated with ZFS storage pools. To support ZFS file systems, the Mount agent has been modified to include "ZFS" as an available file system type.

Configuring ZFS components for use with Veritas Cluster Server

♣ **Important note:** ZFS by default provides an automatic import service based on the following Solaris 10 SMF entry:

svc:/network/shares/group:zfs

What this service provides is the ability for Solaris 10 to, upon creation of a Zpool or reboot, import the storage pool and subsequently automount the ZFS file systems without using the /etc/vfstab file. As a best practice when using Veritas Cluster Server to manage ZPools (or VxVM volumes for that matter) as part of Service group, you will want to configure the "mount point" attribute for each pool and file system to "legacy" so as to allow Veritas Cluster Server to manage the import/export and mount operations exclusively. To identify what the mountpoint attribute for a particular ZPool or ZFS file system is by running the following command:

#> zfs get mountpoint "poolname" or "poolname/file system name"

The necessary syntax to configure this attribute is provided in the following sections.

Example 1: Local Zone Root with Shared Data File Systems (Direct Mount ZFS)

In this example, ZFS will be used to configure both the Zone root and application data file systems. Although not required, it is still a best practice to place local zone (not just shared zone) root file systems on SAN attached storage for increased flexibility. With the release of standalone DMP, those users wishing to place zone root file systems on ZFS can do so by creating the underlying ZPool using Enclosure Based Naming (EBN). The following setting will need to be applied prior to configuring any of the ZFS components.

#> vxdmpadm settune dmp_native_support=on

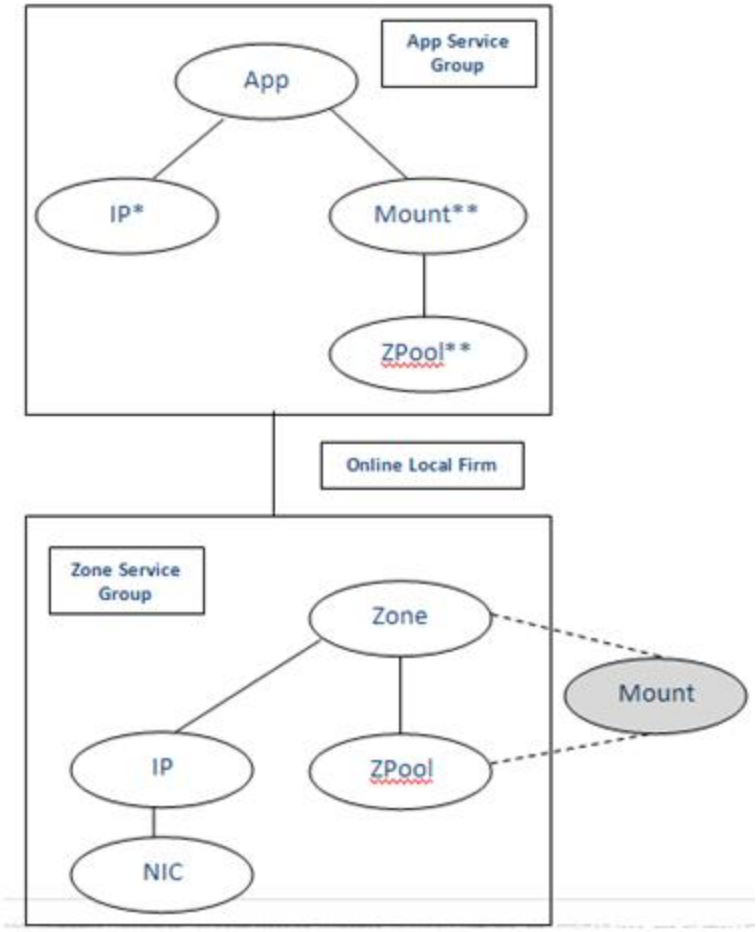


Figure 15: Service Group dependency for Local Zones using ZFS

*This would be the floating Virtual IP that is associated with the application and not the Zone itself.

**This would be the zpool/mount points for the Application Data. You will want to set the path relative the Zone root. For example, if the Zone root is /export/Zones/zonename, and the application wants to access /data then the mount point attribute would be /export/zones/zonename/root/data. You will also want to make sure the file system type is ZFS and that the actual Mountpoint property for the Zpool and FS itself is set to legacy.

From Cluster Node 1:

#> zpool create calzone_zroot c1t50060E8005631F51d1 → creates ZPool on specified device

#> zfs create calzone_zroot/calzone_root→ creates ZFS File System for Zone Root

#> zfs set mountpoint=legacy calzone_zroot → disables automount/auto import

#> zfs set mountpoint=legacy calzone_zroot/calzone_root → disables automount

#> zfs list

NAME	USED	AVAIL	REFER	MOUNTPOINT
calzone_zroot	152K	9.60G	32K	legacy
calzone_zroot/calzone_root	31K	9.60G	32K	legacy

#> **zpool export calzone_zroot**

From Cluster Node 2:

#> zpool create endzone_zroot c2t50060E8005631F41d1 ← Only visible to Node 2

#> zfs create endzone_zroot/endzone_root

#> zfs set mountpoint=legacy endzone_zroot

#> zfs set mountpoint=legacy endzone_zroot/endzone_root

#> zfs list

NAME	USED	AVAIL	REFER	MOUNTPOINT
endzone_zroot	4.47G	5.13G	32K	legacy
endzone_zroot/endzone_root	4.47G	5.13G	4.47G	legacy

#> zpool export endzone_zroot

From Either Cluster Node:

♦ **Important note:** For this operation you will need to select a disk device that is visible to both cluster nodes as it will contain the shared application data.

#> **zpool create appdata_zpool c2t50060E8005631F41d1** ← Shared LUN

#> **zfs create appdata_zpool/appdata**

#> **zfs set mountpoint=legacy appdata_zpool**

#> **zfs set mountpoint=legacy appdata_zpool/appdata**

#> **zfs list**

NAME	USED	AVAIL	REFER	MOUNTPOINT
appdata_zpool	152K	9.06G	32K	legacy
appdata_zpool/appdata	31K	9.06G	31K	legacy
endzone_zroot	152K	9.06G	32K	legacy
endzone_zroot/endzone_root	31K	9.06G	31K	legacy

#> **zpool export appdata_zpool**

♣ **Important note:** To ensure that Veritas Cluster Server properly recognizes the newly created ZPools, you must first export the zone root and data storage pool as shown above, therefore allowing Veritas Cluster Server to successfully bring the ZPool resource online.

Adding the Zone Root ZPool Resources

Adding ZPool Resource

From Either Node:
#> hagrp -add local_vxfs_zone_SG
#> hagrp -modify myzone_SG SystemList node1 0 node2 1
#> hagrp -modify myzone_SG Parallel 1
#> hares -add myzone_ZPool Zpool myzone_SG
#> hares -modify myzone_ZPool Critical 0
#> hares -modify myzone_ZPool ChkZFSMounts 0
#> hares -local myzone_ZPool PoolName
#> hares -modify myzone_ZPool PoolName calzone_zroot -sys node1
#> hares -modify myzone_ZPool PoolName endzone_zroot -sys node2
#> hares -modify myzone_ZPool ZoneResName myzone_ZONE
#> hares -modify myzone_ZPool Enabled 1
#> hares –online myzone_ZPool –sys node1
#> hares –online myzone_ZPool –sys node2

♣ **Important note:** you may choose to have the ZPool agent check the status of the ZFS mounts with the **ChkZFSMounts** attribute. This implies the Mount agent is not part of the Service Group. This document assumes the Mount agent will be configured.

Adding the Zone Root Mount and Network Resources

Adding Mount Resource

From Either Node:
#> hares -add zoneroot_MNT Mount myzone_SG
#> hares -modify zoneroot_MNT Critical 0
#> hares -modify zoneroot_MNT CreateMntPt 1
#> hares -local zoneroot_MNT MountPoint
#> hares -modify zoneroot_MNT MountPoint /zones/endzone/base -sys node2
#> hares -modify zoneroot_MNT MountPoint /zones/calzone/base -sys node1
#> hares -local zoneroot_MNT BlockDevice
#> hares -modify zoneroot_MNT BlockDevice endzone_zroot/endzone_root -sys node2
#> hares -modify zoneroot_MNT BlockDevice calzone_zroot/calzone_root -sys node1
#> hares -modify zoneroot_MNT FSType zfs
#> hares -modify zoneroot_MNT FsckOpt %-n
#> hares -modify zoneroot_MNT Enabled 1

#> hares –online zoneroot_MNT –sys node1

#> hares –online zoneroot_MNT –sys node2

#> chmod 700 /zones/calzone/base ß from node1

#> chmod 700 /zones/endzone/base ß from node2

Add NIC Resource

From Either Node:

#> hares -add zone_NIC NIC myzone_SG

#> hares -modify zone_NIC Device bge0

#> hares -modify zone_NIC Critical 0

#> hares -modify zone_NIC Enabled 1

Add IP Resource

From Either Node:

#> hares -add myzone_IP IP myzone_SG

#> hares -modify IP Critical 0

#> hares -modify IP Device bge0

#> hares -local IP Address

#> hares -modify IP Address 10.10.10.4 -sys node1

#> hares -modify IP Address 10.10.10.5 -sys node2

#> hares -modify IP NetMask 255.255.240.0

#> hares -modify IP Enabled 1

Configure Resource Dependencies

From Either Node:

#> hares -link myzone_ZONE zoneroot_MNT

#> hares -link zoneroot_MNT myzone_ZPool

#> hares -link myzone_IP myzone_ZONE

#> hares –link myzone_IP zone_NIC

#> haconf –dump –makero

From Both Cluster Nodes

Define the Zone on each corresponding Physical Node:

Example:

From Node1:

#> zonecfg -z calzone

create -b

set zonepath=/zones/calzone/base

set autoboot=false

```
end
commit
verify
exit

#> zoneadm –z calzone install
#> zoneadm –z calzone boot
#> zlogin –C calzone
```

Follow Prompts for configuring zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

```
#> cp /etc/resolv.conf /zones/calzonezone/base/root/etc
#> cp /etc/nsswtich* /zones/calzone/base/root/etc
```

This procedure will only work consistently for shared-IP zones.

Configure the Second Zone on Node 2

From Node 2:

```
#> zonecfg -z endzone
create -b
set zonepath=/zones/endzone/base
set autoboot=false
end
commit
verify
exit

#> zoneadm –z endzone install
#> zoneadm –z endzone boot
#> zlogin –C endzone
```

Follow Prompts for configuring zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

```
#> cp /etc/resolv.conf /zones/endzone/base/root/etc
#> cp /etc/nsswtich* /zones/endzone/base/root/etc
```

This procedure will only work consistently for shared-IP zones

Bring the IP Resources Online

From Either Node:

```
#> hares –online myzone_IP –sys Node1
#> hares –online myzone_IP –sys Node2
```

From Either Cluster Node:

♦ **Important note:** When configuring Zones to with "Local" Zone root file systems and shared Data mounts, you must create a parallel service group and localize the Pool, Mount Point and ContainerInfo attributes.

A. Run the hazonesetup script on each cluster node.

From Node1:

#> hazonesetup -t -g local_zfs_zone_SG -r localzone_ZONE -z calzone -u z_vcs_calrzone -p password -l -s node1

From Node2:

#> hazonesetup -t -g local_vxfs_zone_SG -r localzone_ZONE -z endzone -u z_vcs_endzone -p password -l -s node2

Bringing Service Group Online

#> hares -modify myzone_ZONE Enabled 1

#> hagrp –online myzone_SG

Figure 16: Local Zone Service Group with ZFS Root

Adding the Application Data Service Group and Resources

Creating the Service Group and adding the ZPool Resource

From Either Node:

#> hagrp –add zfs_application_SG

#> hares -add mydata_ZPool Zpool zfs_application_SG

#> hares -modify mydata_ZPool Critical 0

#> hares -modify mydata_ZPool ChkZFSMounts 0

#> hares -modify myzone_ZPool PoolName appdata_zpool -sys node1,node2

#> hares -modify myzone_ZPool ZoneResName myzone_ZONE

#> hares -modify myzone_ZPool Enabled 1

#> hares -modify zoneroot_MNT FSType zfs

#> hares -modify zoneroot_MNT FsckOpt %-n

#> hares –online myzone_ZPool –sys node1

♣ **Important note:** you may choose to have the ZPool agent check the status of the ZFS mounts with the ChkZFSMounts attribute. This implies the Mount agent is not part of the Service Group. This document assumes the Mount agent will be configured.

Adding the Application Mount and Network Resources

Adding Mount Resource

From Either Node:

#> hares -add mydata_ZFS Mount myzone_SG

#> hares -modify mydata_ZFS Critical 0

#> hares -modify mydata_ZFS CreateMntPt 1

#> hares -local mydata_ZFS MountPoint

#> hares -modify mydata_ZFS MountPoint /zones/calzone/base/root/mydata -sys node1

#> hares -modify mydata_ZFS MountPoint /zones/endzone/base/root/mydata –sys node2

#> hares -modify mydata_ZFS BlockDevice appdata_zpool/appdata

#> hares -modify mydata_ZFS FSType zfs

#> hares -modify mydata_ZFS FsckOpt %-n

#> hares -modify mydata_ZFS Enabled 1

Add IP Resource

From Either Node:

#> hares -add application_IP IP zfs_application_SG

#> hares -modify IP Critical 0

#> hares -modify IP Device bge0

#> hares -modify IP Address 10.10.10.6

#> hares -modify IP NetMask 255.255.240.0

#> hares -modify IP Enabled 1

Adding Application Resource

Please consult the corresponding solutions guide for Veritas Cluster Server on how to configure monitoring for a specific application type.

https://sort.symantec.com/documents/doc_details/sfha/6.0/Solaris/ProductGuides/

Configure Resource Dependencies

From Either Node:

#> hares -link mydata_ZFS mydata_Zpool

#> hares -link my_application_APP application_IP

#> hares -link my_appliaction_APP mydata_ZFS

#> haconf –dump –makero

#> hagrp –online zfs_application_SG –sys node1

Once the Zone service group is configured, you will want configure separate resources for your application(s). The dependency between the zone/application/parent service group and the storage/child service group should be Online Local Firm. Please follow the Veritas Cluster Server solutions guides for specifics on how to configure the supported applications for failover.

https://sort.symantec.com/documents/doc_details/sfha/6.0/Solaris/ProductGuides/

Example 2: Local Zone Root with Shared Data File Systems (CFS Direct Mount)

In this example, the Zone root will reside on ZFS; however the application data will use Cluster File System mounts. As with the previous example, the zone root file systems will be place on SAN attached storage for flexibility. Also, as is the case with scenario 1, the Zones themselves will not be shared but rather the application will be the unit of failover.

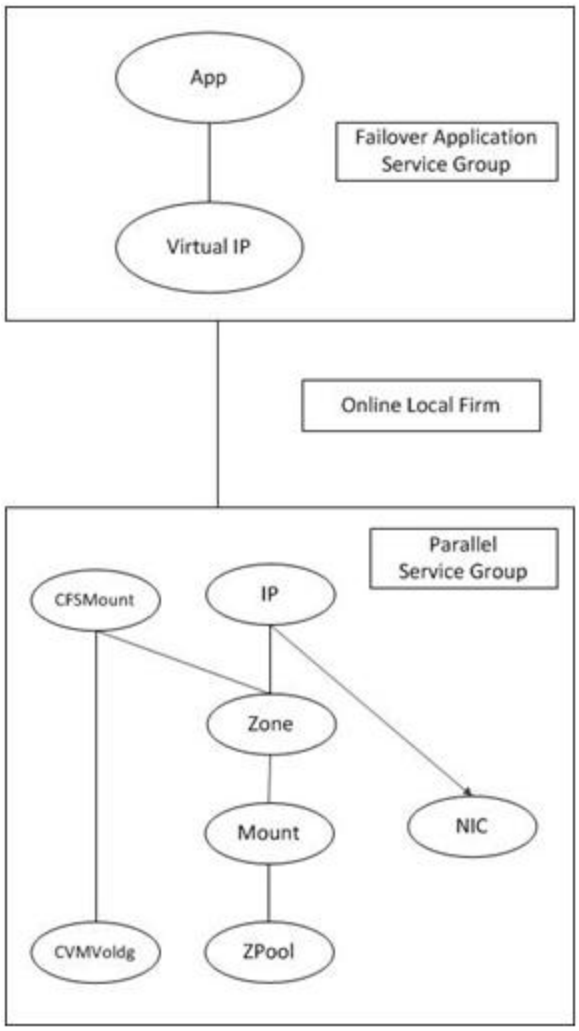


Figure – 18 Service Group Dependency for Direct Mount Cluster File System

From Cluster Node 1:

#> zpool create calzone_zroot c1t50060E8005631F51d1→ creates ZPool on specified device

#> zfs create calzone_zroot/calzone_root → creates ZFS File System for Zone Root

#> zfs set mountpoint=legacy calzone_zroot → disables automount/auto import

#> zfs set mountpoint=legacy calzone_zroot/calzone_root→ disables automount

#> zfs list

NAME	USED	AVAIL	REFER	MOUNTPOINT
calzone_zroot	152K	9.60G	32K	legacy
calzone_zroot/calzone_root	31K	9.60G	31K	legacy

#> zpool export calzone_zroot

From Cluster Node 2:

#> zpool create endzone_zroot c2t50060E8005631F41d1←Only visible to Node 2

#> zfs create endzone_zroot/endzone_root

#> zfs set mountpoint=legacy endzone_zroot

#> zfs set mountpoint=legacy endzone_zroot/endzone_root

#> zfs list

NAME	USED	AVAIL	REFER	MOUNTPOINT
endzone_zroot	4.47G	5.13G	32K	legacy
endzone_zroot/endzone_root	4.47G	5.13G	4.47G	legacy

#> zpool export endzone_zroot

♣ **Important note:** To ensure that VCS properly recognizes the newly created ZPools, you must first export the zone root and data storage pool as shown above, therefore allowing VCS to successfully bring the ZPool resource online.

From Either Cluster Node:

♦ **Important note:** When configuring Zones to with "Local" Zone root file systems and shared Data mounts, you must create a parallel service group and localize the Pool, Mount Point and ContainerInfo attributes.

Build The Zone Service Group

#> haconf –makerw
#> hagrp -add zone_zfs_SG
#> hagrp -modify zone_zfs_SG SystemList node1 0 node2 1
#> hagrp -modify zone_zfs_SG Parallel 1

♣ **Important note:** You will want to leave the Zone resource disabled prior to actually configuring and installing the Zone itself.

Adding the Zone Root ZPool Resources

Adding ZPool Resource

From Either Node:
#> hares -add myzone_ZPool Zpool zone_zfs_SG
#> hares -modify myzone_ZPool Critical 0
#> hares -modify myzone_ZPool ChkZFSMounts 0
#> hares -local myzone_ZPool PoolName
#> hares -modify myzone_ZPool PoolName calzone_zroot -sys node1
#> hares -modify myzone_ZPool PoolName endzone_zroot -sys node2
#> hares -modify myzone_ZPool ZoneResName myzone_ZONE
#> hares -modify myzone_ZPool Enabled 1

#> hares –online myzone_ZPool –sys node1

#> hares –online myzone_ZPool –sys node2

♣ **Important note:** you may choose to have the ZPool agent check the status of the ZFS mounts with the ChkZFSMounts attribute. This implies the Mount agent is not part of the Service Group. This document assumes the Mount agent will be configured.

Adding the Zone Root Mount and Network Resources

Adding Mount Resource

From Either Node:

#> hares -add zoneroot_MNT Mount zone_zfs_SG

#> hares -modify zoneroot_MNT Critical 0

#> hares -modify zoneroot_MNT CreateMntPt 1

#> hares -local zoneroot_MNT MountPoint

#> hares -modify zoneroot_MNT MountPoint /zones/endzone/base -sys node2

#> hares -modify zoneroot_MNT MountPoint /zones/calzone/base -sys node1

#> hares -local zoneroot_MNT BlockDevice

#> hares -modify zoneroot_MNT BlockDevice endzone_zroot/endzone_root -sys node2

#> hares -modify zoneroot_MNT BlockDevice calzone_zroot/calzone_root -sys node1

#> hares -modify zoneroot_MNT FSType zfs

#> hares -modify zoneroot_MNT FsckOpt %-n

#> hares -modify zoneroot_MNT Enabled 1

#> hares –online zoneroot_MNT –sys node1

#> hares –online zoneroot_MNT –sys node2

#> chmod 700 /zones/calzone/base ß from node1

#> chmod 700 /zones/endzone/base ß from node2

Add NIC Resource

From Either Node:

#> hares -add zone_NIC NIC zone_zfs_SG

#> hares -modify zone_NIC Device bge0

#> hares -modify zone_NIC Critical 0

#> hares -modify zone_NIC Enabled 1

Add IP Resource

From Either Node:

#> hares -add myzone_IP IP zone_zfs_SG

#> hares -modify IP Critical 0

#> hares -modify IP Device bge0

#> hares -local IP Address

#> hares -modify IP Address 10.10.10.4 -sys node1

#> hares -modify IP Address 10.10.10.5 -sys node2

#> hares -modify IP NetMask 255.255.240.0

#> hares -modify IP Enabled 1

Configure Resource Dependencies

From Either Node:

#> hares -link myzone_ZONE zoneroot_MNT

#> hares -link zoneroot_MNT myzone_ZPool

#> hares -link myzone_IP myzone_ZONE

#> hares –link myzone_IP zone_NIC

#> haconf –dump –makero

Create CVM Disk Groups, Volumes and File Systems for Application Data

From CVM Master node: vxdctl –c mode:

#> vxdg -s init mydata_dg mydata_dg01=hitachi_usp-vm0_083e

#> vxassist –g mydata_dg make mydata_vol 1g mydata_dg01

#> mkfs –F vxfs /dev/vx/rdsk/mydata_dg/mydata_vol

From Node1:

#> mkdir /zones/endzone/mydata

From Node2:

#> mkdir /zones/calzone/mydata

From CVM Master:

♣ **Important Note:** For this scenario the commands for adding a Cluster Mount to your VCS configuration will require that you first choose the same entry for the MountPoint argument for the **cfsmntadm** command followed by modifying the Mount Point Attribute using the **hares –modify** flag so that it is localized to each Host.

#> cfsmntadm add mydata_dg mydata_vol /zones/mydata zone_zfs_SG node1=suid,rw node2=suid,rw→ This will add a CVMVoldg and CFSMount Resource to the local_vxfs_zone_SG service group

#> hares -local cfsmount1 MountPoint → The naming convention of cfsmount# is the default naming scheme for adding CFSMount resources. You may choose to modify the resource name offline by editing the main.cf or via the copy/paste function on the VCS Java GUI.

#> hares -modify cfsmount1 MountPoint /zones/calzone/mydata -sys node1

#> hares -modify cfsmount1 MountPoint /zones/endzone/mydata -sys node2

#> hares -modify cfsmount1 NodeList node1 node2

#> hares -modify cfsmount1 Primary node1

#> haconf –dump –makero

#> hagrp –online local_vxfs_zone_SG –any

From Both Cluster Nodes

Define the Zone on each corresponding Physical Node:

Example:

From Node1:

```
#> zonecfg -z calzone
create -b
set zonepath=/zones/calzone/base
set autoboot=false
end
commit
verify
exit
```

```
#> zoneadm –z calzone install
#> zoneadm –z calzone boot
#> zlogin –C calzone
```

Follow Prompts for configuring zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

```
#> cp /etc/resolv.conf /zones/calzonezone/base/root/etc
#> cp /etc/nsswtich* /zones/calzone/base/root/etc
```

This procedure will only work consistently for shared-IP zones.

Configure the Second Zone on Node 2

From Node 2:

```
#> zonecfg -z endzone
create -b
set zonepath=/zones/endzone/base
set autoboot=false
end
commit
verify
exit
```

```
#> zoneadm –z endzone install
#> zoneadm –z endzone boot
#> zlogin –C endzone
```

Follow Prompts for configuring zone. Once complete you can use the keystroke ~. to exit the console and return to the OS prompt.

#> cp /etc/resolv.conf /zones/endzone/base/root/etc

#> cp /etc/nsswtich* /zones/endzone/base/root/etc

This procedure will only work consistently for shared-IP zones

Bring the IP Resources Online

From Either Node:

#> hares –online myzone_IP –sys Node1

#> hares –online myzone_IP –sys Node2

From Either Cluster Node:

♦ **Important note:** When configuring Zones to with "Local" Zone root file systems and shared Data mounts, you must create a parallel service group and localize the Pool, Mount Point and ContainerInfo attributes.

A. Run the hazonesetup script on each cluster node.

From Node1:

#> hazonesetup -t -g zone_zfs_SG -r myzone_ZONE -z calzone -u z_vcs_calrzone -p password -l -s node1

From Node2:

#> hazonesetup -t -g zone_zfs_SG -r myzone_ZONE -z endzone -u z_vcs_endzone -p password -l -s node2

Bringing Service Group Online

#> hares -modify myzone_ZONE Enabled 1

#> hares –link cfsmount1 myzone_ZONE

#> hagrp –online zone_zfs_SG -any

Figure - 19 ZFS and Cluster File System Coexistence

Once the Zone service group is configured, you will want configure separate resources for your application(s). The dependency between the zone/application/parent service group and the storage/child service group should be Online Local Firm. Please follow the Veritas Cluster Server solutions guides for specifics on how to configure the supported applications for failover.

https://sort.symantec.com/documents/doc_details/sfha/6.0/Solaris/ProductGuides/

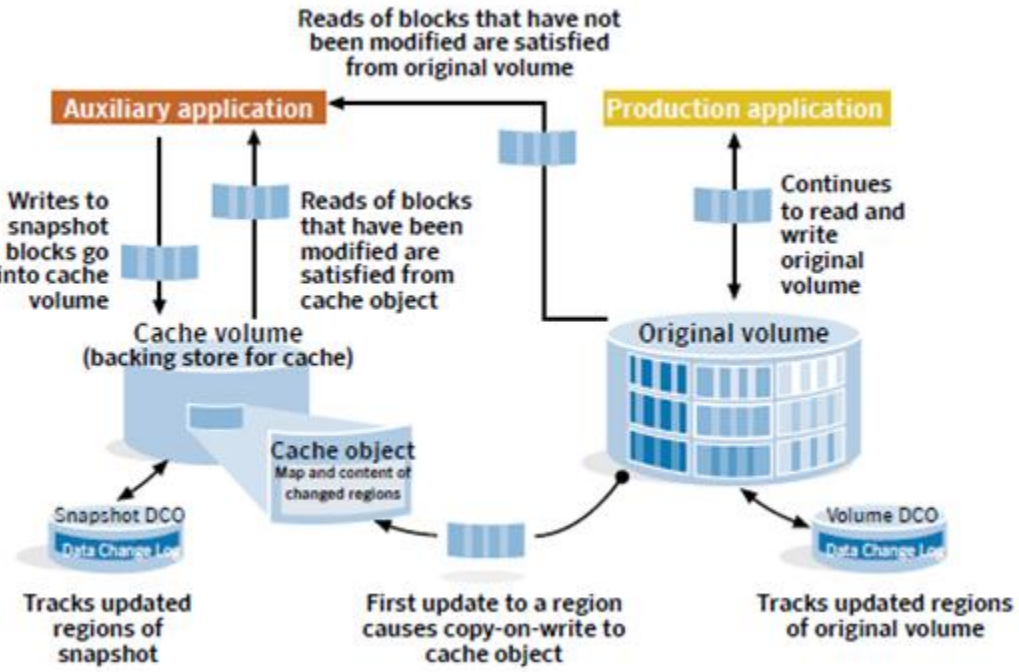
Appendix B: Zone Provisioning with Cluster File System and FlashSnap

FlashSnap Overview:

A snapshot is a virtual image of the content of a set of data at the instant of creation. Physically, a snapshot may be a full (complete bit-for-bit) copy of the data set, or it may contain only those elements of the data set that have been updated since snapshot creation. The latter are sometimes referred to as copy-on-first-write snapshots, because space for data elements is added to the snapshot image only when the elements are updated (overwritten) for the first time in the original data set. Storage Foundation copy-on-first-write snapshots are called space-optimized snapshots.

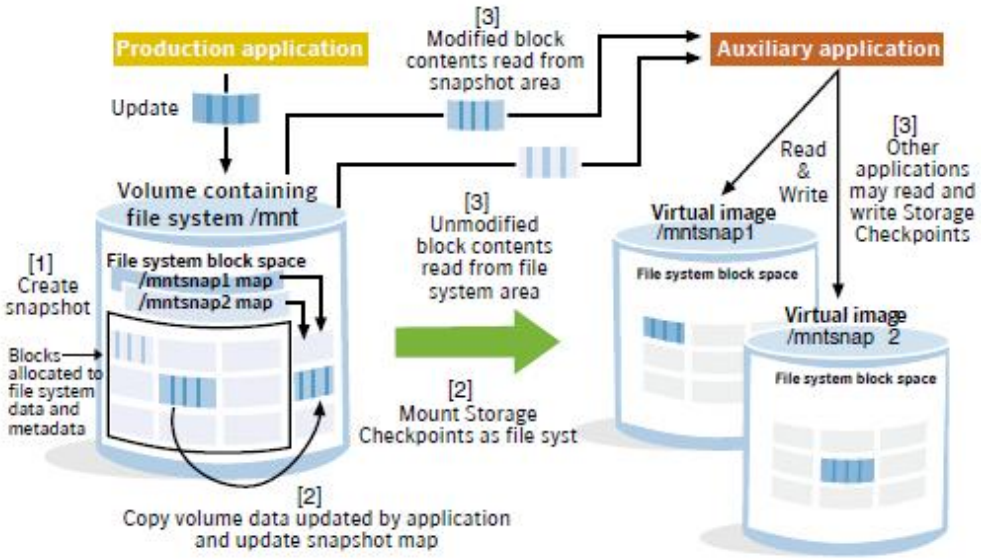
Space Optimized Instant Snap Shots (SOSS):

A Space-Optimized Instant Snapshot uses a disk-based persistent cache object, located on a cache volume, to store prior images of original volume block regions as applications update blocks in them. Cache objects map the addresses of updated block regions of the original volume to cache volume locations where their prior contents are saved. In addition, when VxVM creates a Space-Optimized Instant Snapshot, it creates DCOs for both the original volume and the snapshot. The DCOs are used to track updates to both the original volume and its snapshot.



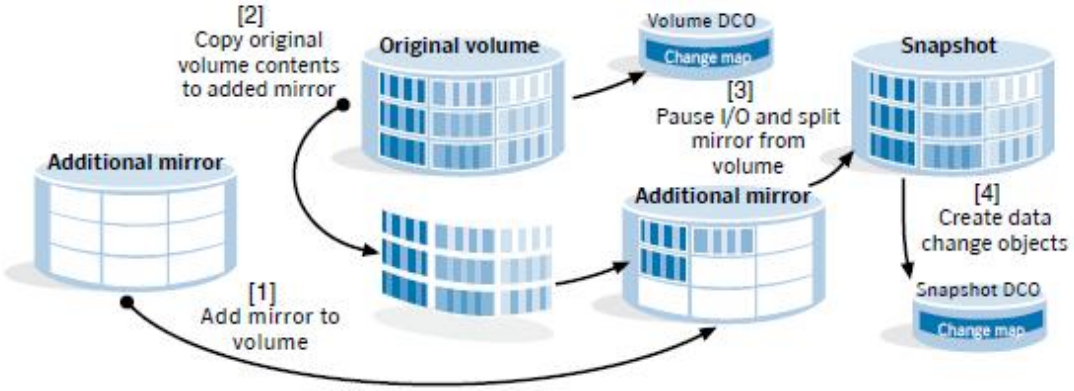
Storage Checkpoints

Storage Checkpoints are allocate-on-first-write snapshots of a VxFS file system. They are entirely contained within the file system block space, and require no external volumes or other structures. A Storage Checkpoint consists of a map that initially contains indirect pointers to the locations of data and metadata in the original file system. The map becomes populated with pointers to prior images of data and metadata as it is updated in the original file system over time.



Full Size Instant Snap Shots (Split Mirror)

Split Mirror Snapshots are fully synchronized mirrors (called plexes, in VxVM terminology) of volume contents that have been split from their original volumes at opportune moments and started as independent volumes, either on or off-host for application use. Split Mirror Snapshot creation begins with the introduction of an additional mirror to a VxVM volume.



Storage Foundation and FlashSnap Use Cases for Zones:

The storage resources required for Whole-root Solaris Zones respond very well to disk based copy solutions and as such the use of snapshot facilities is very advantageous for a variety of operations. Nowhere is this more prevalent than in the case of zone provisioning. Whether you want to expedite the deployment of zones, implement an instantaneous roll back procedure or reduce the storage consumed by an increased number of temporary OS instances (i.e. for dev, test and lab operations), Storage Foundation has the corresponding snapshot type to support these efforts.

Use Case 1: Data Corruption and Patch Roll Back

In this scenario, a Solaris zone whose root resides on VxFS will have a Storage Check Point associated to its root file system.

Use Case 2: New Zone Provisioning with CFS and Space Optimized Snapshots (SOSS)

In this scenario, an unconfigured Solaris zone root file system residing on VxFS will have one or more Space Optimized Snap Shots (SOSS) attached to the parent volume.

Use Case 3: New Zone provisioning off host with full split-mirror snapshots

In this scenario, a Solaris zone whose root file system is on VxFS will have a full sized snapshot mirror attached to its root volume

Use Case 4: Pre-Staging Patches for Local Zone

Details for this option are covered in Appendix C.

Zone Root Roll Back using VxFS Checkpoints

The purpose of this configuration is to provide instantaneous rollback for any changes made to a zone root file system. These may include failed application installs, corrupted packages or inadvertent deletion or modification of critical system files. There is no infrastructure prerequisite as all data is stored with the context of the file system and therefore requires no additional storage space. It is however important to note that during this procedure an outage will be necessary to perform any restore operations.

Creating the Roll Back Checkpoint

Creating the Storage Checkpoint

From Either Node:
#> fsckptadm create cfs_zroot_ckpt /zones/cfs-zroot/base

Listing Checkpoints

#> fsckptadm -l list /zones/cfs-zroot/base

/zones/cfs-zroot
cfs_zroot_ckpt:

ctime = November 5, 2012 09:45:33 PM PST
mtime = November 5, 2012 10:07:52 PM PST
flags = largefiles, removable, mounted,

Mounting/Unmounting Checkpoint (To restore Individual Files)

From Either Node:
#> mkdir /zones/cfs-zroot_ckpt
#> mount –F vxfs –o cluster,ckpt= cfs_zroot_ckpt /dev/vx/dsk/cfs_zroot_dg/cfs_zroot_vol:cfs_zroot_ckpt /zones/cfs_zroot_ckpt
#> umount /zones/cfs_zroot_ckpt

Freezing All Corresponding Zone Service Groups (Assuming Applications are already offline)

#> hagrp –freeze "Service Group"

Unmounting Zone Root File System

From Node1:

#> /opt/VRTS/bin/umount –o mntunlock=VCS /zones/cfs-zroot/base

From Node2 (If using CFS)

#> /opt/VRTS/bin/umount –o mntunlock=VCS /zones/cfs-zroot/base

Restoring Entire Zone Root File System

This process will, upon completion, delete the storage checkpoint. You may choose to create a new checkpoint before remounting the Zone root file system.

From Either Node:

#> /opt/VRTSvxfs/sbin/fsckpt_restore /dev/vx/dsk/cfs_zroot_dg/cfs_zroot_vol cfs_zroot_ckpt /dev/vx/dsk/cfs_zroot_dg/cfs_zroot_vol:

cfs_zroot_ckpt:
ctime = November 5, 2012 09:45:33 PM PST
mtime = November 5, 2012 09:45:33 PM PST
flags = largefiles, removable

UX:vxfs fsckpt_restore: WARNING: V-3-24640: Any file system changes or storage checkpoints made after November 5, 2012 09:45:33 PM PST will be lost.

Restore the filesystem from storage checkpoint cfs_zroot_ckpt ? (ynq) y
(Yes)
UX:vxfs fsckpt_restore: INFO: V-3-23760: File system restored from cfs_zroot_ckpt

Unfreezing Service Group

From Either Node:

#> hagrp –unfreeze "Service Group"

Mounting Zone Root File System and Booting Zone

From Either Node:

#> hares –online "Storage Service Group" –any
#> hares –online "Zone/Application Service Group" –sys node1

Zone Provisioning with Cluster File System and Space Optimized Snapshots

Cluster File System Users may choose to provision zones simply by creating a golden, unconfigured image on a CVM volume and subsequently use the SOSS facility to establish new read/write volumes. Once created, these newly minted zone images may be configured as they would with any traditional zonecfg and zoneadm command syntax. As stated previously, the only infrastructure prerequisite for this operation is the existence of available space in the disk group so as to create the SOSS cache volume.

Preparing a Zone for Snapshot Provisioning

The process to provision new zones using Cluster File System and Space Optimized Snapshots is relatively straightforward. Zone root volumes are treated no differently than any other data source when configuring these types of snapshots. The process begins with installing but not configuring a new zone root image. This will act as the golden image source for all other zone roots and therefore only requires to be installed one time.

Create Source Zone Root File System

Please refer to the section that previously covered creating disk groups, volumes and file systems for use with zone roots.

Preparing a Golden Image Zone (One time Operation)

From Either Node:
#> zonecfg –z gold_image_zone
gold_image_zone: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:gold_image_zone > **create -b**
zonecfg:gold_image_zone > **set zonepath=/zones/gold_image_zone**
zonecfg:gold_image_zone > **set autoboot=false**
zonecfg:gold_image_zone > **commit**
zonecfg:gold_image_zone > **verify**
zonecfg:gold_image_zone > **exit**

#> chmod 700 /zones/gold_image_zone
#> zoneadm –z gold_image_zone install
#> zoneadm –z gold_image_zone detach
#> umount /zones/gold_image_zone

Creating the Zone Root Snapshot

Preparing Zone Root Volume for Snapshot Operations (One Time Operation)

#> vxsnap –g cfs_zroot_dg prepare cfs_zroot_vol

Establishing the Cache Volume (One Time Operation)

#> vxasssit –g cfs_zroot_dg make cfs_zroot_cachevol alloc=cfs_zroot_dg01

Creating the Cache Object (One Time Operation)

#> vxmake -g cfs_zroot_dg cache cfs_zroot_cachobj cachevolname=cfs_zroot_cachevol autogrow=on

Establishing Point-in-Time Space Optimized Snapshot

#> vxsnap -g cfs_zroot_dg make source=cfs_zroot_vol/newvol=newzone1_zroot_vol/cache= cfs_zroot_cacheobj

Verifying Snapshot

#> vxsnap –g cfs_zroot_dg list

#> vxcache –g cfs_zroot_dg stat

Creating a New Zone Using a Space Optimized Snapshot

The following commands make the assumption that the snapshot will be mounted as traditional VxFS file system as opposed to Cluster File System. That said, Cluster File System can certainly be used as long as the DG is imported with "Shared" flag. Much of the commands displayed here can be modified and automated based on your particular environment. What is important to note is that the time to clone a whole root zone and assign various configuration parameters is substantially reduced.

#> mkdir /zones/newzone1

#> mount –F vxfs /dev/vx/dsk/cfs_zroot_dg/newzone1_zroot_vol /zones/newzone1

#> chmod 700 /zones/newzone1

#> zonecfg –z newzone1

newzone1: No such zone configured

Use 'create' to begin configuring a new zne.

zonecfg:newzone1> **create -b**

zonecfg:newzone1> **set zonepath=/zones/newzone1**

zonecfg:newzone1> **set autoboot=false**

zonecfg:newzone1> **commit**

zonecfg:newzone1> **verify**

zonecfg:newzone1> **exit**

#> zoneadm –z newzone1 attach -F

#> zoneadm –z newzone1 boot

#> zlogin –C newzone1 à Follow the normal process for configuring Zone OS.

Zone Provisioning with Cluster File System and Full Size Instant Snapshots

For those environments that require an even greater degree of zone flexibility or the use of off-host processing, the full sized snapshot option will satisfy both of these requirements. By incorporating "Snapshot Mirrors" to an either active zone root volume or an unconfigured golden image, users will be able to provision new zones or stage a cloned zone root for purpose of patch certification. The following procedure outlines the steps required to establish a full-sized instant snapshot for the purposes of zone provisioning.

Preparing a Zone for Snapshot Provisioning

The process to provision new zones using full sized instant snapshots will require duplicate storage capacity. Zone root volumes are treated no differently than any other data source when configuring these types of snapshots. The process begins with installing but not configuring a new zone root image. This will act as the golden image source for all other zone roots and therefore only requires to be installed one time.

Create Source Zone Root File System

Please refer to the section that previously covered creating disk groups, volumes and file systems for use with zone roots.

Preparing a Golden Image Zone (One time Operation)

From Either Node:

#> zonecfg –z gold_image_zone

gold_image_zone: No such zone configured

Use 'create' to begin configuring a new zone.

zonecfg:gold_image_zone > **create -b**

zonecfg:gold_image_zone > **set zonepath=/zones/gold_image_zone**

zonecfg:gold_image_zone > **set autoboot=false**

zonecfg:gold_image_zone > **commit**

zonecfg:gold_image_zone > **verify**

zonecfg:gold_image_zone > **exit**

#> chmod 700 /zones/gold_image_zone

#> zoneadm –z gold_image_zone install

#> zoneadm –z gold_image_zone detach

#> umount /zones/gold_image_zone

Creating the Zone Root Snapshot

Preparing Zone Root Volume for Snapshot Operations (One Time Operation)

#> vxsnap –g cfs_zroot_dg prepare cfs_zroot_vol

Adding a Snapshot Mirror (Assign a free disk for this operation)

#> vxsnap -b -g cfs_zroot_dg addmir cfs_zroot_vol alloc=cfs_zroot_dg02

You will need to wait for the synchronization of the mirrors to complete before proceeding.

Taking a Full Sized Snapshot

#> vxsnap -g cfs_zroot_dg make source=cfs_zroot _vol/newvol=newzone_vol/plex=cfs_zroot _vol-02

Splitting the Disk Group (Private DG)

#> vxdg split cfs_zroot_dg newzone_zroot_dg newzone_vol

#> vxdg deport newzone_zroot_dg

#> vxdg import newzone_zroot_dg (on alternate host)

Splitting the Disk Group (Shared/CVM DG)

#> vxdg split cfs_zroot_dg newzone_zroot_dg newzone_vol

#> vxrecover -b -m -g cfs_zroot_dg

#> cfsdgadm move cfs_zroot_dg newzone_zroot_dg

Mount New Snapshot Zone Root (Alternate Host)

#> mkdir –p /zones/newzone/base

#> mount –F vxfs /dev/vx/dsk/newzone_root_dg/newzone_vol /zones/newzone/base

#> chmod 700 /zones/newzone/base

Mount New Snapshot Zone Root (CFS)

#> haconf -makerw

#> mkdir –p /zones/newzone/base

#> cfsmntadm add newzone_zroot_dg newzone_zroot_vol /zones/newzone/base newzone_SG

node1=rw,suid

#> hagrp –online newzone_SG –any

#> haconf –dump -makero

#> chmod 700 /zones/newzone/base

Creating a New Zone Using a Split Mirror Snapshot

Once the Cluster File System mount is added to the cluster configuration, you will need to export the zone configuration to each node

#> zonecfg –z newzone

newzone1: No such zone configured

Use 'create' to begin configuring a new zone.

zonecfg:newzone1> **create -b**

zonecfg:newzone1> **set zonepath=/zones/newzone/base**

zonecfg:newzone1> **set autoboot=false**

zonecfg:newzone1> **commit**

zonecfg:newzone1> **verify**

zonecfg:newzone1> **exit**

Export Zone Configuration to All Cluster Nodes

From Node1:

#> zonecfg –z newzone export –f /tmp/newzonezone.cfg

#> scp /tmp/newzone node2:/tmp (repeat for all nodes)

From All Remaining Nodes:

#> zonecfg –z newzonezone –f /tmp/newzonezone.cfg

From Node1:

#> zoneadm –z newzone attach -F

#> zoneadm –z newzone boot

#> zlogin –C newzone à Follow the normal process for configuring Zone OS.

Adding the New Zone to Your Veritas Cluster Server Configuration

From Node1:

Please refer to the previous sections for the steps on how to add a new zone to a Veritas Cluster Server configuration.

Appendix C: Applying Patches to Systems with Zones Under Veritas Cluster Server Control

Much of the discussion surrounding Solaris zones centers on the best practices for system patching. Prior to Solaris 10 8/07, the lack of an attach/detach function made this a rather challenging process. That said, with the incorporation of the detach option and more recently the Update on Attach feature, Solaris zone patching is a decidedly easier operation than was the case with earlier iteration of Solaris 10.

As mentioned previously in this guide, the decision of where to place the zone root file system will have a direct effect on the process you must follow to apply system patches. For those environments where the zone itself remains resident while the application is the unit of failover, consistency between the global zone and local zones is less of a challenge. However, what will be a hurdle to overcome is when an individual zone requires that you apply a particular patch, but due to nature of the zone, i.e. being defined only on single host, you will be unable to move the environment to an alternate host so as to avoid testing in production. Moreover, a roll-back target would not exist to restore the zone root in the event of corruption resulting for a failed package or patch install.

Due to the way in which the patchadd and pkgadd commands work from the global zone, the net result of this operation will be that all attached local zones are patched concurrently. This is not the most ideal circumstance for an untested patch to be installed on a production system. Alternatively, for those zones where the root file system is configured on VxFS (CFS) a snapshot, either full-sized or space-optimized, can be taken in order to provide a duplicate image of the zone root that may be tested upon off-host. Combined with the checkpoint process previously outlined, the effect of patching a zone can be mitigated and the process successfully vetted without disrupting any production applications.

The following procedures (in conjunction with the aforementioned Flashsnap operations) will provide the high-level steps necessary for testing Solaris patches for both local and shared zones either via the space optimized or full-sized snapshot method. One important consideration is that each zone has been configured without an IP address assigned. This allows the snapshot zone to be booted concurrently with the original environment so as to avoid causing an IP address conflict.

Space Optimized Method using Cluster File System

The main caveat to this process is the existence of a node within the cluster that is eligible to be patched (i.e. one that does not currently host production applications). This is most commonly available in a N to N cluster topology where a specific node can be used as the patch target. Preferably choose a node not in the systems list for the application or zone service groups.

1. Establish cache volume and cache object for zone root volume
2. Create Space Optimized Snapshot of zone root volume
3. Export the configuration for the original zone to designated patch testing node. You will want to remove any LOFS definitions for data file systems on the snapshot zone
4. Mount the Space Optimized Snapshot on the "Patch Node" using same path as with the source file system
5. Define the temporary zone on the new host using zonecfg
6. Boot the snapshot local zone
7. Proceed with patch testing.
8. Shutdown and detach snapshot zone
9. Unmount snapshot root file system

At this point the snapshot can be refreshed at any time to create an up-to-date zone root for further testing. Although these steps only cover the zone root volume, application data can also be incorporated into this procedure very easily. Choosing to do so will then allow for patch testing of application binaries in addition to the OS itself.

Full Sized Snapshot Method

When choosing the full-sized snapshot method, users will be able to move the snapshot zone root off-host to entirely independent node. This can act as both a provisioning as well patch testing procedure.

1. Add second disk as a snapshot mirror to zone root volume
2. Take snapshot of zone root
3. Split disk group so as to create a new DG using only the snapshot zone root volume
4. Deport and import the new DG to alternate node
5. Mount the snapshot volume/zone root on the target host using the same directory path as with the original file system.
6. Export the configuration for the original zone to the designated patch testing node.
7. Configure the snapshot zone for the new global using the exported cfg file from above (zonecfg –z testzone –f testzone.cfg)
8. Remove any LOFS data entries from the zone configuration.
9. Attach and boot snapshot zone
10. Apply any new packages or patches to the snapshot zone.
11. Halt snapshot zone on alternate host and detach it
12. Deport the new DG from the alternate host
13. Import and join snapshot dg and source dg on production host
14. Reattach/refresh zone root snapshot mirror

About Symantec

Symantec protects the world's information, and is a global leader in security, backup, and availability solutions. Our innovative products and services protect people and information in any environment – from the smallest mobile device, to the enterprise data center, to cloud-based systems. Our world-renowned expertise in protecting data, identities, and interactions gives our customers confidence in a connected world. More information is available at www.symantec.com or by connecting with Symantec at go.symantec.com/socialmedia.

For specific country offices and contact numbers, please visit our website.

Symantec World Headquarters
350 Ellis St.
Mountain View, CA 94043 USA
+1 (650) 527 8000
1 (800) 721 3934
www.symantec.com

2/2013 21281112