Solaris 10 Zone Survival Guide
by Robert Chase

Solaris Zones are a lightweight virtual machine product for the Solaris 10 operating system. You may also see zones referred to as Solaris containers. Older versions of Solaris do not support zones however the zone's themselves can run older versions of Solaris in addition to Linux. Zones present a number of interesting and new challenges to a Systems Administrator. For example the machine you have an issue with may be a real machine on physical hardware or a zone. Knowing the difference can assist you in in making decisions on how to solve the problems you are presented with. While you may be aware of the user impact of the main machine the system could be running a number of Zones which broaden the user impact regarding decisions that you might make to solve the main issue.

Before we get into some of the main commands here are some things that might "hint" that you are unknowingly inside a Solaris Zone.

Here is a df from inside a zone

```
# df -h
Filesystem             size   used  avail capacity  Mounted on
/                      183G   171G    10G    95%    /
/dev                   183G   171G    10G    95%    /dev
/lib                   5.1G   4.4G   573M    89%    /lib
/opt                   5.1G   4.4G   573M    89%    /opt
/platform              5.1G   4.4G   573M    89%    /platform
/sbin                  5.1G   4.4G   573M    89%    /sbin
/usr                   5.1G   4.4G   573M    89%    /usr
proc                     0K     0K     0K     0%    /proc
ctfs                     0K     0K     0K     0%    /system/
contract
swap                   3.7G   176K   3.7G     1%    /etc/svc/
volatile
mnttab                   0K     0K     0K     0%    /etc/mnttab
fd                       0K     0K     0K     0%    /dev/fd
swap                   3.7G     0K   3.7G     0%    /tmp
swap                   3.7G     8K   3.7G     1%    /var/run
```

Here's one on the main host machine. Notice the fundamental difference between them. The Zone does not show physical hardware. This is somewhat misleading though because it is possible to configure physical hardware to be used in a Zone but its certainly a good hint when you don't see physical disks in your df output.

```
# df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0t0d0s0      5.1G   4.4G   573M    89%    /
```

```
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/
contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   3.7G   1.2M   3.7G     1%    /etc/svc/
volatile
objfs                    0K     0K     0K     0%    /system/
object
fd                       0K     0K     0K     0%    /dev/fd
swap                   3.7G     0K   3.7G     0%    /tmp
swap                   3.7G    72K   3.7G     1%    /var/run
/dev/md/dsk/d0         183G   171G    10G    95%    /share
/dev/dsk/c0t0d0s7      2.7G   100M   2.6G     4%    /export/home
/export/home/rchase    2.7G   100M   2.6G     4%    /home/rchase
fibrearray2:/lun1       83G    19G    62G    24%    /lun1
```

Another hint we have that we might be on a machine with a number of Zones on it is the output of ifconfig. Notice the Zones listed with the multiple loopback interfaces and qfe interfaces?

```
# ifconfig -av
lo0:
flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
        inet 127.0.0.1 netmask ff000000
lo0:1:
flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
        zone vm0
        inet 127.0.0.1 netmask ff000000
lo0:2:
flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
        zone vm2
        inet 127.0.0.1 netmask ff000000
lo0:3:
flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
        zone vm1
        inet 127.0.0.1 netmask ff000000
ge0: flags=1000802<BROADCAST,MULTICAST,IPv4> mtu 1500 index 2
        inet 0.0.0.0 netmask 0
        ether 8:0:20:ee:19:4d
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 3
```

```
        inet 192.168.1.140 netmask ffffff00 broadcast
192.168.1.255
        ether 8:0:20:c4:ab:cd
qfe0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 4
        inet 0.0.0.0 netmask 0
        ether 8:0:20:9e:3c:2c
qfe0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 4
        zone vm0
        inet 192.168.1.141 netmask ffffff00 broadcast
192.168.1.255
qfe1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 5
        inet 0.0.0.0 netmask 0
        ether 8:0:20:9e:3c:2d
qfe1:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 5
        zone vm1
        inet 192.168.1.142 netmask ffffff00 broadcast
192.168.1.255
qfe2: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 6
        inet 0.0.0.0 netmask 0
        ether 8:0:20:9e:3c:2e
qfe3: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 7
        inet 0.0.0.0 netmask 0
        ether 8:0:20:9e:3c:2f
qfe3:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 7
        zone vm2
        inet 192.168.1.143 netmask ffffff00 broadcast
192.168.1.255
```

If you were to examine the processes running on a system with a ps -ef you might find many processes like these running. These are zone administration daemons running to manage the individual zones.

```
    root  1162     1    0 07:09:32 ?      0:00 zoneadmd -z vm1
    root   617     1    0 05:57:33 ?      0:00 zoneadmd -z vm0
    root  1242     1    0 07:09:51 ?      0:00 zoneadmd -z vm2
```

Many of the hardware and administrative tools that you find in the regular machine might not work and might give you mysterious error messages that hint around that your in a Solaris Container rather than on physical hardware. Here are some examples.

prtconf is not showing what we normally see.

```
# prtconf
System Configuration:  Sun Microsystems  sun4u
Memory size: 4096 Megabytes
System Peripherals (Software Nodes):

prtconf: devinfo facility not available
```

prtdiag seems to not work at all.

```
# prtdiag -v
prtdiag can only be run in the global zone
```

metstat does not work either

```
# metastat
metastat: vm0: Volume administration unavailable within non-
global zones.
```

These hints are nice but there are a few more defining things that we can do to really make sure that we are in a zone or not. The easiest way is by using the zonename command. The zonename command on the main system will always show that you are in the global zone.

```
# zonename
global
```

If we were inside the zone called vm0 we might see the following.

```
# zonename
vm0
```

Ok so lets say we are not in a zone but we are trying to determine the impact of shutting down a system for a reconfigure boot after we added new hardware. We can use the zoneadm command (which is a quite powerful command which I will have more information about later in this guide) to determine how many zones we have running on the system.

```
# zoneadm list -cv
  ID NAME             STATUS         PATH
   0 global           running        /
   1 vm1              running        /share/zones/vm1
   2 vm0              running        /share/zones/vm0
   3 vm2              running        /share/zones/vm2
```

Interesting. It looks like we have 3 zones that are all running not including the global zone. The path column shows the physical path of the filesystems that this VM is using. If you are on the global zone you can physically go to this path in the root filesystem to delete log files that might be filling disks or copy files into the zone. The status column shows what the status is on the Zone. Lets have a little fun and change the status with a few zoneadm commands.

```
# zoneadm -z vm1 halt
# zoneadm -z vm2 halt
```

The zoneadm command needs the -z to refer to the specific zone that we want to control. The halt option shuts down the zone. Notice the status has now changed.

```
# zoneadm list -cv
  ID NAME             STATUS         PATH
   0 global           running        /
   2 vm0              running        /share/zones/vm0
   - vm1              installed      /share/zones/vm1
   - vm2              installed      /share/zones/vm2
```

Ok we have had our fun. Our users probably want access to their systems again. Let's boot the zones. Notice the ID column has changed?

```
# zoneadm -z vm1 boot
# zoneadm -z vm2 boot

# zoneadm list -cv
  ID NAME             STATUS         PATH
   0 global           running        /
   2 vm0              running        /share/zones/vm0
   4 vm1              running        /share/zones/vm1
   5 vm2              running        /share/zones/vm2
```

Lets explore this a little more with a reboot command on vm2. Notice now the ID for VM2 is now 6. Quite useful for tracking zone restarts eh? Upon reboot of the global zone (which is always 0) this returns back to the sequential numbering we saw in previous examples.

```
# zoneadm -z vm2 reboot

# zoneadm list -cv
  ID NAME             STATUS         PATH
   0 global           running        /
   2 vm0              running        /share/zones/vm0
   4 vm1              running        /share/zones/vm1
   6 vm2              running        /share/zones/vm2
```

Lets go back for a moment and explore the filesystem element. Notice all of our zones seem to be installed in the directory /share/zones. This is not a standard but is where I installed the zones on my Enterprise 450. But lets take a peek at the directory structure we see underneath. Notice our root filesystem looks like what we might see on a normal machine? Instead of like many other VM products where the “disk” is a disk image this is a directory that we can navigate and physically interact with.

```
# cd /share/zones
# ls
vm0  vm1  vm2
# cd vm0
# ls
dev   root
# cd root
# ls
bin      etc       home      mnt       platform  sbin      tmp
var dev  export    lib       opt       proc      system    usr
```

zoneadm is capable of a number of different functions. From shutdown and boot of zones to the initial installation of the zone’s software. Running zoneadm without any subcommands gives us the following quick help. The full man page of zoneadm has a lot more detailed information.

```
# zoneadm
usage:  zoneadm help
        zoneadm [-z <zone>] list
        zoneadm -z <zone> <subcommand>

Subcommands:

help
boot [-s]
halt
ready
reboot
list [-cipv]
verify
install
uninstall [-F]
clone [-m method] zonename
move zonepath
detach
attach [-F]
```

Another highly useful command is the zlogin command which allows us to log into a "console like" interface from the global zone. Lets try logging into vm0. Notice we drop right to a root prompt.

```
# zlogin vm0
[Connected to zone 'vm0' pts/3]
Last login: Tue Mar 18 03:45:27 on pts/3
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
#
```

Lets get a bit more advanced here. Lets configure and install our own zone. While this is not what we would normally do on a production server, it is useful to see how the zonecfg command works. Here is a "recipe" for a very simple zone. The command line is interactive and changes as we go along in our zone configuration.

```
# zonecfg -z vm3
vm3: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:vm3> create
zonecfg:vm3> set zonepath=/share/zones/vm3
zonecfg:vm3> set autoboot=true
zonecfg:vm3> add inherit-pkg-dir
zonecfg:vm3:inherit-pkg-dir> set dir=/opt
zonecfg:vm3:inherit-pkg-dir> end
zonecfg:vm3> add net
zonecfg:vm3:net> set address=192.168.1.144
zonecfg:vm3:net> set physical=qfe3
zonecfg:vm3:net> end
zonecfg:vm3> add attr
zonecfg:vm3:net> set name=comment
zonecfg:vm3:net> set type=string
zonecfg:vm3:net> set value="vm3"
zonecfg:vm3:net> end
zonecfg:vm3> verify
zonecfg:vm3> commit
zonecfg:vm3> exit
```

After we have configured our zone we can use the zoneadm command to install the software to the zone. After a few minutes of copying files we can use the zoneadm command to boot our new zone.

Zonecfg can also be used to look at the configuration of a zone. Here we are looking at the configuration of our zone vm0 using the info command within zonecfg. Once we have seen the information that interested us we can use the exit command to leave the configuration menu. Keep in mind when you are in the configuration menu you have the ability to make changes so use caution in the zonecfg menu's.

```
# zonecfg -z vm0
zonecfg:vm0> info
zonename: vm0
zonepath: /share/zones/vm0
autoboot: true
pool:
limitpriv:
inherit-pkg-dir:
        dir: /lib
inherit-pkg-dir:
        dir: /platform
inherit-pkg-dir:
        dir: /sbin
inherit-pkg-dir:
        dir: /usr
inherit-pkg-dir:
        dir: /opt
net:
        address: 192.168.1.141
        physical: qfe0
attr:
        name: comment
        type: string
        value: vm0
zonecfg:vm0> exit
```

Here are some resources for further reading on the Sun website. While this guide gives some basic information its always good to read the full documentation.

Solaris Containers-Resource Management and Solaris Zones
http://dlc.sun.com/pdf/817-1592/817-1592.pdf

System Administration Guide: Advanced Administration
http://dlc.sun.com/pdf/817-0403/817-0403.pdf

Questions or comments? robert.b.chase@gmail.com