```
-----------------------------------------------------------------------
   I m p o r t a n t   N o t e  -  P l e a s e   R e a d   T h i s
-----------------------------------------------------------------------

  This article was written on the very day that the N1 Grid Container
  or Zones feature was released to the world in the build 51 beta of
  Solaris 10.  It was written with what I knew back then.  On day one.

  Everything here is still accurate but far better articles have been
  written on the topic since this one.  This article was merely the
  first.  It would be irresponsible of me to not point out to you, my
  kind reader, that you would do very well to read the documentation
  available at docs.sun.com and also PLEASE SEE THE FINE ARTICLE that
  is written by Brendan Gregg :


  Solaris 10 Zones document written by Brendan Gregg


  Thank you,

                Dennis Clarke
                Tue Jul  5 12:41:40 EDT 2005
```

Every now and again an entirely innovative approach to computer technology appears on the market. If you are anything like me then you have probably been working with computer technology for as long as you can remember. This includes the original PONG and probably 6502 assembly language programming on a Commodore PET with 8K of RAM. You have owned at least one Hewlett Packard calculator that used RPN and a Timex Sinclair Z80 computer. You could speak four different programming languages by the age of seventeen. I have this amazing attraction to really innovative technology. Things that actually capture my attention not because they are powerful but because they are entirely innovative in the way that they work. OKay, I admit that PONG was not slick but it was at least captivating at the time.

Solaris 10 is not only powerful but it is very innovative.

Solaris 10 can open your mind to entirely new solutions in much the same way that a ten megaton bomb can clear a room. Yes, it IS that slick.

Solaris has this wonderful new feature called "Zones". In addition to the many features that Solaris usually ships with we now have a new and very powerful feature that allows the administrator to create a virtual server within the server. Your server is no longer just a collection of user accounts and applications but it is now a collection of virtual servers within a global zone server. BSD Jail you say? A chroot environment? The comparison can be made to these older tools in much the same way that you can compare an aardvark to an elephant. They both walk and they both have four legs but one of them will carry you and a ton of cargo, the other, well, it just happens to have one vowel.

I currently have a user account setup for a developer who can not tell that he is in a virtual machine when he logs on. He is alone in his own server because I put him in his own zone. By himself. With an entire server to work with. For himself and his applications.

```
 He says : "Very cool"
   I say : "Very cool indeed, welcome to your own little world."
```

How to setup your very own zone!

The first step here is to get a piece of hardware to test with. You can use an Intel or AMD Opteron unit or a UltraSparc server. For my testing and playing I choose one of many Netra T1 UltraSparc units that I have in a rack. I download the CDROM ISO files from Sun. I then use lofiadm and a few other steps to create a jumpstart server for network booting. Simply put, I did the entire process remotely, from home, with a laptop and a modem. Like I said, Solaris is really slick.

Most admins that work with Solaris are very aware of how to boot a server from across the net and perform an install. That is what I did.

After the initial install of Solaris 10 build 51 I had the following config here at blastwave.org :

```
bash-2.05b# uname -a
SunOS zoner 5.10 s10_51 sun4u sparc SUNW,UltraSPARC-IIi-cEngine

bash-2.05b# prtconf -v | grep Memory
Memory size: 320 Megabytes
```

The file system layout looks like so :

```
bash-2.05b# df -ak
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0     371137   74357  259667    23%    /
/devices                   0       0       0     0%    /devices
/dev/dsk/c0t0d0s6    3009594 1708386 1241017    58%    /usr
proc                       0       0       0     0%    /proc
mnttab                     0       0       0     0%    /etc/mnttab
fd                         0       0       0     0%    /dev/fd
/dev/dsk/c0t0d0s1     740495   73752  607504    11%    /var
```

```
swap                  1262208      48 1262160     1%    /var/run
swap                  1262160       0 1262160     0%    /tmp
/dev/dsk/c0t0d0s5     1375228     693 1319526     1%    /opt
/dev/dsk/c0t0d0s7     2055705      30 1994004     1%    /export/home
-hosts                      0       0       0     0%    /net
auto_home                   0       0       0     0%    /home
zoner:vold(pid489)          0       0       0     0%    /vol
```

That would be because I chose an old fashioned way of doing things and I split up my basic filesystems across the primary boot disk.

```
bash-2.05b# isainfo -v
64-bit sparcv9 applications
32-bit sparc applications

$ ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.35.200 netmask ffffff00 broadcast 192.168.35.255
hme1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.35.201 netmask ffffff00 broadcast 192.168.35.255

bash-2.05b# psrinfo -v
Status of virtual processor 0 as of: 02/26/2004 10:30:42
  on-line since 02/25/2004 23:13:02.
  The sparcv9 processor operates at 360 MHz,
        and has a sparcv9 floating point processor.
```

Notice anything new there? It says "virtual processor". Virtual? Seems pretty real to me. Toto, I don't think we're in Kansas anymore.

No indeed. We are, in fact, in a the "global zone" of Solaris 10. The zoneadm tool confirms this :

```
bash-2.05b# zoneadm list -vc
  ID NAME             STATUS         PATH
   0 global           running        /
```

It is here in the new global zone that we will create our other zones for applications and for users. If we need to isolate an application or a user group from the rest of the world then we simply create a zone for them and then let them run. Simple concept but how do we do it?

The first thing that I do is create a filesystem area for the new zone to reside in. I also mount it under a mount point named /zone/1 and I ensure that only the root user has access to it thus :

```
$ ls -lap /zone
total 8
drwxr-xr-x   3 root     other        512 Feb 26 12:42 ./
drwxr-xr-x  22 root     root         512 Feb 26 12:42 ../
drwx------   5 root     root         512 Feb 26 13:27 1/
```

We use zonecfg to create a new zone. Do this from the global zone and as the root user.

```
bash-2.05b# zonecfg -z zone1
zone1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/zone/1
zonecfg:zone1> set autoboot=true
zonecfg:zone1> add net
zonecfg:zone1:net> set address=192.168.35.210
zonecfg:zone1:net> set physical=hme1
zonecfg:zone1:net> end
zonecfg:zone1> info
zonepath: /zone/1
autoboot: true
pool:
inherit-pkg-dir:
        dir: /lib
inherit-pkg-dir:
        dir: /platform
inherit-pkg-dir:
        dir: /sbin
inherit-pkg-dir:
        dir: /usr
net:
        address: 192.168.35.210
        physical: hme1
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> ^D
```

Simple really. The zonecfg tool is interactive and I specified that I want to "create" a zone. The filesystem that I created is the new "zonepath" and I want this new virtual server to boot along with the global zone when the "real" server boots. Who can tell what is "real" and what isn't? It won't matter anymore. I also set the ip address for the zone as well as the interface to bind to. Finally I asked for zonecfg to show me what I just did via the simple "info" command. I then used "verify" and "commit" to ensure that the config is complete. That is all. Nothing fancy.

I then used zonecfg and zoneadm to verify that in fact what I had just done was in fact, er, well, done. Really I just like playing with new technology and so will you!

```
bash-2.05b# zonecfg -z zone1 info
zonepath: /zone/1
autoboot: true
pool:
inherit-pkg-dir:
        dir: /lib
inherit-pkg-dir:
        dir: /platform
inherit-pkg-dir:
        dir: /sbin
inherit-pkg-dir:
        dir: /usr
net:
        address: 192.168.35.210
        physical: hme1

bash-2.05b# zoneadm list -vc
  ID NAME             STATUS         PATH
   0 global           running        /
   - zone1            configured     /zone/1
```

The next step to perform is to "install" the zone.

```
bash-2.05b# zoneadm -z zone1 install
Preparing to install zone <zone1>.
Creating list of files to copy from the global zone.
Copying <2521> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <808> packages on the zone.
Initializing package <7> of <808>: percent complete: 0%
.
.  < this goes on for some time >
.
Initialized <808> packages on zone.
Successfully initialized zone <zone1>.

bash-2.05b# df -ak /zone/1
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t1d0s0    1972734   76238 1797860     5%    /zone/1
```

Again I use zoneadm to see the results of my actions :

```
bash-2.05b# zoneadm list -vc
  ID NAME             STATUS         PATH
   0 global           running        /
   - zone1            installed      /zone/1
```

See that? The STATUS is now "installed".

Now lets boot that new virtual server that we created!

```
bash-2.05b# zoneadm -z zone1 boot
bash-2.05b# zoneadm list -vc
  ID NAME             STATUS         PATH
   0 global           running        /
   2 zone1            running        /zone/1
```

I now have a virtual server running? Really? Let’s ping it :

```
bash-2.05b# ping 192.168.35.210
192.168.35.210 is alive
```

For our further enjoyment let’s nmap port scan it from another server :

```
# nmap -sS -O -v -v -P0 -T Aggressive -n -oN /tmp/zone1.log zone1

Starting nmap 3.20 ( www.insecure.org/nmap/ ) at 2004-02-26 14:12 EST
Host 192.168.35.210 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.35.210 at 14:12
The SYN Stealth Scan took 443 seconds to scan 1611 ports.
Warning:  OS detection will be MUCH less reliable because we did not find at least 1

open and 1 closed TCP port
All 1611 scanned ports on 192.168.35.210 are: closed
Too many fingerprints match this host for me to give an accurate OS guess
TCP/IP fingerprint:
SInfo(V=3.20%P=sparc-sun-solaris2.8%D=2/26%Time=403E474B%O=-1%C=1)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=N)
PU(Resp=N)
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 551.944 seconds
```

As far as anyone can tell, there is a server out there at the required IP address but how do we actually use it?

We now need to login to the zone console via zlogin. When we do we will be presented with the opportunity to "setup" the new server as if we were doing an install only there really isn't much to install or setup for that matter. The hard work has been done for us :

I now will use zlogin to login to the zone1 console and I will specify

```
# zlogin -C -e\@ zone1
[Connected to zone 'zone1' console]
```

This is where we are presented with an install sequence that is familiar to all Solaris admins. After you answer the basic config questions for your new virtual server you will see that the virtual server boots :

```
[NOTICE: zone rebooting]

Version s10_51 64-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Hostname: zone1
The system is coming up.  Please wait.
starting rpc services: rpcbind done.
syslog service starting
                     SunOS Release 5.10 Ver
prtconf: devinfo facility not available
prtconf: devinfo facility not available
prtconf: cannot open /dev/openprom: No such file or directory
prtconf: cannot open /dev/openprom: No such file or directory
prtconf: devinfo facility not available
Creating new rsa public/private host key pair
Creating new dsa public/private host key pair
The system is ready.

zone1 console login:

zone1 console login: root
Password:

Feb 27 09:03:55 zone1 login: ROOT LOGIN /dev/console
Sun Microsystems Inc.   SunOS 5.10      s10_51  May 2004
#
```

There you have it! A new virtual server has been born. This new server is neatly wrapped inside the global zone. I don't have another way to describe it really. Perhaps it is "beside" or "outside". Does it matter? Not really. The new server has a hostname zone1 (for the sake of simplicity) but I could have made the main hostname jupiter and the new zone io or europa. The new server is reachable via the net. I run ps -ef and see the usual suspects in place and running :

```
# ps -ef
     UID   PID  PPID   C    STIME TTY         TIME CMD
    root  7530  7424   0 09:01:46 ?           0:00 /usr/sbin/inetd -s
  daemon  7451  7424   0 09:01:45 ?           0:00 /usr/lib/crypto/kcfd
    root  7516  7424   0 09:01:46 ?           0:00 /usr/lib/autofs/automountd
    root  7667  7653   0 09:04:18 console     0:00 ps -ef
    root  7424  7424   0 09:01:33 ?           0:00 zsched
    root  7521  7424   0 09:01:46 ?           0:00 /usr/sbin/cron
    root  7515  7424   0 09:01:46 ?           0:00 /usr/sbin/syslogd
    root  7653  7427   0 09:03:30 console     0:00 -sh
    root  7588  7424   0 09:02:49 ?           0:00 /usr/lib/im/htt -port 9010 -syslog -message_locale C
    root  7427  7424   0 09:01:44 ?           0:00 init
    root  7652  7427   0 09:03:30 ?           0:00 /usr/lib/saf/sac -t 300
    root  7526  7424   0 09:01:46 ?           0:00 /usr/sbin/nscd
    root  7656  7652   0 09:03:30 ?           0:00 /usr/lib/saf/ttymon
    root  7641  7424   0 09:02:52 ?           0:01 /usr/sfw/sbin/snmpd
    root  7568  7424   0 09:02:49 ?           0:00 /usr/lib/utmpd
   smmsp  7658  7424   0 09:03:49 ?           0:00 /usr/lib/sendmail -Ac -q15m
  daemon  7476  7424   0 09:01:45 ?           0:00 /usr/sbin/rpcbind
    root  7605  7588   0 09:02:50 ?           0:00 htt_server -port 9010 -syslog -message_locale C
    root  7636  7424   0 09:02:51 ?           0:00 /usr/dt/bin/dtlogin -daemon
    root  7657  7424   0 09:03:49 ?           0:00 /usr/lib/sendmail -bd -q15m
    root  7655  7424   0 09:03:30 ?           0:00 /usr/lib/ssh/sshd
#
```

The filesystems look a bit odd in that they are not actually associated with disk devices or metadevices :

```
# df -ak
Filesystem            kbytes    used   avail capacity  Mounted on
/                    1972734   76154 1797944     5%    /
/dev                 1972734   76154 1797944     5%    /dev
/lib                  371137   74367  259657    23%    /lib
/platform             371137   74367  259657    23%    /platform
/sbin                 371137   74367  259657    23%    /sbin
/usr                 3009594 1708386 1241017    58%    /usr
proc                       0       0       0     0%    /proc
```

```
mnttab                     0       0       0     0%    /etc/mnttab
auto_home                  0       0       0     0%    /home
-hosts                     0       0       0     0%    /net
swap                 1220032       0 1220032     0%    /tmp
swap                 1220064      32 1220032     1%    /var/run
fd                         0       0       0     0%    /dev/fd
#
```

The rest of the config of this virtual server is not surprising at all :

```
# uname -a
SunOS zone1 5.10 s10_51 sun4u sparc SUNW,UltraSPARC-IIi-cEngine
# psrinfo -v
Status of virtual processor 0 as of: 02/27/2004 09:08:06
  on-line since 02/26/2004 11:48:33.
  The sparcv9 processor operates at 360 MHz,
        and has a sparcv9 floating point processor.
# isainfo -v
64-bit sparcv9 applications
32-bit sparc applications
# prtconf -v
System Configuration:  Sun Microsystems  sun4u
Memory size: 320 Megabytes
System Peripherals (Software Nodes):

prtconf: devinfo facility not available


# ifconfig -a
lo0:1: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme1:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
        inet 192.168.35.210 netmask ffffff00 broadcast 192.168.35.255
```

Special Note : The virtual server in the zone hides the system details. The hardware on which the zone is running is not exposed to the applications or users. This explains why prtconf reveals nothing about the underlying hardware architecture. The isainfo tool clearly tells us that we are able to run 64-bit sparcv9 and 32-bit sparc applications. As far as the user or application is concerned we have a V880 running. Or a 280R. Or a E15K! It really doesn't matter because we are in a virtual server that could be within any class of hardware.

Well, there we have it. A virtual server. A new zone is born and running. I now create a user account or two and then use the escape character from the zlogin command to exit the console :

```
# exit

zone1 console login:

zone1 console login: @.
[Connection to zone 'zone1' console closed]
#

#
```

Now we are back in the real world! Or at least we are in the global zone. I log out of the server entirely and nmap port scan the virtual server again:

```
# nmap -sS -O -v -v -P0 -T Aggressive -n -oN /tmp/zone1.log zone1

Starting nmap 3.20 ( www.insecure.org/nmap/ ) at 2004-02-27 09:34 EST
Host 192.168.35.210 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.35.210 at 9:34
Adding open port 37/tcp
Adding open port 22/tcp
Adding open port 513/tcp
Adding open port 515/tcp
Adding open port 514/tcp
Adding open port 7100/tcp
Adding open port 7/tcp
Adding open port 21/tcp
Adding open port 587/tcp
Adding open port 19/tcp
Adding open port 544/tcp
Adding open port 9/tcp
Adding open port 2105/tcp
Adding open port 111/tcp
Adding open port 13/tcp
Adding open port 79/tcp
Adding open port 540/tcp
Adding open port 25/tcp
Adding open port 543/tcp
Adding open port 23/tcp
Adding open port 512/tcp
The SYN Stealth Scan took 484 seconds to scan 1611 ports.
For OSScan assuming that port 7 is open and port 1 is closed and neither are firewalled
Interesting ports on 192.168.35.210:
(The 1590 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
25/tcp     open        smtp
```

```
37/tcp     open        time
79/tcp     open        finger
111/tcp    open        sunrpc
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
540/tcp    open        uucp
543/tcp    open        klogin
544/tcp    open        kshell
587/tcp    open        submission
2105/tcp   open        eklogin
7100/tcp   open        font-service
Remote operating system guess: Solaris 9 Beta through Release on SPARC
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=F1B6%IPID=I%TS=100HZ)
T1(Resp=Y%DF=Y%W=C0B7%ACK=S++%Flags=AS%Ops=NNTMNW)
T2(Resp=N)
T3(Resp=N)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=N)
PU(Resp=N)

Uptime 0.914 days (since Thu Feb 26 11:47:32 2004)
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=61878 (Worthy challenge)
TCP ISN Seq. Numbers: 7D44D7F9 7D46507E 7D49EF7E 7D4E40EC 7D5145C4 7D5426FD
IPID Sequence Generation: Incremental

Nmap run completed -- 1 IP address (1 host up) scanned in 517.027 seconds
```

Essentially all of the usual network services are running on that virtual server. Finally I login to it via ssh :

```
$ ssh -2 -4 -e\^ -l dclarke zone1
The authenticity of host 'zone1 (192.168.35.210)' can't be established.
RSA key fingerprint is f0:0b:a1:de:ad:be:ef:01:a4:21:53:8d:ae:de:00:00.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'zone1,192.168.35.210' (RSA) to the list of known hosts.
Password:
Last login: Fri Feb 27 09:53:53 2004 from blaster
Sun Microsystems Inc.   SunOS 5.10      s10_51  May 2004
$
```

What more can I say? I have setup a user account for a developer on the virtual server and he can login. There I can grant access to filesystems and other authorities as required. He can do what ever he pleases and I don't have to worry about resources being tied up or security being breached. I can give him root access and not be concerned at all. Each zone has its own set of objects including processes, network interfaces, System V IPC objects and a unique root file system. Processes in one zone cannot access or control objects in other zones unless a common access point is shared from the global zone. All I need is a server with the hardware resources that can then be split up across zones. If I want an eight processor V880 to be used more effectively then I will manage the zones and allocate the CPU horsepower as needed.

But that is another topic, that would be under Resource Management. Which has been overhauled and made into a powerful tool within Solaris. The new Solaris 10 release will be feature rich and massively powerful in its ability to swallow up tasks. I look forward to playing with it because it really makes complex situations into a fun process.


```
Dennis Clarke             Admin and Director for blastwave.org
dclarke @ blastwave.org   Home of CSW - Community Software for Solaris
```


Further Reading : BigAdmin System Administration Portal

©2002-2005 blastwave.org
Built by Dennis Clarke
Admin Owner and Director
dclarke@blastwave.org

Sun™ Logo Published with Permission from Sun Microsystems, Inc.