

Zones

Zone migration and shared storage - since Solaris 11.3

- In Solaris 11.3, things have changed:-
 - * Shared storage supports NFS for kernel zones.
 - * Live kernel zone migration is now available.
- The use of NFS shared storage provides an easier mechanism to implement ZOSS without needing a SAN.
- There follows an example of creating a Kernel zone on NFS shared storage:-
 - * Configuring a kernel zone *migkzone1*, and replacing the default storage with an NFS URI.
 - * Live migration of the zone.
- Firstly configure the kernel zone with *zonecfg* in the normal way, but before using *commit*, add the storage like this:-

```
zonecfg:migkzone1> select device 0
zonecfg:migkzone1:device> set storage=nfs://root:staff@cork/
volume1/migzones/migzone2_root
zonecfg:migkzone1:device> set create-size=20g
zonecfg:migkzone1:device> end
zonecfg:migkzone1> commit
zonecfg:migkzone1> exit
```

- * Note the *set create-size* option.
- * The *set storage* option is quite complex, and is explained on the next page.

Zones

Kernel Zone shared storage - since Solaris 11.3

UID/GID for access *NFS share path*

set storage=nfs://root:staff@cork/volume1/migzones/migzone2_root

server name or IP *Filename for zone root*

- When specifying the storage option details, a UID and GID are required.
- Make sure the NFS server has the equivalent user and group entries.
- Note that using a UID other than *root* may be a better option.
- The *suri(5)* manual page has details of the format required.
- Having defined the zone, with its NFS storage location, the next step is to install it.

Zones

Zone migration and shared storage - since Solaris 11.3

```
# zoneadm -z migkzone1 install -x storage-create-missing
```

```
Progress being logged to /var/log/zones/
zoneadm.20150909T154916Z.migkzone1.install
pkg cache: Using /var/pkg/publisher.
Install Log: /system/volatile/install.3036/install_log
AI Manifest: /tmp/zoneadm2583.4Layyf/devel-ai-manifest.xml
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
```

```
Installation: Starting ...
```

```
  Creating IPS image
```

```
  Installing packages from:
```

```
    solaris
```

```
      origin: http://192.168.200.198/
```

```
The following licenses have been accepted and not displayed.
```

```
Please review the licenses for the following packages post-install:
```

```
  consolidation/osnet/osnet-incorporation
```

```
Package licenses may be viewed using the command:
```

```
  pkg info --license <pkg_fmri>
```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	497/497	66552/66552	615.6/615.6	0B/s

PHASE	ITEMS
Installing new actions	91019/91019

Updating package state database	Done
---------------------------------	------

Updating package cache	0/0
------------------------	-----

Updating image state	Done
----------------------	------

Creating fast lookup database	Done
-------------------------------	------

```
Installation: Succeeded
```

```
  Done: Installation completed in 470.609 seconds.
```

- The *-x storage-create-missing* option is required to ensure that the file for the root file system in the NFS location is created.
- In this example the filename will be *migzone2_root*.
- The new file can be seen on the NFS server:

```
# ls -lh /net/cork/volume1/migzones
```

```
-rw----- 1 root  nobody   20G Sep 10 03:30 migzone2_root
```

Zones

Live Kernel Zone migration - since Solaris 11.3

- Now that we have the kernel zone installed, we can think about migrating it.
- This is a fairly simple process, but can become more complex if you use encryption during the migration.
- There are also some technical requirements, in addition to those already mentioned for kernel zones, which are:-
 - ❖ The operating system on both source and target hosts must be Solaris 11.3 or greater.
 - ❖ If migrating between SPARC systems, you must have the following firmware versions installed:
 - * T4 with at least Firmware 8.8; T5, M5, or SPARC M6 system with at least Firmware 9.5
 - ❖ Live migration source and target hosts must also have the following:
 - * All storage used by the zone must be shared and accessible by both the source and target systems.
 - * The zone configuration must be set up so that configuration is compatible and consistent on both the source and target hosts. Note that the actual configuration details will also be migrated!

Zones

Live Kernel Zone migration - since Solaris 11.3

- ❖ The migration process requires two SMF services on the target system:-
 - * The kernel zone live migration daemon, *kz-migr*.
 - * The Remote Administrative Daemon (*rad*).
- ❖ Network Time Protocol (NTP) is also required. (Ask instructor if you do not know how to set it up).
- To start the two required services on the target system (*wasp*):-

```
wasp # svcadm enable rad:remote
wasp # svcadm enable kz-migr
wasp # svcs rad:remote
STATE      STIME    FMRI
online     10:51:04  svc:/system/rad:remote
wasp # svcs kz-migr
STATE      STIME    FMRI
online     10:51:18  svc:/network/kz-migr:stream
```

- The services should be started on both source and target systems if the migration is to be carried out on both directions.
- Now we are ready to perform the migration process; well, almost - there are a couple of gotchas on the target system:-
 - * Make sure the root account is type normal, as entered in */etc/user_attr*.
 - * SSH root login is enabled.

Zones

Live Kernel Zone migration - since Solaris 11.3

- First perform a dry run on the source (*bee*):-

```
bee # zoneadm -z migkzone1 migrate -c none -n wasp
```

```
Password: *****
```

```
zoneadm: zone 'migkzone1': Importing zone configuration.
```

```
zoneadm: zone 'migkzone1': Attaching zone.
```

```
zoneadm: zone 'migkzone1': Booting zone in 'migrating-in' mode.
```

```
zoneadm: zone 'migkzone1': Checking migration compatibility.
```

```
zoneadm: zone 'migkzone1': Cleaning up.
```

```
zoneadm: zone 'migkzone1': Dry-run migration successful.
```

- Now the real thing:-

```
bee # zoneadm -z migkzone1 migrate -c none wasp
```

```
Password:
```

```
zoneadm: zone 'migkzone1': Importing zone configuration.
```

```
zoneadm: zone 'migkzone1': Attaching zone.
```

```
zoneadm: zone 'migkzone1': Booting zone in 'migrating-in' mode.
```

```
zoneadm: zone 'migkzone1': Checking migration compatibility.
```

```
zoneadm: zone 'migkzone1': Starting migration.
```

```
zoneadm: zone 'migkzone1': Suspending zone on source host.
```

```
zoneadm: zone 'migkzone1': Waiting for migration to complete.
```

```
zoneadm: zone 'migkzone1': Migration successful.
```

```
zoneadm: zone 'migkzone1': Halting and detaching zone on source host.
```

* *-c none* indicates that no encryption is required.

* SSH *authorized_keys* files can be used to prevent the need to type a password when migrating.

- This should take about 30 seconds with a training system, but maybe longer on a production zone.
- Over on the target machine (*wasp*) *zlogin* to the *migkzone1* zone to verify that it is functioning as expected.

Zones

Kernel Zones 11.2 - Final Notes

- Several other operations are available with kernel zones, including:-
 - * Cloning using the *zoneadm clone* command instead of using a unified archive.
 - * Suspending and resuming - a feature unique to kernel zones.
 - * Using encryption when migrating.
- Full details can be found, for 11.2, at:-
http://docs.oracle.com/cd/E36784_01/html/E37629/index.html
- Details for 11.3 encrypted migration are here:-
http://docs.oracle.com/cd/E53394_01/html/E54751/gpihx.html