

OpenSolaris™ Zones

Presentation given at ApacheCon US 2006

Narayana Janga
Shivani Khosa

Sun Microsystems Inc.

OpenSolaris Zones

- ◆ Agenda
 - ◆ What are OpenSolaris Zones?
 - ◆ Features
 - ◆ Runtime Model
 - ◆ Configuration/Administration
 - ◆ Service Virtualization

OpenSolaris Zones

- ◆ Agenda (2)
 - ◆ Fair Share Scheduling (FSS)
 - ◆ Zones, Projects and Pools
 - ◆ Zones in Action (Code Examples)
 - ◆ Possible Uses
 - ◆ References
 - ◆ Q & A

OpenSolaris Zones

- ◆ What are zones?
 - ◆ A virtualized operating system environment
 - ◆ Each zone has its own characteristics e.g. zonename, IP addresses, hostname, naming services, root and non-root users
 - ◆ Isolated application environments within a single OS instance

OpenSolaris Zones

Notes for previous slide:

Virtualization dramatically reduces the cost of deploying and maintaining multiple machines and applications. The most common need for virtualization is application consolidation. Many of the larger applications have become so complex that they become a system in themselves - and often they don't play nicely with other applications on the box. So "one app per machine" has become the common paradigm. The second most common need is security, either for your application administrators or your developers. Other reasons certainly exist (rapid test environment deployment, distributed system simulation, etc), but these are the two primary ones.

So what does virtualization buy us? It's all about reducing costs, but there are really two types of cost associated with running a system:

Hardware costs - This includes the cost of the machine, but also the costs associated with running that machine (power, A/C).

Software management costs - This includes the cost of deploying new machines, and upgrading/patching software, and observing software behavior.

As we'll see, different virtualization strategies provide different qualities of the above savings.

http://www.genunix.org/wiki/index.php/Solaris10_Tech_FAQ

OpenSolaris Zones

- ◆ What are zones? (2)
 - ◆ By default, the OS runs in a “global” zone
 - ◆ The administrator can virtualize the execution environment by defining one or more “non-global” zones

OpenSolaris Zones

◆ Features

◆ Security

- ◆ Network services can be run limiting the damage possible in the event of security violation.

◆ Isolation

- ◆ Multiple applications can be run on the same machine even when they operate in different trust domains.

OpenSolaris Zones

- ◆ Features (2)

- ◆ Virtualization

- ◆ Zones present a virtualized environment to applications removing the physical details of the hardware from view.

- ◆ Granularity

- ◆ Since zones are implemented in software, they aren't limited to granularity defined by hardware boundaries. Instead, zones offer sub-CPU granularity.

OpenSolaris Zones

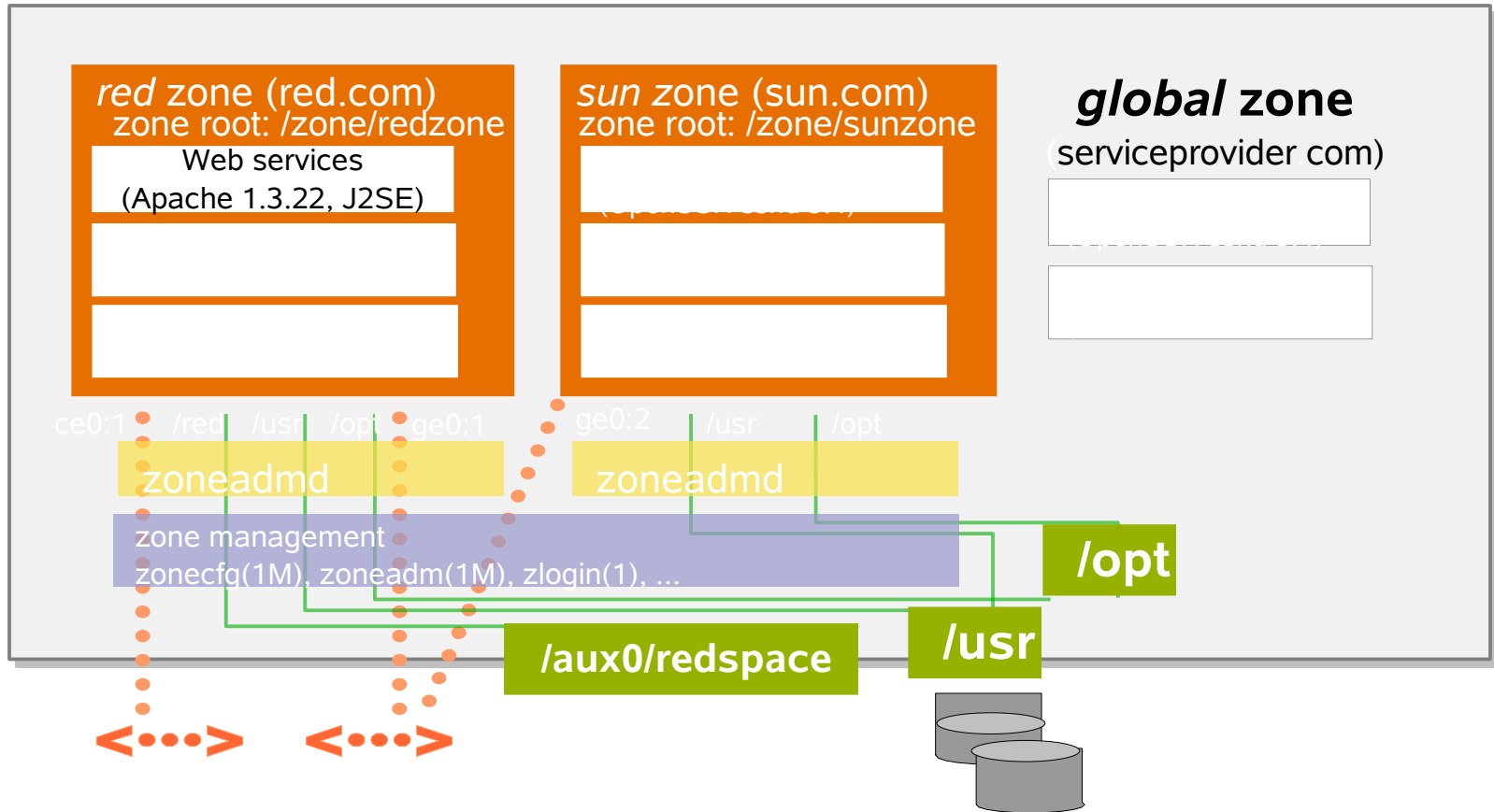
- ◆ Features (3)

- ◆ Transparency

- ◆ The environment presented to applications in zones is nearly identical to the standard SolarisTM application environment.

OpenSolaris Zones

◆ Runtime Model - Zones Architecture



OpenSolaris Zones

◆ Runtime Model - Zones Architecture

Notes: for previous slide:

This diagram demonstrates that different versions of the same application may be run without negative consequences in different zones.

Basic process isolation is also demonstrated.

Each zone is given access to at least one logical network interface. Applications running in distinct zones cannot observe the network traffic of the other, even though their respective streams of traffic travel through the same physical interface.

Each zone is provided a portion of the file system.

This figure also explains two important abstractions, namely, Virtual Platform and application environment.

The Virtual Platform is the set of services and resources that allow zones to function e.g. Network interfaces, devices, zoneadm, zone console.

The application environment is the virtualized runtime state of the zone.

OpenSolaris Zones

- ◆ Runtime Model (2)
 - ◆ To manage the virtual platform and the application environment, two new processes are used by the zone runtime:
 - ◆ zoneadmd
 - ◆ Manages resources associated with the zone
 - ◆ zsched
 - ◆ Tracks kernel resources associated with the zone

OpenSolaris Zones

◆ Runtime Model (2)

Notes on previous slide:

•To manage the virtual platform and the application environment, two new processes are used by the zone runtime. zoneadm manages most of the resources associated with the zone and zsched is a system process (like sched) which is used to track kernel resources associated with the zone.

•1) zoneadm(1M)

•zoneadm(1M) is the primary process responsible for managing the zone's virtual platform. It is also responsible for setup and teardown of the application environment. There is one zoneadm running for each active (ready, running, shutting down) zone on the system. zoneadm is responsible for consulting the zone configuration and then setting up the zone as directed. This entails:

- Calling the zone_create(2) system call; this allocates a zone ID and starts zsched
- Setting zone-wide resource controls.
- Registering the zone with devfsadm(1M).
- Plumbing virtual network interfaces.
- Mounting loopback and conventional file systems.
- Instantiating and initializing the zone console device.

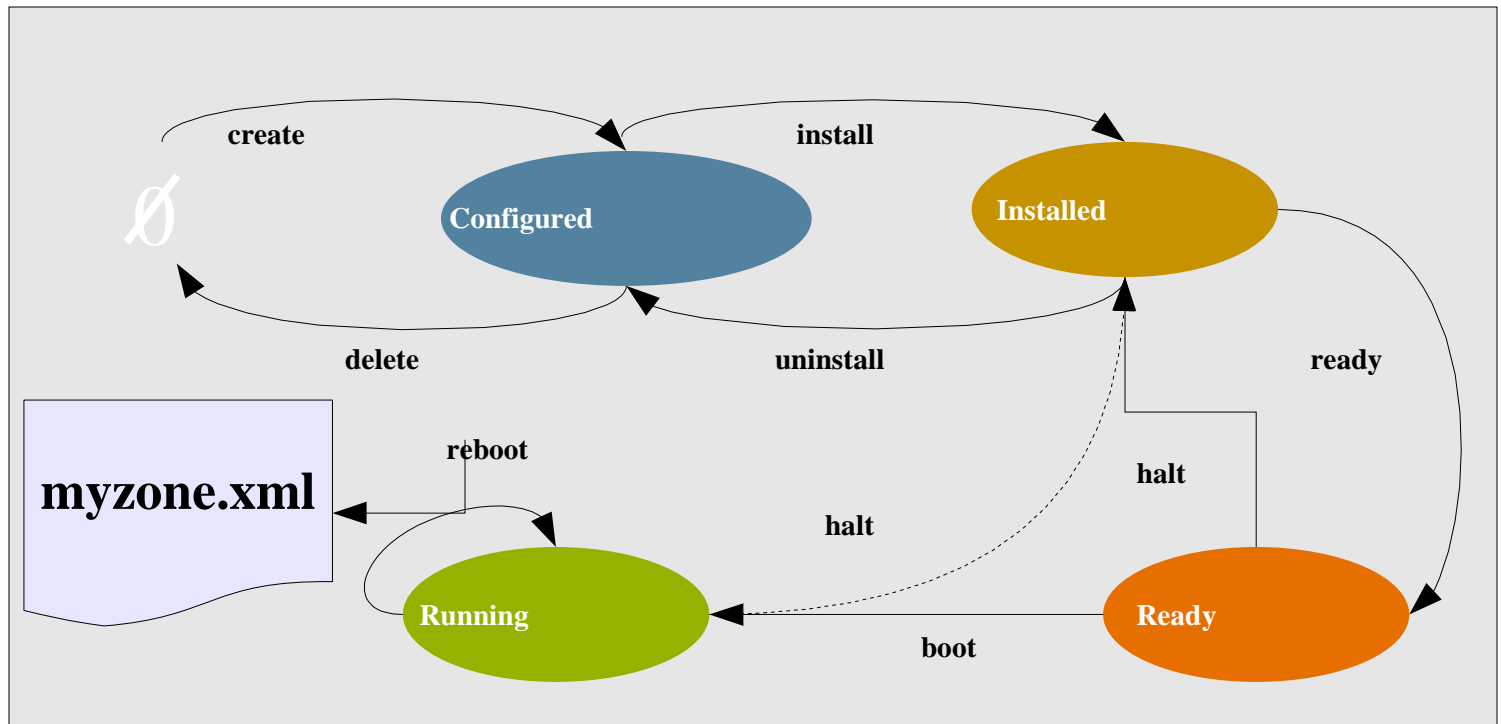
•zoneadm is automatically started by zoneadm(1M) if not already running, and can be contacted by userland applications (such as zoneadm in the global zone), and the kernel (as part of uadmin(2) calls from the managed zone).

•2) zsched

•Every active (ready through shutting down) zone has an associated kernel process, zsched. Kernel threads doing work on behalf of the zone are owned by zsched. It exists largely to enable the zones subsystem to keep track of per-zone kernel threads

OpenSolaris Zones

◆ Runtime Model (3) – Zones State Diagram



OpenSolaris Zones

◆ Runtime Model (3) – Zones State Diagram

•Notes on previous slide:

•Non-global zones go through the following state transitions:

- ZONE_STATE_CONFIGURED
- ZONE_STATE_INSTALLED
- ZONE_STATE_READY
- ZONE_STATE_RUNNING

•

•Some other zone states are:

- ZONE_STATE_INCOMPLETE
- ZONE_STATE_DOWN

•

•New zone state being introduced is:

- ZONE_STATE_MOUNTED

OpenSolaris Zones

- ◆ Configuration/Administration
 - ◆ Zone administration uses mainly the following commands:
 - ◆ `zonectfg(1M)`
 - ◆ `zoneadm(1M)`
 - ◆ `zlogin(1M)`

OpenSolaris Zones

- ◆ Configuration/Administration (2)
 - ◆ `zonectfg` (1M)
 - ◆ Creates Zone configuration
 - ◆ Configures Zones (adds resources and properties)
 - ◆ Stores the configuration in a private XML file under `/etc/zones`
 - ◆ `zoneadm` (1M)
 - ◆ Performs administrative steps for zones such as list, install, (re)boot, halt, etc.

OpenSolaris Zones

- ◆ Configuration/Administration (3)
 - ◆ `zlogin` (1M)
 - ◆ Allows user to log in to the zone
 - ◆ Performs administration and maintenance

OpenSolaris Zones

- ◆ Configuration/Administration (4)
 - ◆ Global Scope Properties
 - ◆ zonepath
 - ◆ Path in global zone to root directory under which zone will be installed
 - ◆ zonename
 - ◆ Name of the zone being created
 - ◆ autoboot
 - ◆ To boot or not when global zone boots
 - ◆ pool
 - ◆ Resource pools to which zones should be bound

OpenSolaris Zones

- ◆ Configuration/Administration (4)
 - ◆ Global Scope Properties
 - ◆ `bootargs`
 - ◆ To specify boot arguments for zones
 - ◆ `limitpriv`
 - ◆ To provide zones with certain privileges so that certain privileged applications can be run inside them

OpenSolaris Zones

- ◆ Configuration/Administration (5)
 - ◆ Resources
 - ◆ fs
 - ◆ file system
 - ◆ inherit-pkg-dir
 - ◆ Directory which should have its associated packages inherited from the global zone.
 - ◆ net
 - ◆ Network device

OpenSolaris Zones

- ◆ Configuration/Administration (5)
 - ◆ Resources
 - ◆ device
 - ◆ Devices
 - ◆ dataset
 - ◆ For exporting ZFS datasets into non-global zones

OpenSolaris Zones

◆ Configuration/Administration (5)

Notes on previous slide:

Zone and ZFS integration - http://blogs.sun.com/dp/entry/zfs_and_zones_z_s

In ZFS, storage is pooled, and you can create one or many file systems from the pool. Each file system acts as a container for those below it: file system properties set on a parent are inherited to children. Even better, you can allocate one or many of these containers to a zone-- and then the zone can manage that container for itself, creating new child file systems. This is an incredibly powerful form of delegated administration and I think it's a great example of how our suite of technologies can be converged.

Here's an example:

```
•trolls # zpool create mypool c0d0s3
•trolls # zpool list
•NAME                SIZE      USED    AVAIL    CAP  HEALTH    ALTROOT
•mypool                95.5M    32.5K   95.5M    0%   ONLINE    -
•trolls # zfs create mypool/myzone-data
•trolls # zfs set compression=on mypool/myzone-data
•trolls # zfs set quota=30m mypool/myzone-data
•So at this point, we've created a storage pool, enabled compression, and made sure that myzone-data will never grow larger than 30MB in size. Now, we'll add that dataset to a zone on the system:
•trolls # zonecfg -z myzone
•zonecfg:myzone> add dataset
•zonecfg:myzone:dataset> set name=mypool/myzone-data
•zonecfg:myzone:dataset> end
•zonecfg:myzone> ^D
•trolls #
•trolls # zoneadm -z myzone boot
```

OpenSolaris Zones

- ◆ Configuration/Administration (6)
 - ◆ Resources (2)
 - ◆ rctl
 - ◆ resource controls
 - ◆ attr
 - ◆ generic attributes

OpenSolaris Zones

- ◆ Service Virtualization
 - ◆ Processes
 - ◆ File Systems
 - ◆ Networking
 - ◆ Identity
 - ◆ CPU Visibility
 - ◆ Packaging
 - ◆ Devices
 - ◆ Resource Management

OpenSolaris Zones

- ◆ Service Virtualization - Processes
 - ◆ Local Zone
 - ◆ Only processes in the same zone can be seen or affected
 - ◆ Certain system calls are not permitted or have restricted scope
 - ◆ `proc(4)` has been virtualized to show only processes in the same zone
 - ◆ Global Zone
 - ◆ All processes can be seen but control is privileged

OpenSolaris Zones

- ◆ Service Virtualization - File Systems
 - ◆ Each zone is allocated its own root file system and cannot see that of others
 - ◆ Unlike with `chroot(2)`, processes cannot escape out of a zone
 - ◆ File systems like `/usr` can be inherited in a read-only manner

OpenSolaris Zones

- ◆ Service Virtualization - Networking
 - ◆ Single TCP/IP stack for the system so zones are shielded from the configuration details for devices, routing etc.
 - ◆ Each zone can be assigned IPv4/IPv6 addresses and has its own port space
 - ◆ Applications can bind to INADDR_ANY and will only get traffic for that zone
 - ◆ Zones cannot see the traffic of others

OpenSolaris Zones

- ◆ Service Virtualization - Networking (2)
 - ◆ Packets coming from a zone have a source address belonging to that zone
 - ◆ A zone can only send packets on an interface on which it has an address
 - ◆ A zone can only use a router if it's directly reachable from the zone
 - ◆ The router has to be in the same IP subnet as the zone

OpenSolaris Zones

- ◆ Service Virtualization - Networking (3)
 - ◆ Zones can't change network configuration or routing table
 - ◆ Can't see other zones configuration either
 - ◆ `/dev/ip` is not present in the zone
 - ◆ SNMP agents must open `/dev/arp` instead
 - ◆ Multiple zones can share a broadcast address
 - ◆ Multiple zones can join the same multi-cast group

OpenSolaris Zones

- ◆ Service Virtualization - Networking (4)
 - ◆ Zones can't have dedicated physical interfaces assigned to them
 - ◆ IPFilter doesn't work between zones
 - ◆ It can be configured from the global zone to filter traffic to/from zones
 - ◆ No DHCP for Zones IP addresses

OpenSolaris Zones

- ◆ Service Virtualization - Networking (4)
 - ◆ No Dynamic Routing
 - ◆ But the non-global zone can take advantage of dynamic routing that the global zone is partaking in
 - ◆ IP Multipathing

OpenSolaris Zones

- ◆ Service Virtualization - Identity
 - ◆ Each zone controls its node name, time zone, naming services like LDAP and NIS, etc.
 - ◆ `sysidtool` can set this up
 - ◆ Separate `/etc/passwd` files mean that root can be delegated to the zone
 - ◆ User IDs may map to different names when domains differ

OpenSolaris Zones

- ◆ Service Virtualization - CPU Visibility
 - ◆ By default, all zones see all CPUs
 - ◆ Restricted view is enabled automatically when resource pools are enabled

OpenSolaris Zones

- ◆ Service Virtualization - Packaging
 - ◆ Zones can add their own packages
 - ◆ Patches can be made to those packages
 - ◆ OS Patches
 - ◆ Applied in global zone
 - ◆ Then, in the non-global zones one-by-one

OpenSolaris Zones

- ◆ Service Virtualization - Packaging (2)
 - ◆ **SUNW_PKG_ALLZONES**: Package should be kept consistent between the global zone and all non-global zones
 - ◆ **SUNW_PKG_HOLLOW**: Causes package name to appear in Non-Global Zones (NGZ) for dependency purposes but the contents are not installed
 - ◆ Typically used for kernel modules

OpenSolaris Zones

- ◆ Service Virtualization - Devices
 - ◆ Each Zone has its own devices
 - ◆ Zones see a subset of “safe” pseudo devices in their `/dev` directory
 - ◆ `/dev` exists in non-global zones, `/devices` does not
 - ◆ Devices like `random`, `console`, `null` are safe
 - ◆ But others like `/dev/ip` are not

OpenSolaris Zones

- ◆ Service Virtualization - Devices (2)
 - ◆ Zones can modify the permissions of their devices but cannot use `mknod(2)`
 - ◆ Physical device files like those for raw disks can be put in a zone with caution
 - ◆ Devices may be shared among zones, but careful study of security concerns is needed before doing this

OpenSolaris Zones

- ◆ Service Virtualization - Resources
 - ◆ Solaris Resource Management (SRM)
 - ◆ Constraint mechanisms with resource bound specifications
 - ◆ Scheduling mechanisms for CPU resources (FSS)
 - ◆ Partition mechanisms
 - ◆ With zone and resource pool bound resource usage
 - ◆ Virtualization of project namespace

OpenSolaris Zones

- ◆ Service Virtualization - Resources (2)
 - ◆ Solaris Resource Management (SRM)
 - ◆ Without SRM, Solaris OS gives all activity on the system equal access to resources.
 - ◆ Resource limits can be set on a zone as well
 - ◆ `zone.max-lwps` limits the number of created LWPs
 - ◆ `zone.cpu-shares` specifies the number of FSS shares it is entitled to from its assigned resource pool

OpenSolaris Zones

◆ Service Virtualization - Resources (2)

Notes on previous slide:

You set the resource caps limits to zones in the following way:

Unlike CPU shares, CPU caps let you naturally limit CPU usage for a project or for a zone. So, a cap of "150" means "150%" or "never use more than 1 and 1/2 CPUs".

```
•# zonecfg -z myzone
•zonecfg:myzone> add rctl
•zonecfg:myzone:rctl> set name=zone.cpu-cap
•zonecfg:myzone:rctl> add value
(priv=privileged,limit=150,action=deny)
•zonecfg:myzone:rctl> end
•zonecfg:myzone> exit
```

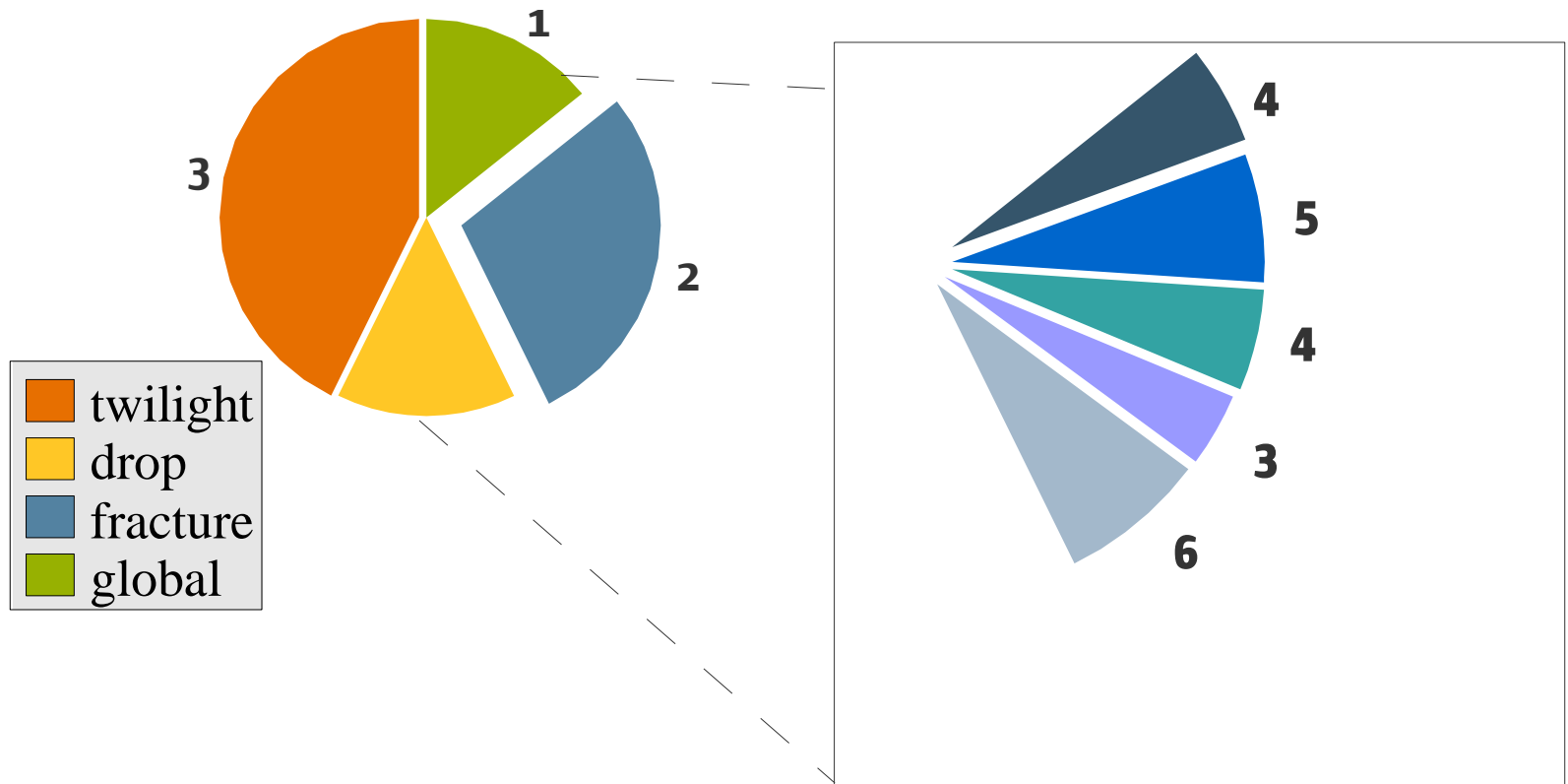
•

The following site shows information about resource caps:

<http://www.opensolaris.org/os/project/rm/rctls/cpu-caps/>

OpenSolaris Zones

◆ Fair Share Scheduling (FSS)



OpenSolaris Zones

◆ Fair Share Scheduling (FSS)

Notes on previous slide:

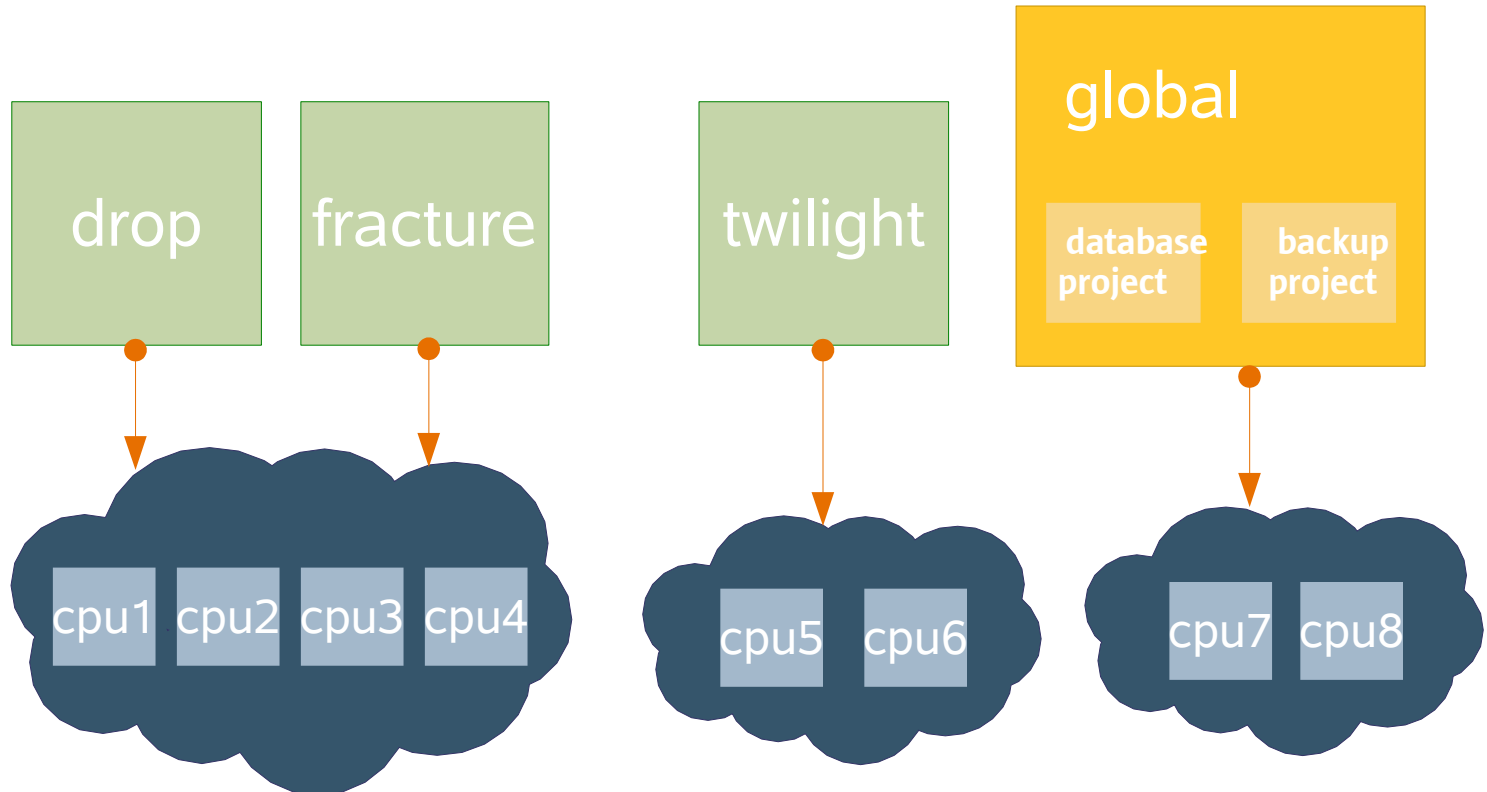
The *Fair Share Scheduler* (FSS) controls allocation of CPU resources using *CPU shares*. The importance of a workload is expressed by the number of shares the system administrator allocates to the project representing the workload. The Fair Share Scheduler ensures that CPU resources are distributed among active projects based on the number of shares assigned to each project.

A CPU share defines a relative entitlement of the CPU resources available to a project on the system. It is important to note that CPU shares are *not* the same as CPU percentages. Shares define the *relative importance* of projects with respect to other projects.

The Fair Share Scheduler calculates the proportion of CPU resources allocated to a project by dividing the shares for the project by the total number of shares of active projects. An *active project* is a project with at least one process using CPU resources.

OpenSolaris Zones

◆ Zones, Projects and Resource Pools



OpenSolaris Zones

◆ Zones, Projects and Resource Pools

•Notes on previous slide:

Fair Share Scheduler and Zones

To prevent a local zone from monopolizing the system, the global zone administrator can set zone-wide resource controls. The `zone.cpu-shares` resource control limits the amount of CPU resources a zone is entitled to in the same way that the Fair Share Scheduler does this for projects. A zone with a higher number of `zone.cpu-shares` is allowed to use more CPU resources than a zone with a low number of shares. The `zone.cpu-shares` resource control is configured using the `zonectg(1M)` command. Note that this requires the Fair Share Scheduler to be the default scheduler for the system, or that the zones be bound to a pool with a processor set with FSS as the scheduler.

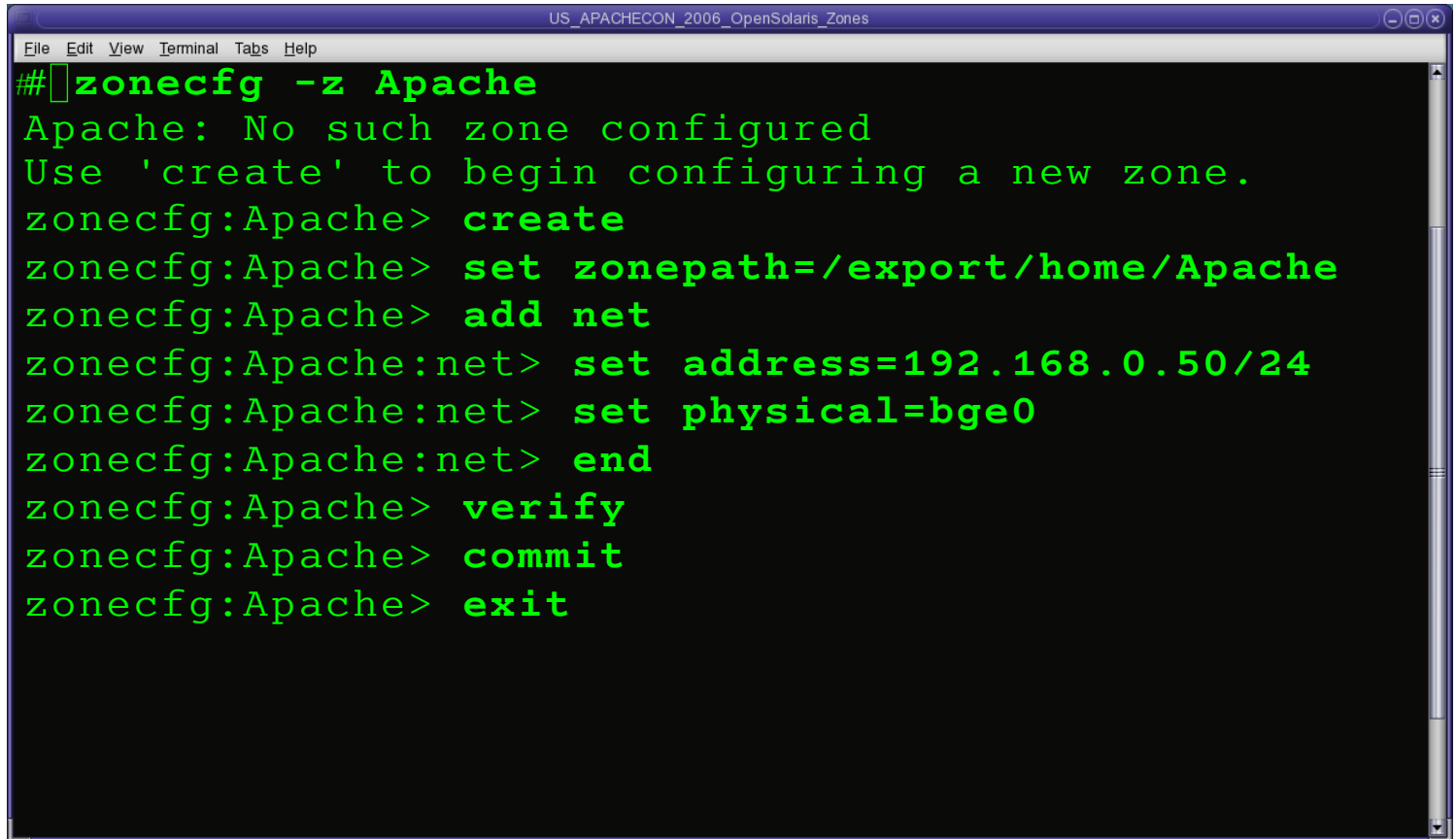
Combined with the regular Fair Share Scheduler inside a zone, this leads to a two-level distribution of CPU resources. First, the `zone.cpu-shares` configured by the global zone administrator determine the amount of CPU resources to which a zone is entitled. The amount of CPU resources available to the zone is then further distributed across the active projects in the zone according to the `project.cpu-shares` defined by the local zone administrator.

Resource Controls

All standard resource controls are available inside the local zone and can be used by the local zone administrator to perform resource management in the zone. The global zone administrator can limit the number of lightweight processes (LWPs) created by a zone by setting the `zone.max-lwps` resource control on a zone.

OpenSolaris Zones

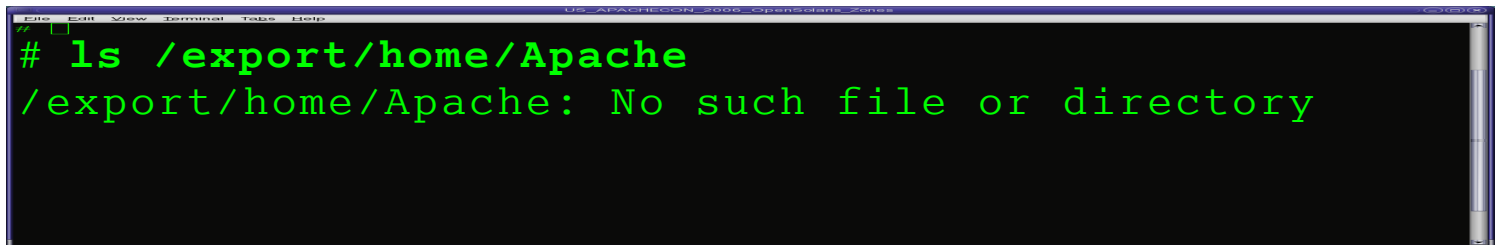
◆ Zones in Action - Creating a Zone



```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
## zonecfg -z Apache
Apache: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:Apache> create
zonecfg:Apache> set zonepath=/export/home/Apache
zonecfg:Apache> add net
zonecfg:Apache:net> set address=192.168.0.50/24
zonecfg:Apache:net> set physical=bge0
zonecfg:Apache:net> end
zonecfg:Apache> verify
zonecfg:Apache> commit
zonecfg:Apache> exit
```

OpenSolaris Zones

◆ Zones in Action - Creating a Zone (2)

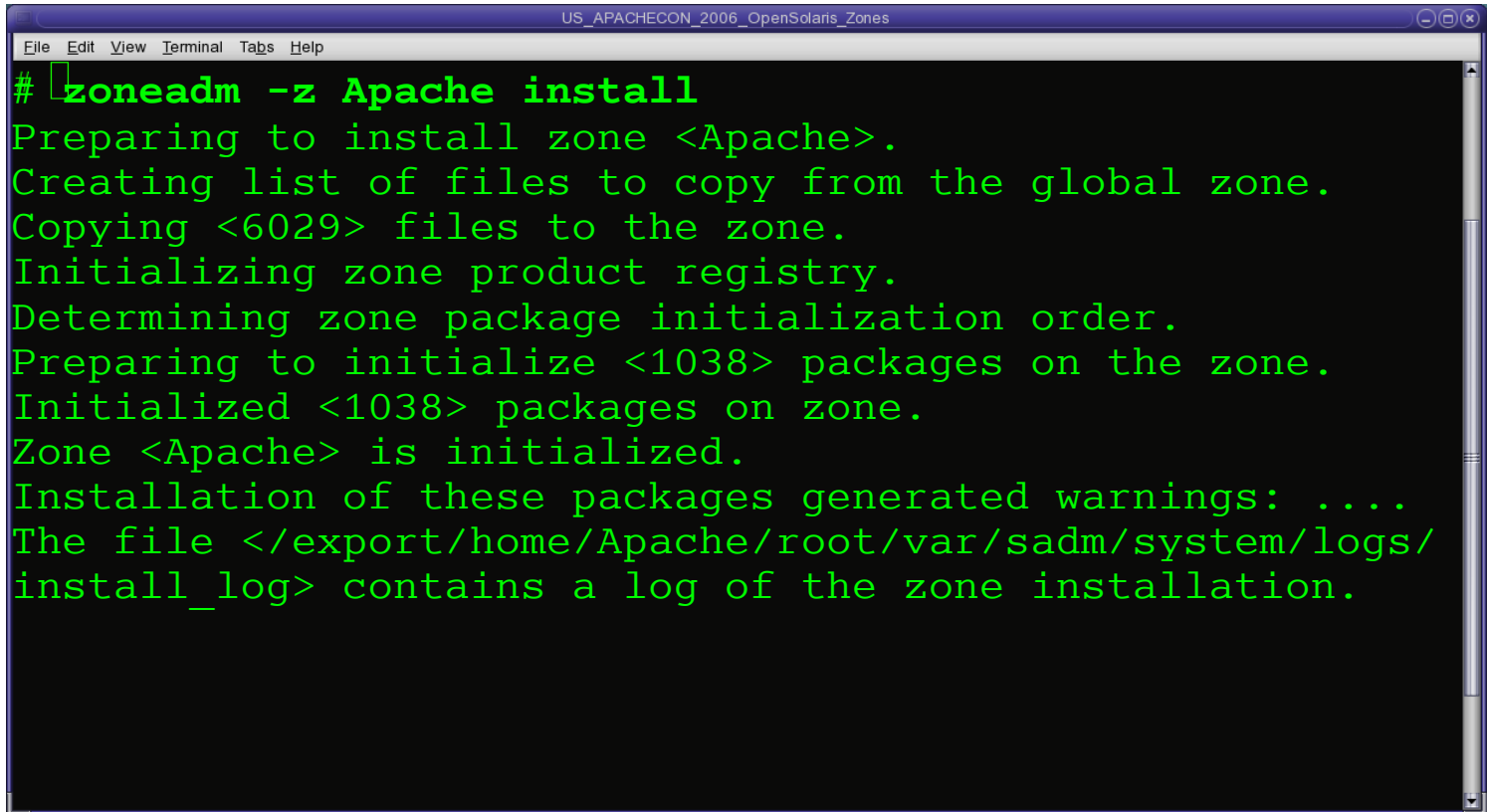
A terminal window with a black background and green text. The window title is "US_APACHECON_2006_OpenSolaris_Zones". The prompt is "#". The command entered is "ls /export/home/Apache". The output is "/export/home/Apache: No such file or directory".

```
US_APACHECON_2006_OpenSolaris_Zones
# ls /export/home/Apache
/export/home/Apache: No such file or directory
```

- ◆ Zone “Apache”
 - ◆ Create only configures the zone
 - ◆ zoneadm will install and boot the zone

OpenSolaris Zones

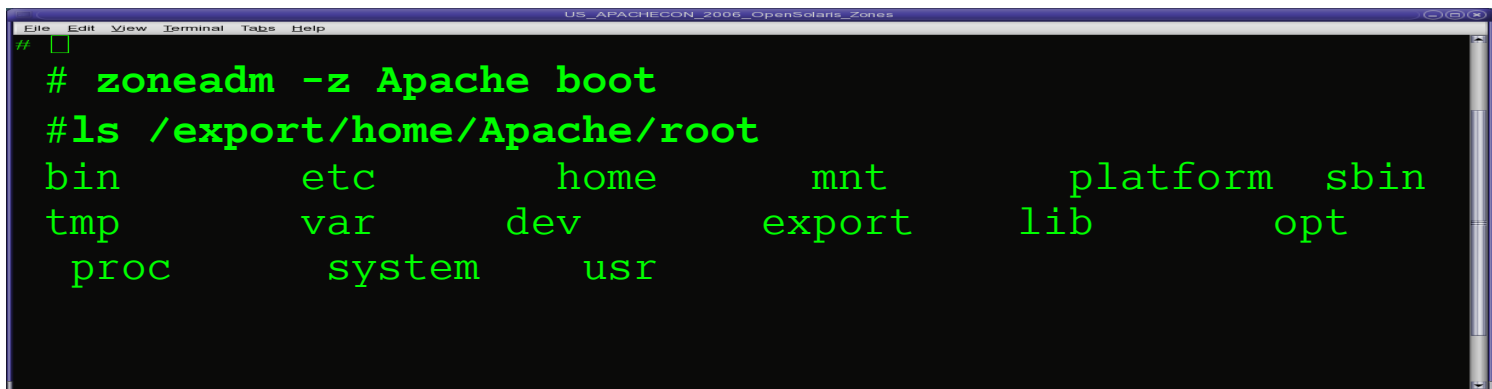
◆ Zones in Action - Installing a Zone



```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# zoneadm -z Apache install
Preparing to install zone <Apache>.
Creating list of files to copy from the global zone.
Copying <6029> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1038> packages on the zone.
Initialized <1038> packages on zone.
Zone <Apache> is initialized.
Installation of these packages generated warnings: ....
The file </export/home/Apache/root/var/sadm/system/logs/
install_log> contains a log of the zone installation.
```

OpenSolaris Zones

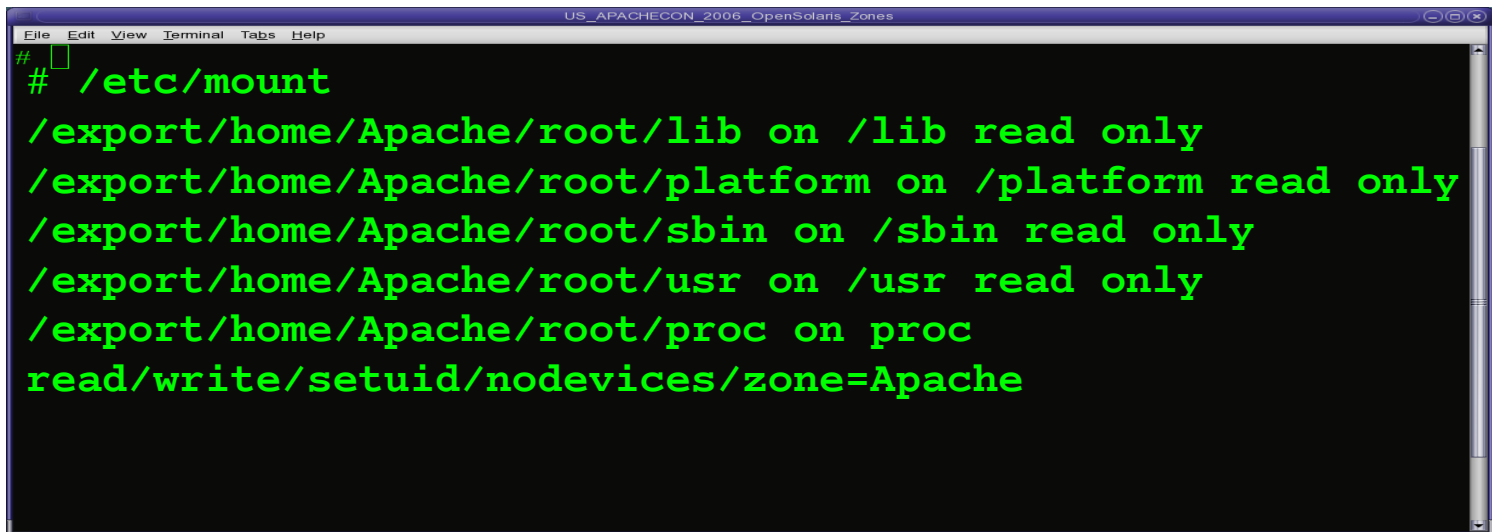
- ◆ Zones in Action - Installing a Zone (2)
 - ◆ The necessary directories are created
 - ◆ The zone is ready for booting



```
US_APACHECON_2006_OpenSolaris_Zones
# #
# zoneadm -z Apache boot
#ls /export/home/Apache/root
bin      etc      home    mnt      platform sbin
tmp      var      dev     export   lib      opt
proc     system  usr
```

OpenSolaris Zones

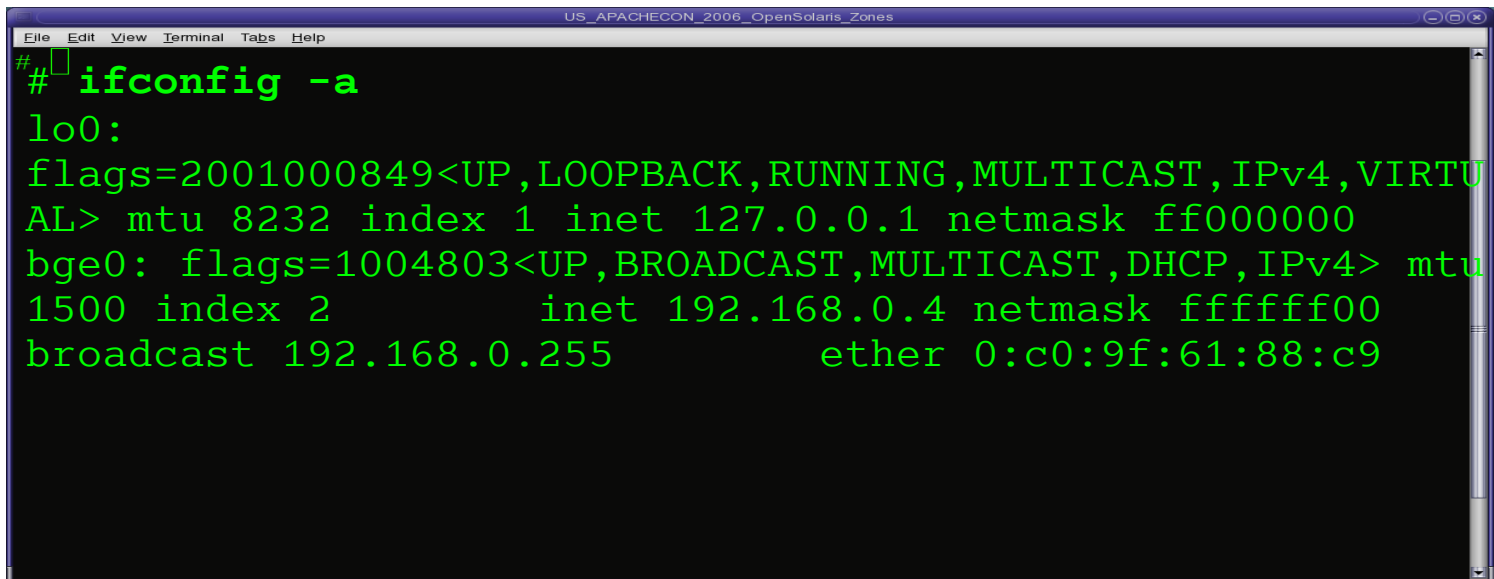
- ◆ Zones in Action - Installing a Zone (3)
 - ◆ Packages are not reinstalled!
 - ◆ usr, lib, sbin etc are mounted read-only



```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# /etc/mount
/export/home/Apache/root/lib on /lib read only
/export/home/Apache/root/platform on /platform read only
/export/home/Apache/root/sbin on /sbin read only
/export/home/Apache/root/usr on /usr read only
/export/home/Apache/root/proc on proc
read/write/setuid/nodevices/zone=Apache
```

OpenSolaris Zones

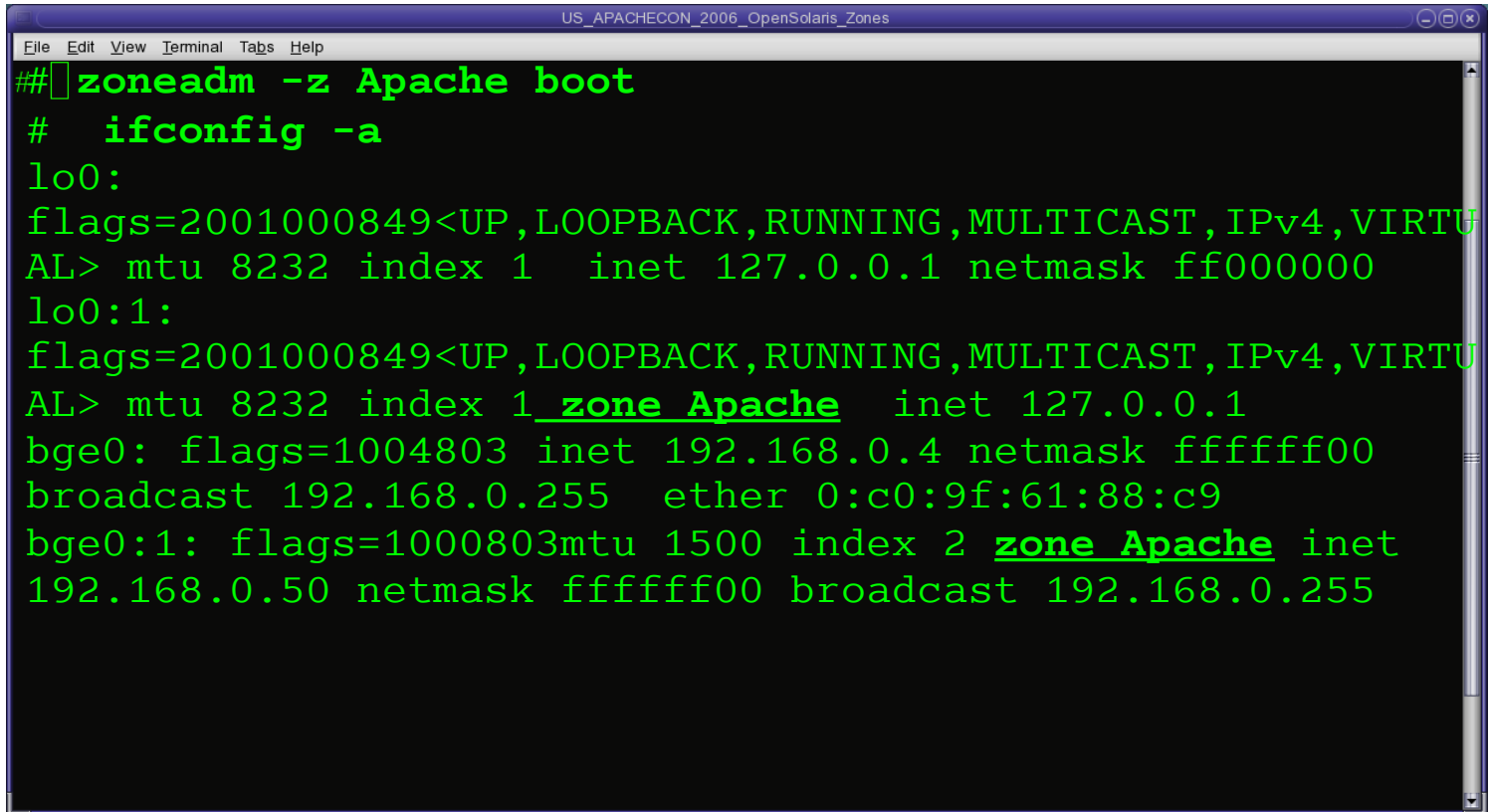
- ◆ Zones in Action - Installing a Zone (4)
 - ◆ zone “Apache” not ready to use yet
 - ◆ Need to “boot” for interfaces to show up



```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# ifconfig -a
lo0:
flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1 inet 127.0.0.1 netmask ff000000
bge0: flags=1004803<UP,BROADCAST,MULTICAST,DHCP,IPv4> mtu
1500 index 2          inet 192.168.0.4 netmask ffffffff00
broadcast 192.168.0.255          ether 0:c0:9f:61:88:c9
```

OpenSolaris Zones

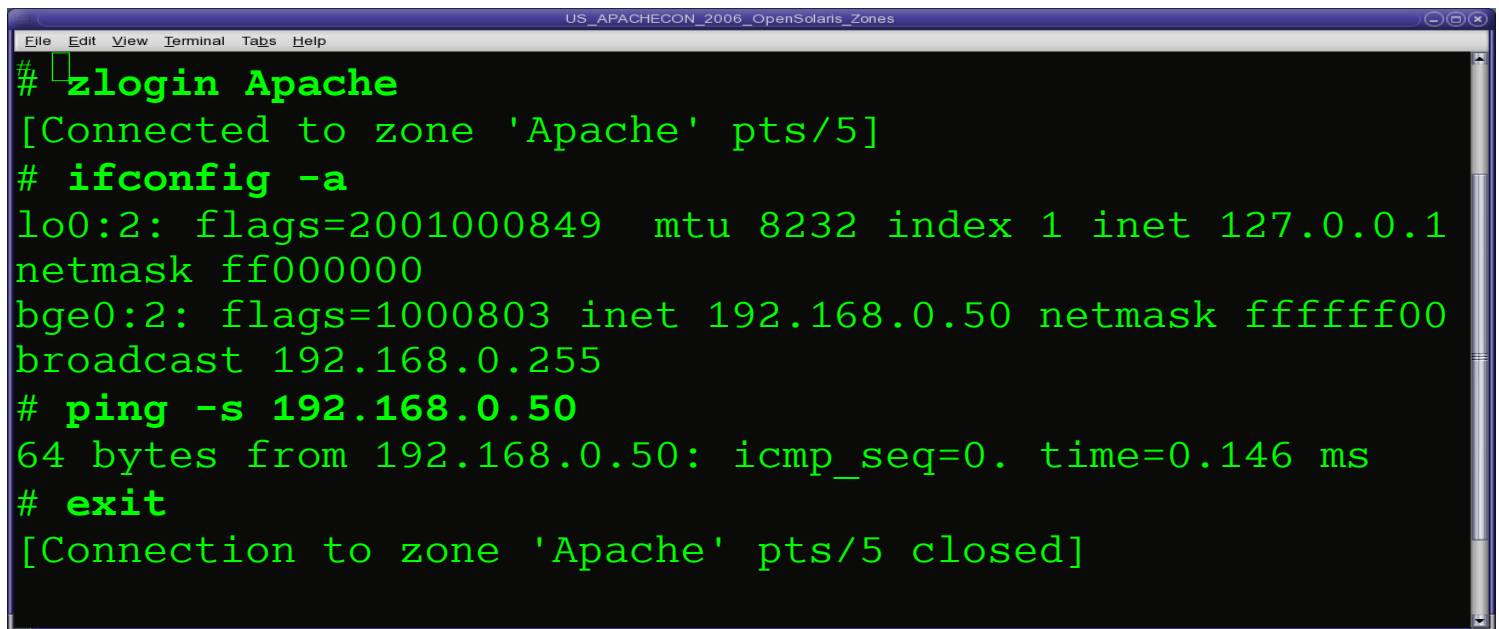
◆ Zones in Action - Booting the Zone



```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
## zoneadm -z Apache boot
# ifconfig -a
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 Apache inet 127.0.0.1
bge0: flags=1004803 inet 192.168.0.4 netmask ffffffff00
broadcast 192.168.0.255 ether 0:c0:9f:61:88:c9
bge0:1: flags=1000803mtu 1500 index 2 zone Apache inet
192.168.0.50 netmask ffffffff00 broadcast 192.168.0.255
```

OpenSolaris Zones

- ◆ Zones in Action - Logging into the Zone
 - ◆ Run `zlogin -C Apache` first and configure
 - ◆ The zone is now ready!



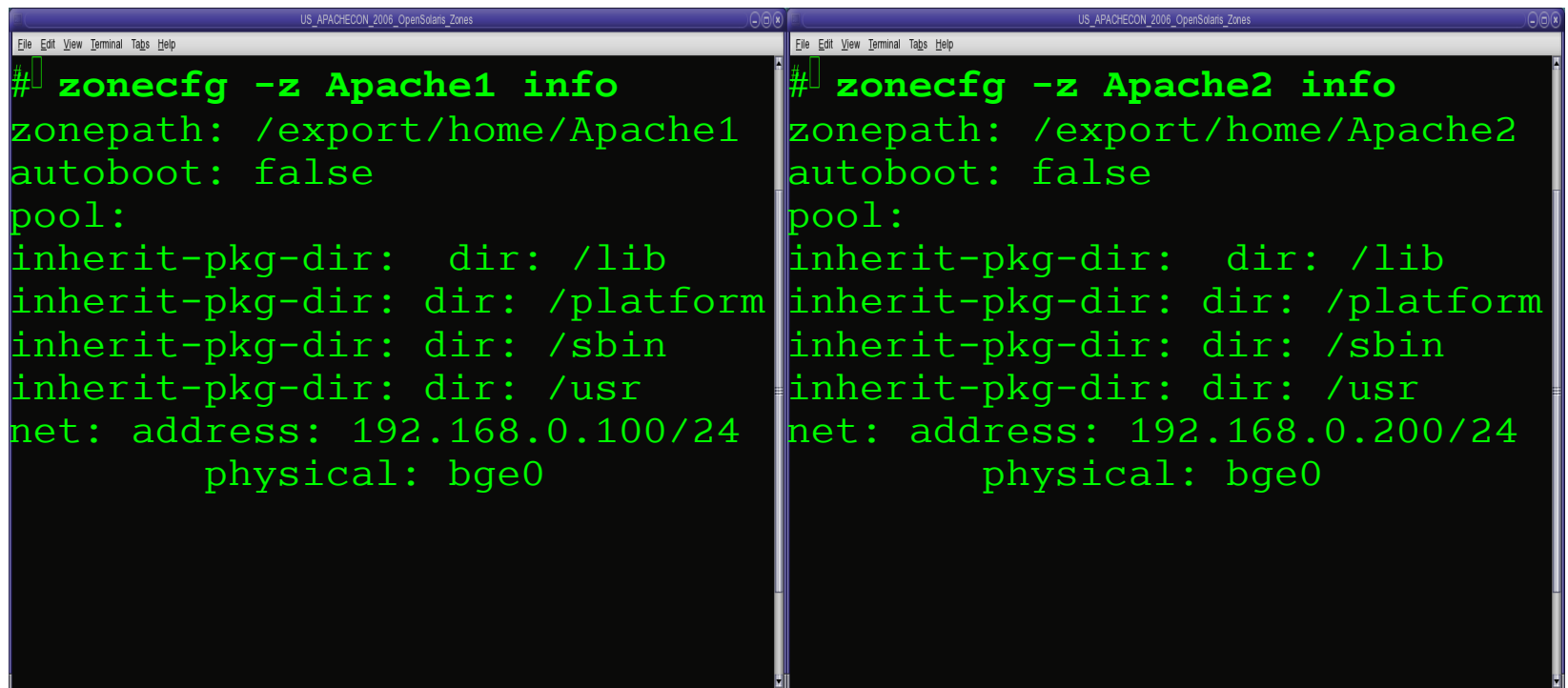
```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# zlogin Apache
[Connected to zone 'Apache' pts/5]
# ifconfig -a
lo0:2: flags=2001000849  mtu 8232  index 1  inet 127.0.0.1
netmask ff000000
bge0:2: flags=1000803  inet 192.168.0.50  netmask ffffffff00
broadcast 192.168.0.255
# ping -s 192.168.0.50
64 bytes from 192.168.0.50: icmp_seq=0.  time=0.146 ms
# exit
[Connection to zone 'Apache' pts/5 closed]
```

OpenSolaris Zones

- ◆ Zones in Action - Web Server Virtualization
 - ◆ Support two different set of user groups
 - ◆ Group 1 requires Apache Web server 1.3.9
 - ◆ Group 2 requires Apache Web server 2.0.50
 - ◆ One physical host
 - ◆ Simultaneous access to both web servers
 - ◆ Each web server and system should be protected should one of them be compromised

OpenSolaris Zones

- ◆ Zones in Action - Web Server Virtualization (2)
 - ◆ Create two local zones Apache1 and Apache2

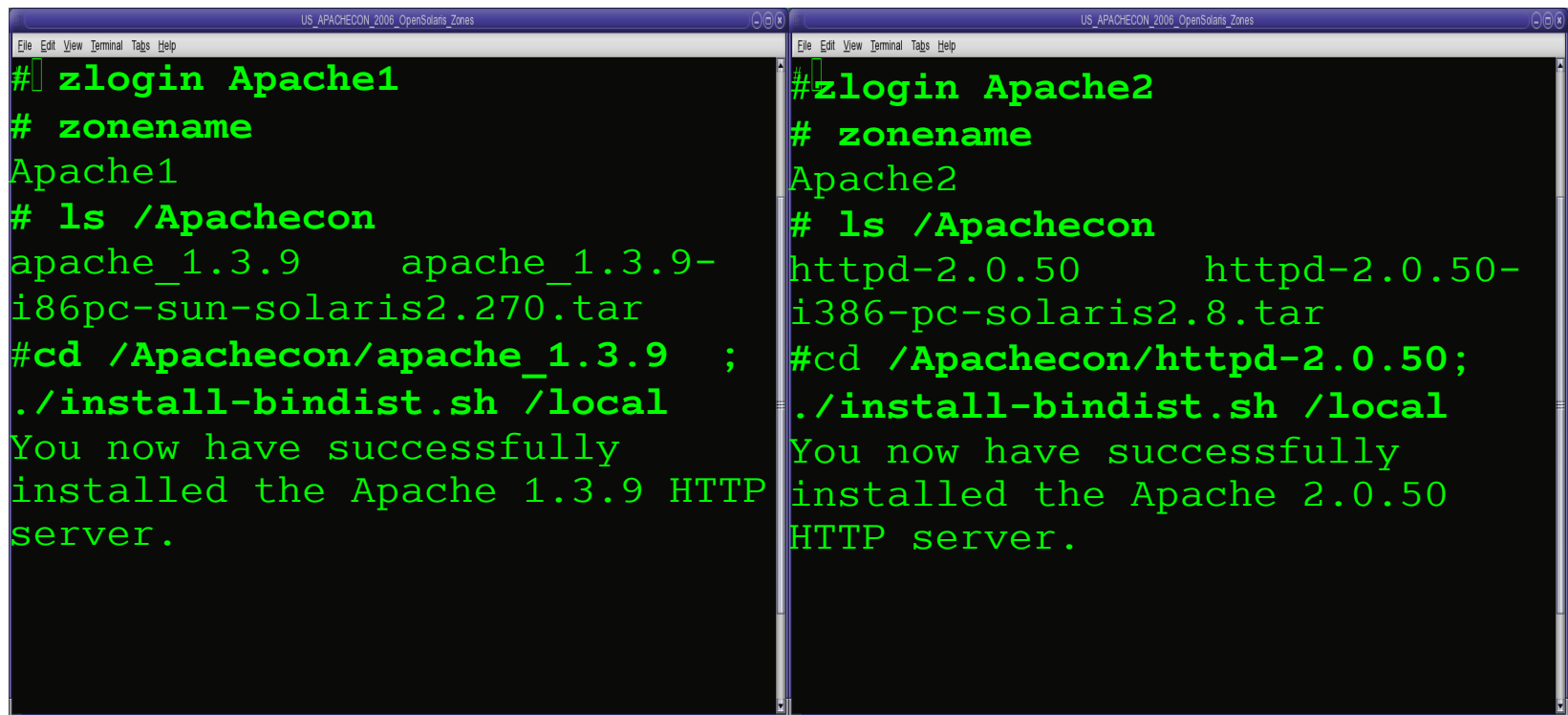


```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# zonecfg -z Apache1 info
zonepath: /export/home/Apache1
autoboot: false
pool:
inherit-pkg-dir: dir: /lib
inherit-pkg-dir: dir: /platform
inherit-pkg-dir: dir: /sbin
inherit-pkg-dir: dir: /usr
net: address: 192.168.0.100/24
      physical: bge0

US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# zonecfg -z Apache2 info
zonepath: /export/home/Apache2
autoboot: false
pool:
inherit-pkg-dir: dir: /lib
inherit-pkg-dir: dir: /platform
inherit-pkg-dir: dir: /sbin
inherit-pkg-dir: dir: /usr
net: address: 192.168.0.200/24
      physical: bge0
```

OpenSolaris Zones

- ◆ Zones in Action - Web Server Virtualization (3)
 - ◆ Log in to each zone and install the application



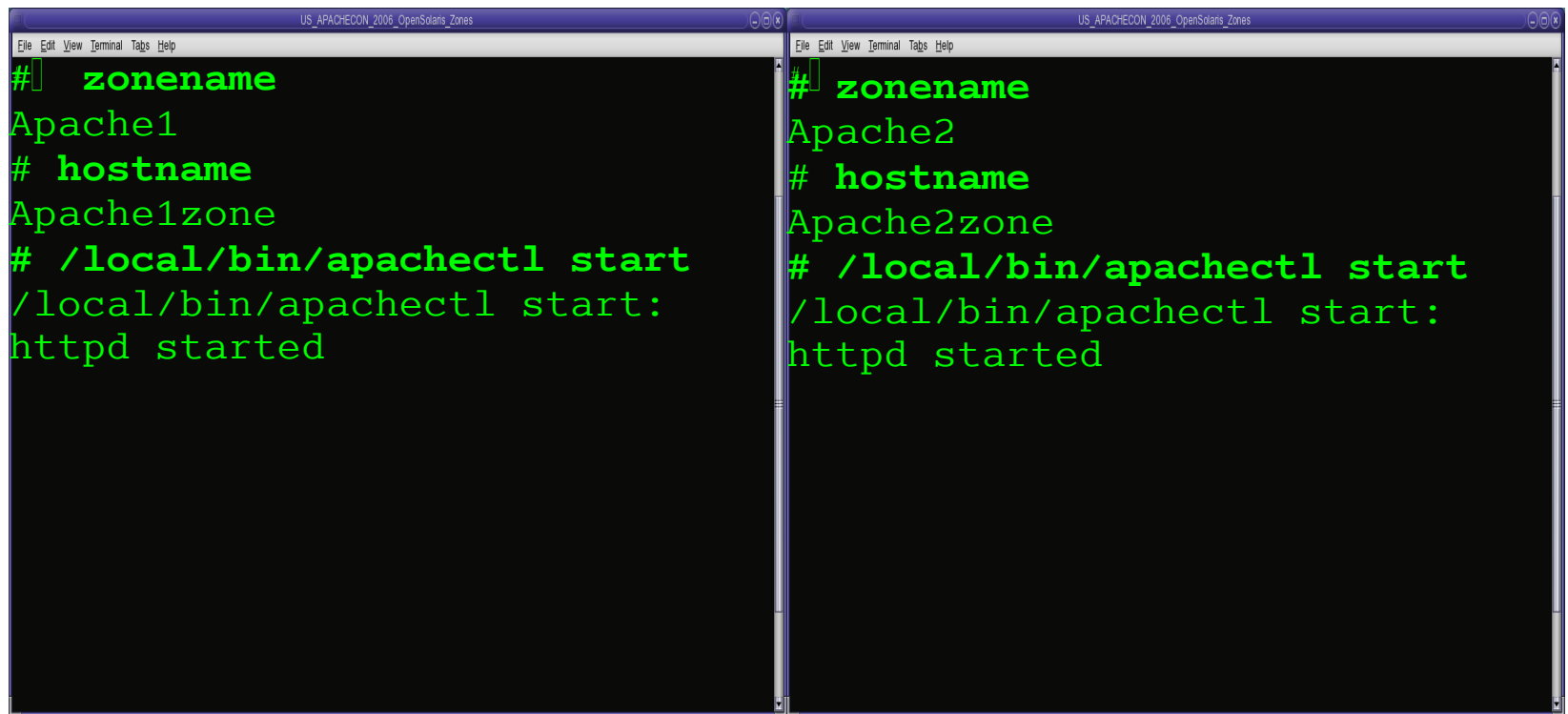
The image shows two side-by-side terminal windows. The left window is titled 'zlogin Apache1' and shows the installation of Apache 1.3.9. The right window is titled 'zlogin Apache2' and shows the installation of Apache 2.0.50. Both windows show the same sequence of commands: logging in to the zone, listing the contents of the /Apachecon directory, and running the install script.

```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# zlogin Apache1
# zonename
Apache1
# ls /Apachecon
apache_1.3.9      apache_1.3.9-
i86pc-sun-solaris2.270.tar
#cd /Apachecon/apache_1.3.9 ;
./install-bindist.sh /local
You now have successfully
installed the Apache 1.3.9 HTTP
server.

US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# zlogin Apache2
# zonename
Apache2
# ls /Apachecon
httpd-2.0.50      httpd-2.0.50-
i386-pc-solaris2.8.tar
#cd /Apachecon/httpd-2.0.50;
./install-bindist.sh /local
You now have successfully
installed the Apache 2.0.50
HTTP server.
```

OpenSolaris Zones

- ◆ Zones in Action - Web Server Virtualization (4)
 - ◆ Start the application



The image shows two side-by-side terminal windows. Each window has a title bar that reads "US_APACHECON_2006_OpenSolaris_Zones" and a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The left terminal window shows the following text:

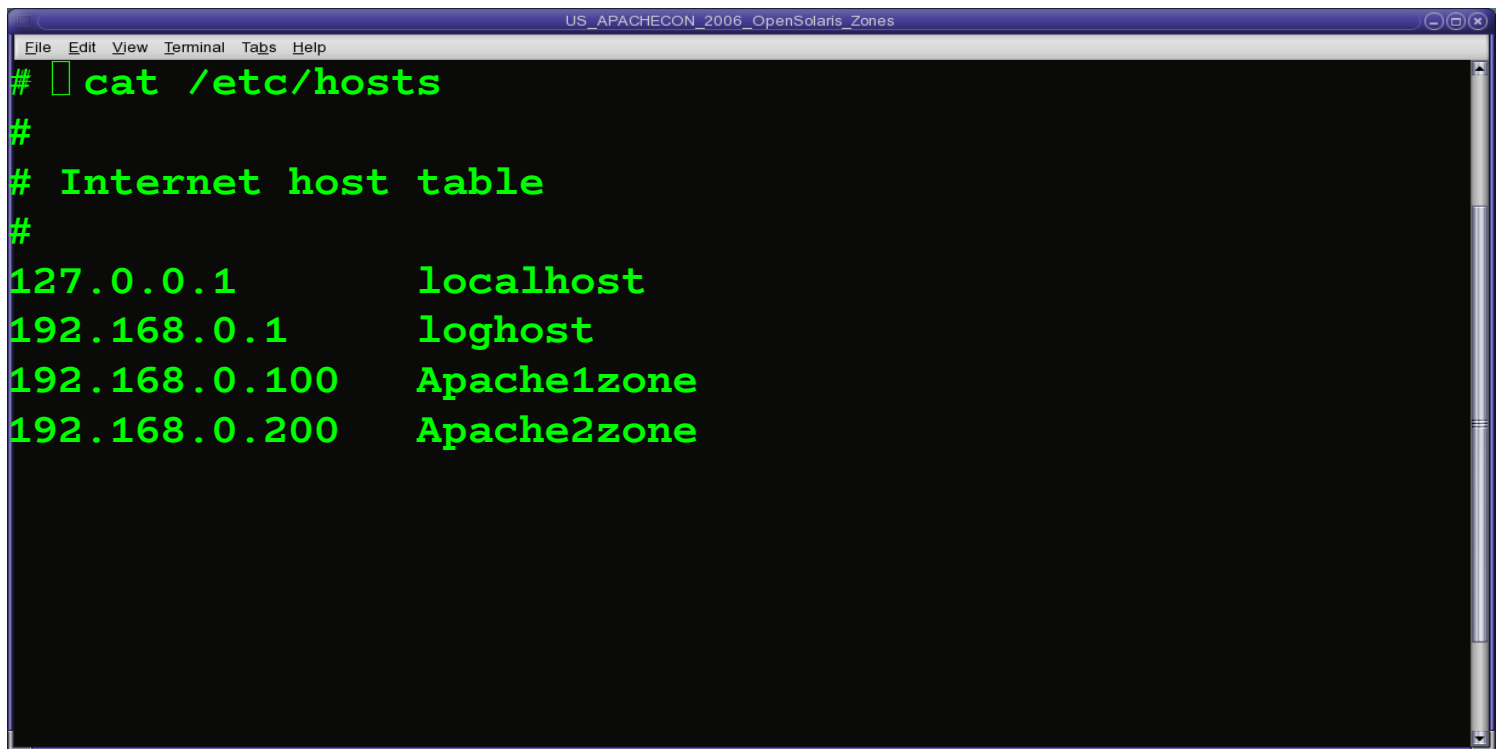
```
# zonename
Apache1
# hostname
Apache1zone
# /local/bin/apachectl start
/local/bin/apachectl start:
httpd started
```

The right terminal window shows the following text:

```
# zonename
Apache2
# hostname
Apache2zone
# /local/bin/apachectl start
/local/bin/apachectl start:
httpd started
```

OpenSolaris Zones

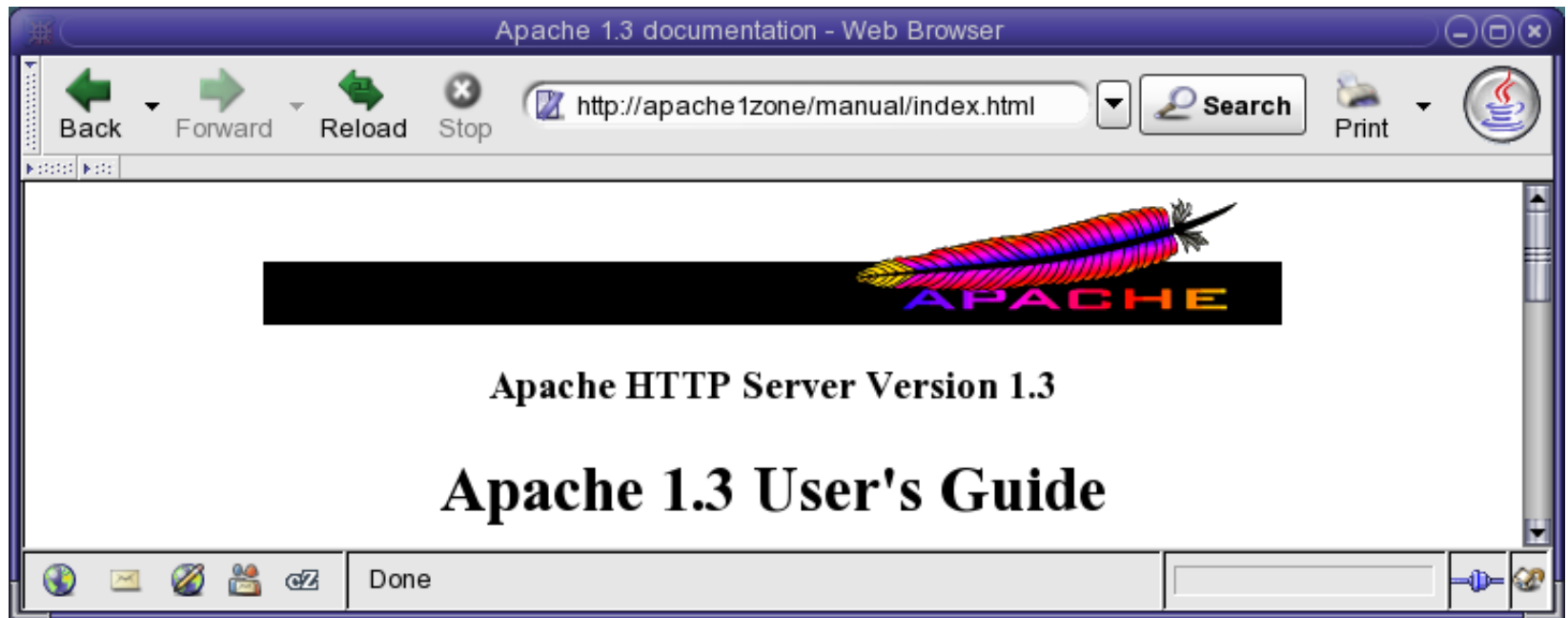
- ◆ Zones in Action - Web Server Virtualization (5)
 - ◆ In “global” zone

A terminal window titled "US_APACHECON_2006_OpenSolaris_Zones" showing the output of the command "cat /etc/hosts". The output lists IP addresses and their corresponding hostnames: 127.0.0.1 localhost, 192.168.0.1 loghost, 192.168.0.100 Apache1zone, and 192.168.0.200 Apache2zone.

```
File Edit View Terminal Tabs Help
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1      localhost
192.168.0.1   loghost
192.168.0.100 Apache1zone
192.168.0.200 Apache2zone
```

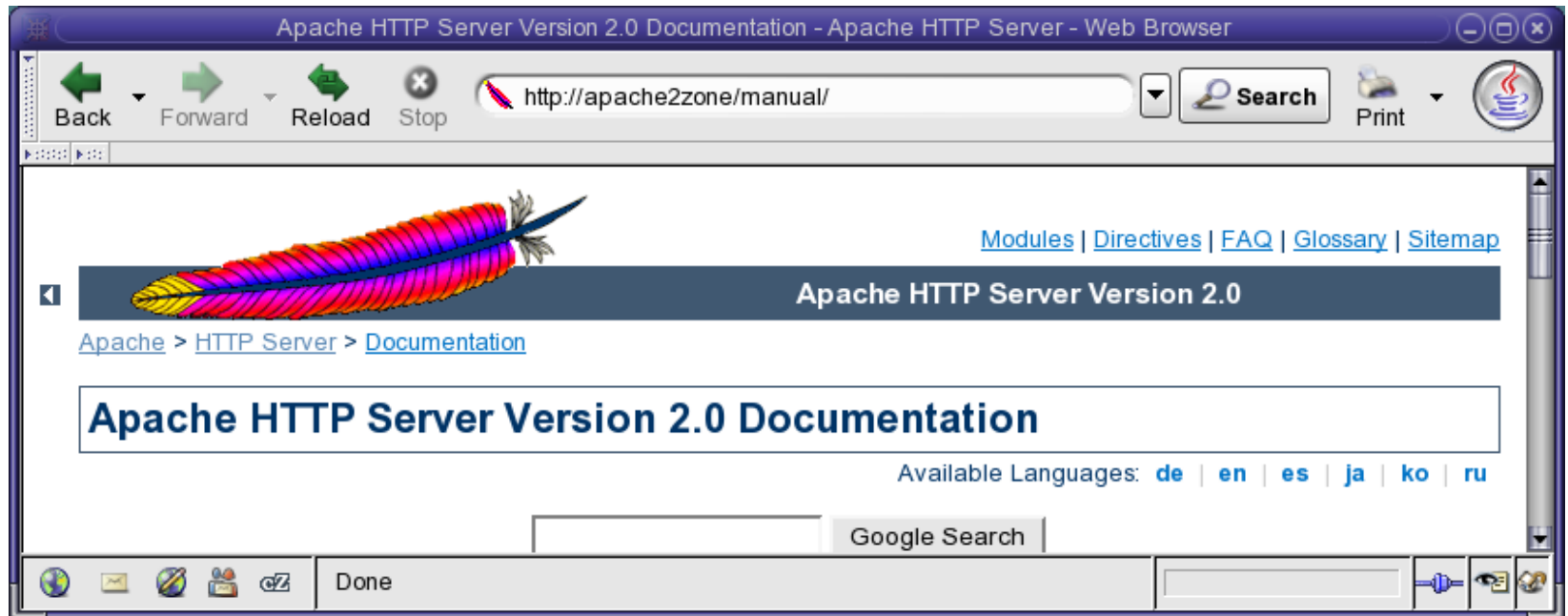
OpenSolaris Zones

- ◆ Zones in Action - Web Server Virtualization (6)



OpenSolaris Zones

◆ Zones in Action - Web Server Virtualization (7)

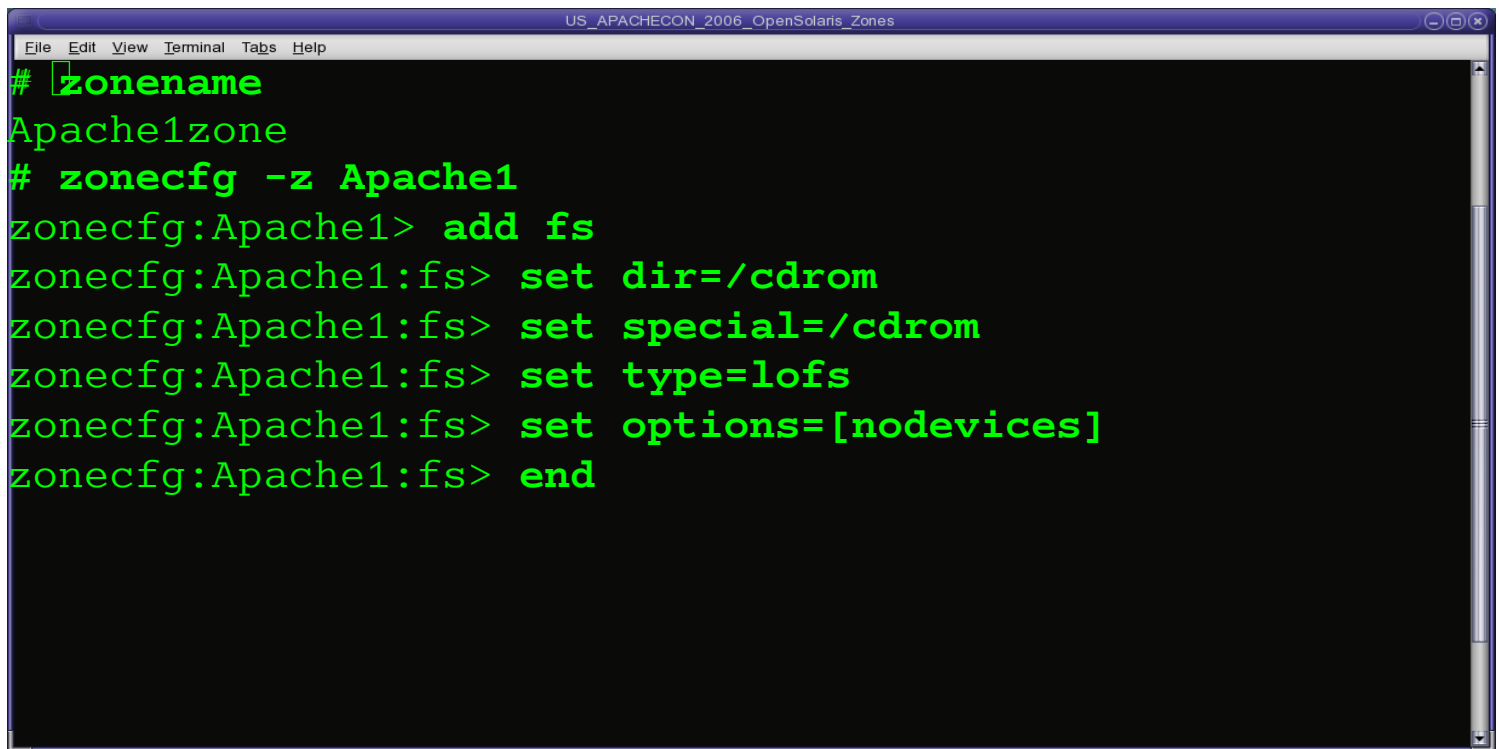


OpenSolaris Zones

- ◆ Zones in Action - Web Server Virtualization (8)
 - ◆ End user sees each zone as a different system
 - ◆ Each web server has its own name service
 - ◆ `/etc/nsswitch.conf`
 - ◆ `/etc/resolv.conf`
 - ◆ and so on
 - ◆ A malicious attack on one web server is contained to that zone
 - ◆ Port conflicts no longer a problem!

OpenSolaris Zones

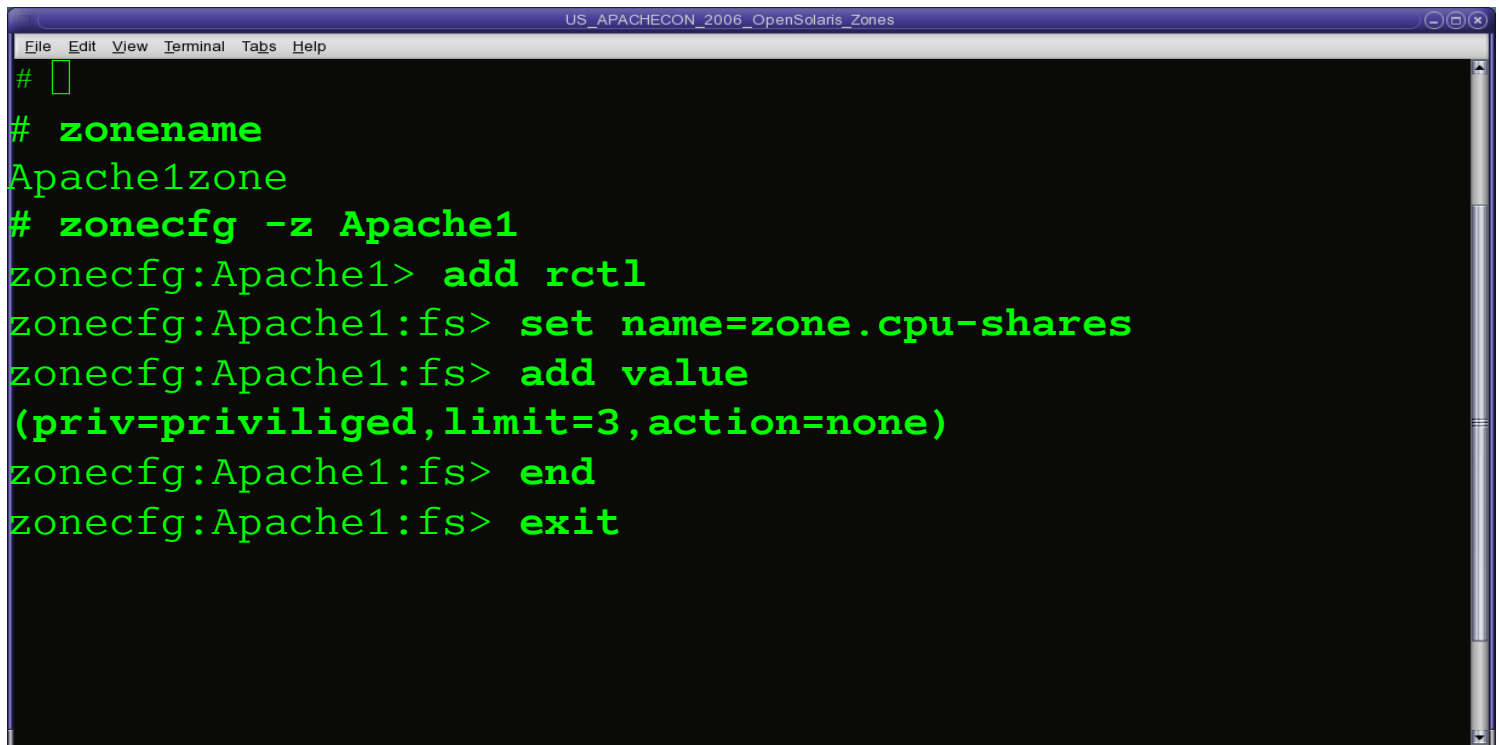
- ◆ Zones in Action – CDROM in local zone



```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# zonename
Apache1zone
# zonecfg -z Apache1
zonecfg:Apache1> add fs
zonecfg:Apache1:fs> set dir=/cdrom
zonecfg:Apache1:fs> set special=/cdrom
zonecfg:Apache1:fs> set type=lofs
zonecfg:Apache1:fs> set options=[nodevices]
zonecfg:Apache1:fs> end
```

OpenSolaris Zones

◆ Zones in Action – Assigning CPU shares



```
US_APACHECON_2006_OpenSolaris_Zones
File Edit View Terminal Tabs Help
# 
# zonename
Apache1zone
# zonecfg -z Apache1
zonecfg:Apache1> add rctl
zonecfg:Apache1:fs> set name=zone.cpu-shares
zonecfg:Apache1:fs> add value
(priv=privileged,limit=3,action=none)
zonecfg:Apache1:fs> end
zonecfg:Apache1:fs> exit
```

OpenSolaris Zones

- ◆ Recent enhancements
 - ◆ Integration with ZFS (Available with Solaris Express and Solaris 10 OS)
 - ◆ Moving a non-global zone within the system
 - ◆ Migration of zones across hosts
 - ◆ Patching via NFS mounted file systems
 - ◆ Faster provisioning w/ ZFS

OpenSolaris Zones

- ◆ Recent enhancements (2)
 - ◆ Zone Boot Arguments II
 - ◆ System V resource controls for Zones
 - ◆ zone/project.max-locked-memory Resource Controls
 - ◆ Amendment to zone/project.max-locked-memory Resource Controls

OpenSolaris Zones

- ◆ Some customer deployments
 - ◆ Apache Community
 - ◆ <http://www.apache.org/dev/solaris-zones.html>
 - ◆ Joyent
 - ◆ <http://www.textdrive.com/hosting/container>

OpenSolaris Zones

- ◆ Example Uses
 - ◆ Hostile and untrustworthy applications
 - ◆ Example: Two web servers each binding to port 80
 - ◆ Untrustworthy software that should be isolated
 - ◆ In educational institutions
 - ◆ Data center consolidation
 - Multiple database instances with administrators

OpenSolaris Zones

- ◆ Example Uses (2)
 - ◆ Software development
 - ◆ Testing – Each zone can host a different test environment

OpenSolaris Zones

- ◆ References

- ◆ Architectural Diagrams & Illustrations

- ◆ Daniel Price, Andrew Tucker: Solaris Zones: Operating System Support for Consolidating Commercial Workloads. LISA 2004: 241-254

- ◆ Menno Lageman, Solaris Containers--What They Are and How to Use Them.

- <http://www.sun.com/blueprints/0505/819-2679.html>

OpenSolaris Zones

◆ References

◆ OpenSolaris Zones Community

◆ <http://opensolaris.org/os/community/zones>

◆ Zones Page on BigAdmin Site

◆ <http://www.sun.com/bigadmin/content/zones>

◆ Zones FAQ page

◆ <http://opensolaris.org/os/community/zones/faq>

OpenSolaris Zones

- ◆ Acknowledgements
 - ◆ Many thanks to the following Sun Microsystems colleagues for their assistance in providing and reviewing the materials that made this tutorial possible:
 - ◆ Ganesh Hiregoudar and Vineeth Pillai
 - ◆ Daniel Price and David Comay
 - ◆ Please check out their blogs for the latest Zones info at:
 - ◆ <http://opensolaris.org/os/blogs>

Copyright 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd. X/Open is a registered trademark of X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, OpenSolaris, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.