

Express-Guide
~to~
Basic Setup of
SOLARIS ZONES
Native Zones & Ix-Branded Zones



by, [ABK](http://www.twitter.com/aBionic) ~ <http://www.twitter.com/aBionic>

::Task Detail::

- ◆ Creating/Managing Solaris Zones
 - ◆ mirroring an existing CentOS Box over a Zone
 - ◆ setting up SVN service over a Zone
 - ◆ setting CIFS over a Zone
-

::Background::

Solaris Zones are a part of ***Solaris Container Technology***. Zones manage the namespace isolation for containers implementing virtualization.

In Solaris 10, ***containers are zone using Resource Management Feature*** via Solaris Resource Manager. There is no performance overhead to this approach.

Resources are ***used as a 'Dynamic resource Pool' managed between containers using a 'Fair Share Scheduler'***.

Broadly there are two types of zones:

◆ **Native Zone**

These are basic ***stripped down native Solaris O.S. Instances***.

Native zones are further of two types where

- '***Small Zones***' (also known as Sparse Root Zone) have several system directories shared with Solaris O.S. or Global Zone in non-writable mode
- '***Big Zones***' (also known as Whole Root Zone) have all independent directories

◆ **Ix-Branded Zone**

these are zones ***installed from installer of O.S.***, currently only linux branded zones are available also called Ix-zones.

::Execution Method::

(a.) Creating native {small,big} and lx-branded zones

◆ **Setting up Resource Pool to be used by zones**

- Enabling Resource Pool features
 - #pooladm -e
- Saving current resource pool
 - #pooladm -s
- List current Pools
 - #pooladm
 - ***{generally only 'pool_default' is present on fresh zone}***
- Configuring 'default_pool' to enable Fair Share Scheduler over it
 - #poolcfg -c 'modify pool pool_default (string pool.scheduler="FSS")'
 - #pooladm -c
- Priority Controller moving all processes and resources under Fair Share Scheduler
 - #priocntl -s -c FSS -i class TS
 - #priocntl -s -c FSS -i pid 1

◆ **Configuring a Solaris Zone**

- This lists the current zones
 - #zoneadm list -cv
- Configuring a new Native Zone
 - registering a new Zone
 - #zonecfg -z newZoneName
- regarding 3 different types of zones follow respective command
 - for creating a ***native small-zone*** {with shared directories}
 - zonecfg:newZoneName>create
 - for creating a ***native big-zone*** {with independent directories}
 - zonecfg:newZoneName>create -b
 - for creating a ***lx-branded*** zone
 - zonecfg:lxZoneName>create -t SUNWlx

- assigning it a location on HDD to be installed
 - `zonecfg:newZoneName>set`
 `zonepath=/export/home/zones/newZoneName`
- Adding a Network Interface Resource to it
 - `zonecfg:newZoneName>add net`
 - `zonecfg:newZoneName:net>set address=192.168.16.61`
 - `zonecfg:newZoneName:net>set physical=eth0`
 - `zonecfg:newZoneName:net>end`
- Assign a Resource Pool (should be already existing) to it
 - `zonecfg:newZoneName>set pool=pool_default`
- Adding a resource controller to this Zone
 - `zonecfg:newZoneName>add rctl`
 - `zonecfg:newZoneName:rctl>set name=zone.cpu-shares`
 - `zonecfg:newZoneName:rctl>`
 `add value (priv=privileged,limit=1,action=none)`
 - `zonecfg:newZoneName:rctl>end`
- Giving a CD-ROM access (required if installing lx-zone from ISO or CD)
 - `zonecfg:newZoneName>add fs`
 - `zonecfg:newZoneName:fs>set dir=/cdrom`
 - `zonecfg:newZoneName:fs>set special=/cdrom`
 - `zonecfg:newZoneName:fs>set typr=lofs`
 - `zonecfg:newZoneName>set options=[nodevices]`
 - `zonecfg:newZoneName>end`
- Verify, Save and Exit
 - `zonecfg:newZoneName>verify`
 - `zonecfg:newZoneName>commit`
 - `zonecfg:newZoneName>exit`
- Creating the HDD location for Zone
 - `#mkdir -p /export/home/zones/newZoneName`
- Granting required permissions to location
 - `#chmod 700 /export/home/zones/newZoneName`
- Confirming the registration of Zone Configuration
 - `#zoneadm list -cv`

- It should show a listing for currently created zone like
 - `newZoneName configured at /export/home/zones/newZoneName`, it is native and shared (small-zone)
- ◆ **Installing the already configured zone**
 - **Installing** the zone if it's a **Native** {small or big} zone
 - `#zoneadm -z newZoneName install`
 - if it's a **lx-brand zone** with O.S. **TarBall**, automatically **creating ZFS**
 - `#zoneadm -z newZoneName install -d /tmp/os.tgz`
 - if it's a **lx-branded zone** with O.S. **TarBall**, **not creating ZFS**
 - `#zoneadm -z newZoneName install -x nodataset -d /tmp/os.tgz`
 - if no archive path is given then default is Disc Drive, but if you are installing from Disc Drive, you need to install VOLFS like:
 - `#svcadm enable svc:/system/filesystem/volfs:default`
 - `#svcs | grep volfs`
 - If its installed without any error, just check its status using
 - `#zoneadm list -cv`
 - it should show a listing for currently created zone like ***newZoneName installed /export/home/zones/newZoneName native shared***
- ◆ **Using the installed Zone**
 - Now either make it ready to boot, or directly boot which will make it ready itself
 - `#zoneadm -z newZoneName ready`
 - It should show a listing for currently created zone like
 - `newZoneName ready /export/home/zones/newZoneName native shared`
 - `#zoneadm -z newZoneName boot`
 - It should show a listing for currently created zone like
 - ***newZoneName running /export/home/zones/newZoneName native shared***
 - To **login**
 - `#zlogin newZoneName`
 - Now you are inside the Zone, running 'uname -a' should present you with newZoneName

- To **login** into Zone **Console** like remote connect
 - `#zlogin -C newZoneName`
- To **exit** the zone
 - `#exit`
- To **halt** the zone simply use
 - `#zoneadm -z newZoneName halt`
- it should show a listing for currently created zone like
 - ***newZoneName running /export/home/zones/newZoneName native shared***
- To **reboot** the zone simply use
 - `#zoneadm -z newZoneName reboot`
- To **uninstall** the zone
 - `#zoneadm -z newZoneName uninstall -F`

(b.) Mirroring an existing CentOS Box over a Zone

- ◆ There are two ways to achieve this
 - TarBall the entire distro you want to port to Zone and use that TarBall to install the Zone.
 - Suppose, you already have an lx-branded zone and use the same. Then you need to use utility like RSync to Sync the files from Source Machine to lx-Zone.
- ◆ You can also add packages like svn, gcc, make, netsnmp, openssl, CoolStack's (apache2, mysql, php, perl, python, ruby, squid) to lx-zone and they work great over Zone.

(c.) Setting up SVN service over a Zone

- ◆ Users connect to svn mirror servers, the WebDAV SVN module serves content from the local system, and sends commits to the main server. Then main server pushes commit to mirrors using 'svnsync' over a protected link only writable by main server.
 - Install Collabnet SVN client & server binaries {available at

'<http://www.collab.net/downloads/subversion/solaris.html>'] }

- Create a symlink collabnet modules a

- `#ln -s /opt/CollabNet_Subversion/modules/mod_dav_svn.so /etc/httpd/modules/mod_dav_svn.so`
- `#ln -s /opt/CollabNet_Subversion/modules/mod_authz_svn.so /etc/httpd/modules/mod_authz_svn.so`

- Add below lines to 'httpd.conf' under Apache2 directory as

```
LoadModule dav_svn_module /etc/httpd/modules/mod_dav_svn.so
LoadModule authz_svn_module /etc/httpd/modules/mod_authz_svn.so
<Location /someproject>
DAV svn
SVNPath /repos/svn/repos/someproject
AuthzSVNAccessFile /repos/svn/access/someproject/svn_access.conf
AuthType Basic
AuthName "Active Directory LDAP Authentication"
AuthBasicProvider ldap
AuthzLDAPAuthoritative off
AuthLDAPBindDN user@adserver.thoughtworks.com
AuthLDAPBindPassword somePassword
AuthLDAPURL "ldap://adserver.company.com:389/ou=Principal.dc=dcString1,dc=dcStrin2?SAMAccountName?sub?\(&\(objectClass=user\)\)"
require vaild-user
SVNPathAuthz off
</Location>
```

- Reload httpd service

-

- Add following lines to '/repos/svn/access/someproject/svn_access.conf'

- `can_write_group=aduserA, aduserB, aduserC`
- `read_only_group=aduserD, aduserE, aduserF`
- `no_access_group=aduserG, aduserH, aduserJ`
- `[repository:]`
- `@can_write_group=rw`
- `@read_only_group=r`
- `@no_access_group=`

- Create a repository as follows:

- `svnadmin create /repos/svn/repos/someproject`
- change permissions as follows
- `chmod -R g+w /repos/svn/repos/someproject`
- `chown -R apache.apache /repos/svn/repos/someproject`

- Similarly, you can setup mirror server with the configuration given at Link Above.

(d.) Setting CIFS over a Zone

- ◆ Initial reading disclosed its not possible over local zones, only global zone could support CIFS.
 - ◆ So just did practical with setting up SAMBA server on Solaris Zones; implemented SWAT (Samba Web Admin Tool) for easy configuration.
 - for Solaris 10, SAMBA came up real easy to configure
 - `#svcs samba wins swat`
 - `#svcadm enable samba`
 - `#svcadm enable wins`
 - `#svcadm enable swat`
 - Simply browsing `http://samba_Zone_IPaddress:901/` presents with a nice SWAT GUI to configure SAMBA service on that zone.
To get start with, you need to
 - `> select 'Shares', add new share with proper configuration`
 - `> select 'Users', to add Users`
 - `> Restart Services from UI itself`
 - `> now try accessing this share from Windows as normal Windows Share using User created`
-

::Tools/Technology Used::

- ◆ Solaris Zones: <http://www.solarisinternals.com/wiki/index.php/Zones>
 - ◆ CoolStack Software Bundles: {now superseded by WebStack} ~ <http://hub.opensolaris.org/bin/view/Project+webstack/sunwebstack>
 - ◆ Rsync: <http://en.wikipedia.org/wiki/Rsync>
 - ◆ SVN: <http://subversion.apache.org/>
 - ◆ Apache: <http://www.apache.org/>
 - ◆ CIFS: <http://msdn.microsoft.com/en-us/library/aa302188.aspx>
 - ◆ Samba: <http://www.samba.org/>
 - ◆ SWAT: <http://linux.die.net/man/8/swat>
-

::Inference::

- ◆ Solaris Zones is a highly under-used and over-capable technology.

- ◆ Due to its minimal overhead architecture on Virtualization, its the best option according to me for Virtualization of Linux Boxes.
 - ◆ There is still a great scope left to be developed in this technology.
-

::Troubleshooting/Updates::

- ◆ **Problem:** The Apache mod_dav and mod_dav_svn module was failing to integrate with SVN implementation.

Solution:

Initially I was using CoolStack's Software Bundle of Apache+PHP+MySQL due to ease of use on Native-Small Zone, but found out that actually it's implementation raised the incompatibility issue. So, created a Native Big-Zone and used standard Apache release, and it worked.
