



Implementing Solaris™ Zones with Veritas™ Cluster Server by Symantec

July 2006

*Eric Hennessey
Group Technical Product Manager
Unix Clustering Solutions*

Implementing Solaris Zones with Veritas Cluster Server by Symantec

Contents

Introduction	4
Interaction between Veritas Cluster Server (VCS) and Solaris local zones	5
VCS agent framework changes	5
VCS agent changes	6
Configuring a local zone to work with VCS	7
Sample local zone config	7
Making the local zone “VCS Ready”	8
VCS service group configuration	9
Establishing interzone communication	11
Best practices for local zone configuration in VCS	12
Applying patches to systems with zones under VCS control	12
Developing custom agents that support zones	13
Modifying an existing agent to support local zones	14
Summary	15
Acknowledgements	15

Introduction

With the release of Solaris 10, Sun introduced the concept of local zones. Zones are an isolation mechanism for applications in which an application is started 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 stepping on other applications. From an implementation view, a local zone does not instantiate a separate operating system kernel as is done with virtual machines; instead zones operate as resource “containers.” Zones allow the system administrator to isolate an application and manage system resource allocation between applications running in other zones. Zones extend the concept of resource management from simply controlling resource allocations between applications to more robust isolation, where one application cannot affect the operation of another. Local zones run under the control of the base operating system, which when coupled with local zones, is called the global zone.

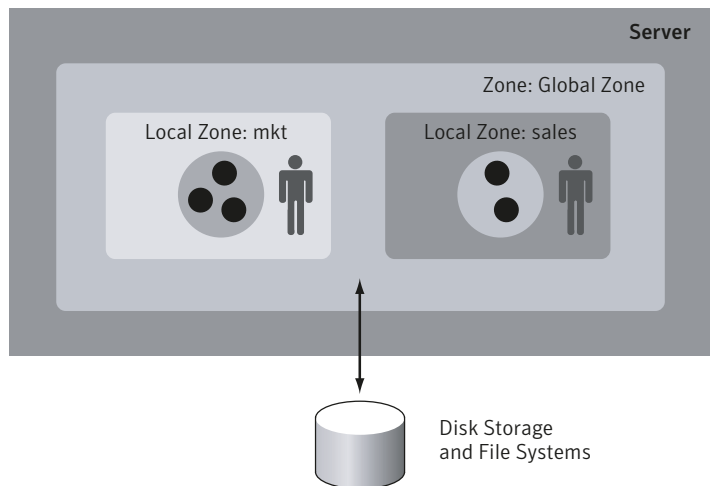


Figure 1. Relationship of local zones to the global zone

For more information on zones and resource management, refer to the Sun Blueprint “Solaris Containers—What They Are and How to Use Them” (May 2005, no. 819-2679-10).

Support for Solaris Zones in a clustered environment was added beginning with Veritas Cluster Server (VCS) version 4.1. A system administrator can now 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 system administrators with the information needed to correctly configure local zones in a VCS cluster, and to provide best practices for systems maintenance when local zones are placed under VCS control. Best practices noted in the document will be preceded by this symbol: ✓ By the same token, we'll attempt to address configurations and actions that should specifically be avoided. Such topics will be denoted with the following symbol: ✗

Prior knowledge of and experience with VCS is assumed.

Interaction between VCS and Solaris local zones

In a Solaris Zones environment, VCS always runs in the global (root) zone of the Solaris operating system. VCS 4.1 can manage applications running in the global zone in what can be considered a “classic” way of managing applications. It also can manage starting and stopping of local zones and the applications contained within the zones.

Management of local zones and applications within the zones required many changes to the basic VCS agent framework, a new zone agent, and modifications to several existing agents. This is all described below.

VCS agent framework changes

The VCS agent framework is a core set of functions compiled into every agent. The agent framework is responsible for connecting with the VCS engine (HAD) and carrying out core agent logic. The VCS agent framework has been modified in version 4.1 to understand the concept of containers. Container, in this context, is a generic term that lets the VCS agent framework know that some functions may have to be carried out in a different way than in a “normal” VCS environment. For example, in a Solaris Zones environment, the VCS agents run with the core VCS daemons in the global zone. For specific resource types, the agent may need to control processes running within a local zone rather than the global zone. At the same time, VCS is expected to operate in environments not using zones, so agents must be capable of operating in either mode.

Implementing Solaris Zones with Veritas Cluster Server by Symantec

The VCS 4.1 agent framework has a new static attribute called “ContainerType.” For agents that have been modified to operate in a zone environment, the ContainerType attribute is set to “Zone.” These include agents that are expected to manage resources that may operate within a local zone in some environments and the global zone in others. The second attribute added at the agent framework level is called “ContainerName.” If ContainerName is a null string, then the application defined by that agent will run in the global (root) zone. If set to the name of a valid, configured zone, the agent will start, stop, and monitor the application within the confines of the named local zone.

VCS agents operate on resources within local zones via an interface provided by the Solaris Zones system. This interface allows a root-level user to run a script that is executed within a specified local zone. With the new agent framework modifications to support zones, script-based entry points for agents managing resources within a local zone are executed within the specified local zone. Due to the nature of the global/local zone interface, only agent entry points written in a scripting language, such as shell or Perl, are supported. The VCS 4.1 agent support for Solaris Zones does not include support for agent entry points written in C++.

VCS agent changes

Along with agent framework modifications, several core agents have been modified or created to support Solaris Zones. These are described below.

IP/NIC agents

The IP and IPMultiNIC agents have been modified to manage an application’s virtual IP addresses within a local zone, when desired. When a zone is brought up by VCS as part of a service group, the IP addresses within the zone are brought up by agents for the IP or IPMultiNIC resource, rather than by the IP agent of the zone being automatically started by the zone boot configuration. This behavior is needed to allow a zone to be brought up for maintenance reasons, such as applying patches, without the IP address coming up and causing issues in an HA cluster.

The Corresponding NIC and MultiNIC agents that monitor physical NIC resources in the global zone have their ContainerType attribute defaulted to Null, as they are never used in a local zone.

X Important note: While support for Solaris Zones is included in VCS 4.1, the IP and IPMultiNIC agents were not modified for full support until VCS 4.1 MP1. Make sure MP1 is installed prior to configuring local zones to work with VCS.

Zone agent

The Zone agent is a new bundled agent as of VCS 4.1. This agent starts, stops, and monitors a local zone in which an application is intended to run. The Zone agent starts (onlines) a local zone using the Solaris zoneadm boot command, stops it using the zoneadm halt command, and to monitor the zone, uses the zoneadm list command to determine if the zone is in a “running” state.

The Zone resource is configured in the service group as a “child” of the application resource that will run within the zone. [It’s highly unusual to refer to a figure that appears later in the document. It’s customary to refer to figures in the order in which they appear in a document, so if it’s important to reference Figure 6 here, perhaps it should be moved forward in the document (and renumbered).]

Other bundled agents

Along with the bundled Zone agent, several bundled and enterprise agents have been modified to support the use of local zones:

- ✓ Oracle®
- ✓ IBM® DB2®
- ✓ Application
- ✓ Process

Configuring a local zone to work with VCS

When configuring a local zone that will be under VCS control, there is only one deviation from the default configuration: The zone must be configured so that it will not boot when the system is started, as VCS will take care of booting the zone.

Sample local zone config

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

Nodes: birdie and bogey

Local zone name: myzone

Local zone host name: myzone

Zone root file system: /export/home/myzone (local disk storage)

Application data mounted at: /export/home/mp3 (in 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:

```
myzone:installed:/export/home/myzone
```

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 `<zonenumber>.xml`. Figure 2 contains the entire zone configuration for our sample zone.

```
<?xml version="1.0" encoding="UTF-8"?>
<!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="myzone" zonepath="/export/home/myzone" autoboot="false">
  <inherited-pkg-dir directory="/lib"/>
  <inherited-pkg-dir directory="/platform"/>
  <inherited-pkg-dir directory="/sbin"/>
  <inherited-pkg-dir directory="/usr"/>
  <filesystem special="/export/home/mp3" directory="/mp3" type="lofs"/>
</zone>
```

Figure 2. Local zone configuration file in XML format

There are several things worth noting in the above configuration. First, the `autoboot` option is set to `false`. This is necessary because VCS will take care of booting the zone, and is not default zone behavior. Second is the lack of any network information. VCS will make the appropriate IP address available to the zone after it's booted. And last is the loopback file system entry for the file system containing the application's data. This is a VxFS file system mounted in the global zone and made available to the local zone through this entry.

Making the local zone "VCS ready"

The following paragraphs will cover the Solaris commands used to make each of these entries. To leave out the network portion of the zone configuration, just omit any "add net" commands during the zone configuration process. Figure 3 illustrates the commands issued to effect these changes.

```
zonecfg -z myzone    ← Starts zone configuration session
set autoboot=false
add fs
  set dir="/mp3"      ← Directory under zone root to mount file system
  set special="/export/home/mp3" ← Directory file system is mounted in global zone
  set type="lofs"     ← Identifies as loopback file system
end
commit
exit
```

Figure 3. Zone configuration session commands

Note that these steps must be performed on each node in the cluster where the local zone is configured to run.

VCS service group configuration

Configuring a VCS service group with a local zone is a straightforward process. A zone resource is added to the service group, and the only required attribute for the zone resource is the name of the zone. The zone resource is configured as a parent of any file systems (mount resources) required by the application, and as a child resource of the application and the IP resource.

Consider the service group depicted in Figure 4. In this example, we have a Samba service under VCS control. For this configuration, the Samba service is configured to run in the global zone. The Samba service is at the top of the resource dependency tree, with network and storage resources configured as children of the Samba resource.



Figure 4. VCS service group with Samba running in a global zone

The VCS configuration for the Samba service shown in Figure 4 is very simple, since there are no attributes required by VCS to start the Samba service. Figure 5 shows the section from the cluster's main.cf file that describes the Samba configuration.

```
Samba mp3_smb (  
)
```

Figure 5. VCS main.cf extract of the Samba service running in a global zone

In Figure 6, we show the same service group, but with the Samba service running within a local zone, shown in this example as the resource named mp3_zone. Note that the Samba service is dependent upon the virtual IP with the Zone resource configured as a child of the IP resource. This is because the local zone must be booted (online) prior to the IP agent configuring the application's virtual IP inside the zone.



Figure 6. VCS service group with Samba running inside a local zone

Figure 7 shows that the section of main.cf describing the Samba service running in the local zone only has one attribute: the ContainerName indicating the name of the local zone in which the application is to run.

```
IP mp3_ip (
  Device = hme0
  Address = "172.16.1.93"
  NetMask = "255.255.255.0"
  ContainerName = myzone
)
.
.
.
Samba mp3_smb (
  ContainerName = myzone
)

Zone mp3_zone (
  ZoneName = myzone
)
```

Figure 7. Portion of main.cf describing zone-specific configuration options

Establishing interzone communication

Before an agent can perform operations on an application in a local zone, communications must be established between the local and global zones. This document assumes a non-secure cluster. For information on using zones in secure clusters, refer to Appendix B of the VCS User's Guide.

After the zone is created and completely configured, interzone communications must be established. To do so, perform the following steps:

- Add a new cluster user, with group administrator privileges for the group containing the zone.
- Ensure that the local zone can resolve the host name of the global zone, either through DNS or through the /etc/hosts file.
- Make sure the global zone's host name is in DNS or the local /etc/hosts file.
- Log into the local zone (zlogin myzone).
- Set the environment variable VCS_HOST to the host name of the global zone.
- Issue the command /opt/VRTSvcs/bin/halogin <username> <password>.

The above steps need to be performed just once, but they must be performed on each node that will run the zone under VCS control.

Best practices for local zone configuration in VCS

When configuring the local zone, there are a few things to keep in mind that will help when the zone is under VCS control.

Choosing the location for the zone root file system

VCS 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 zone installation must be performed only once. This has a disadvantage, however, when it is time to apply system patches, as will be described in more detail below.

It's therefore recommended that the zone root file system be installed to local storage.

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, this process has the following disadvantages:

- Since the IP isn't being monitored, IP-related faults won't be detected.
- While applying patches, the local zone will be booted, resulting in an IP address conflict with the node where the service group running the local zone is online.

VCS 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 VCS to control the virtual IP.

Applying patches to systems with zones under VCS control

The typical approach to applying patches to systems in a VCS cluster is to first patch inactive systems, fail applications over to the patched systems at a convenient time, and then patch the remaining systems. This allows the system administrator to apply system patches in a manner that minimizes service disruption.

While it's not strictly necessary to freeze a service group while applying system patches, many people will do it anyway. However, when applying patches to systems in a cluster in which local zones are being managed, freezing the service group becomes imperative. This is because the Solaris patch utility will examine the system being patched for any zones configured on it.

The patch utility will then boot those zones so that the appropriate files and directories on the zone root file system will be patched, too.

✘ If a system is being patched, its local zones will be booted. This will result in a concurrency violation with the active local zone on the system currently hosting the service group. Failure to freeze the service group will cause VCS to shut down the local zone on the node being patched, resulting in a failure of the patch utility on that node.

✘ Failure to place the zone root file system on local storage makes it impossible to patch inactive nodes. This is because the inactive node will not have the file system with the zone's root mounted, which will cause the patch utility to exit with an error because it cannot boot the zone. As a result, application services will have to be shut down during the patch process.

Developing custom agents that support zones

When developing a custom agent for an application, it's a fairly trivial task to make the agent capable of managing that application in a local zone. Agents built to support local zones can also manage applications in the global zone. The only requirement is that any agent designed to support local zones be a script-based agent; that is, the entry points must be implemented as Perl or shell scripts.

It's also quite simple to modify an existing script-based agent to support local zones without interfering with its ability to manage an application in the global zone.

Figure 8 shows the types declaration file for the Samba agent used in this paper.

```
type Samba (
    static str ContainerType = Zone
    static int RestartLimit = 2
    static boolean FireDrill = 1
    static str ArgList[] = { ContainerName }
    str ContainerName
)
```

Figure 8. Samba types declaration file

The addition of the static attribute ContainerType with a default value of "Zone" sets up the agent to be used with local zones. The attribute ContainerName, included in the ArgList attribute, identifies to the agent framework the local zone in which to run the application. If the ContainerName attribute is left blank, the application is run in the global zone.

Modifying an existing agent to support local zones

Assume for a moment that the Samba types declaration shown in Figure 8 lacked the ContainerType and ContainerName attributes. Modifying the types declaration can be done in the following manner:

- **Offline all resources of the type:** `hares-offline mp3_smb-sys bogey`
- **Stop the agent:** `haagent-stop Samba-sys bogey; haagent-stop Samba-sys birdie`
- **Make the configuration writeable:** `haconf-makerw`
- **Add the ContainerType static attribute:** `haattr-add-static Samba ContainerType-string-scalar Zone`
- **Add the ContainerName attribute:** `haattr-add Samba ContainerName`
- **Add the ContainerName attribute to the argument list:** `hatype-modify Samba ArgList-add ContainerName`
- **Save and make the configuration read-only:** `haconf-dump-makero`
- **Start the agent:** `haagent-start Samba-sys bogey; haagent-start Samba-sys birdie`
- **Online any resources of the type:** `hares-online mp3_smb-sys bogey`

Summary

Controlling applications running in Solaris local zones using Veritas Cluster Server is a relatively straightforward process. However, there are configurations that, while supported, are clearly a good idea to avoid. So, remember the best practices when using Solaris Zones with VCS:

- ✓ Install VCS 4.1 MP1 prior to configuring zones with VCS.
- ✓ Place the local zone root file system on local storage.
- ✓ Use the IP and IPMultiNIC agents to manage the IP addresses for local zones.
- ✓ Freeze any service groups containing local zone resources prior to patching other servers in the cluster.

Solaris Zones are a key feature of Solaris 10, and are likely to continue to be enhanced by Sun. Symantec is committed to continued support for local zones, and our changes will reflect those enhancements to ensure the most robust support for this technology.

Acknowledgements

This document and the best practices put forth here were not developed in the vacuum of a lab environment. I'd specifically like to thank two customers—Andrew Blatt of ACB Consulting, and another from whom I've not yet received permission to name—for their experience, expertise, and time in helping to identify the best practices for managing Solaris Zones with Veritas Cluster Server in a real-world environment.

About Symantec

Symantec is the world leader in providing solutions to help individuals and enterprises assure the security, availability, and integrity of their information.

Headquartered in Cupertino, Calif., Symantec has operations in more than 40 countries.

More information is available at www.symantec.com.

For specific country offices and contact numbers, please visit our Web site. For product information in the U.S., call toll-free 1 (800) 745 6054.

Symantec Corporation
World Headquarters
20330 Stevens Creek Boulevard
Cupertino, CA 95014 USA
+1 (408) 517 8000
1 (800) 721 3934
www.symantec.com

Copyright © 2006 Symantec Corporation. All rights reserved. Symantec, the Symantec logo, and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. IBM and DB2 are trademarks of International Business Machines Corporation in the United States, other countries, or both. Solaris is a trademark or registered trademark of Sun Microsystems, Inc., in the U.S. or other countries. Other names may be trademarks of their respective owners. Printed in the U.S.A.
06/06 10726027