
ORACLE®

ORACLE

SOLARIS

Simple, Flexible, Fast: Virtualization in 11.4

Oracle Tech Days, Vienna

Jan Pechanec
Oracle Solaris Virtualization Team
Mar 2018

ORACLE

Safe Harbor Statement

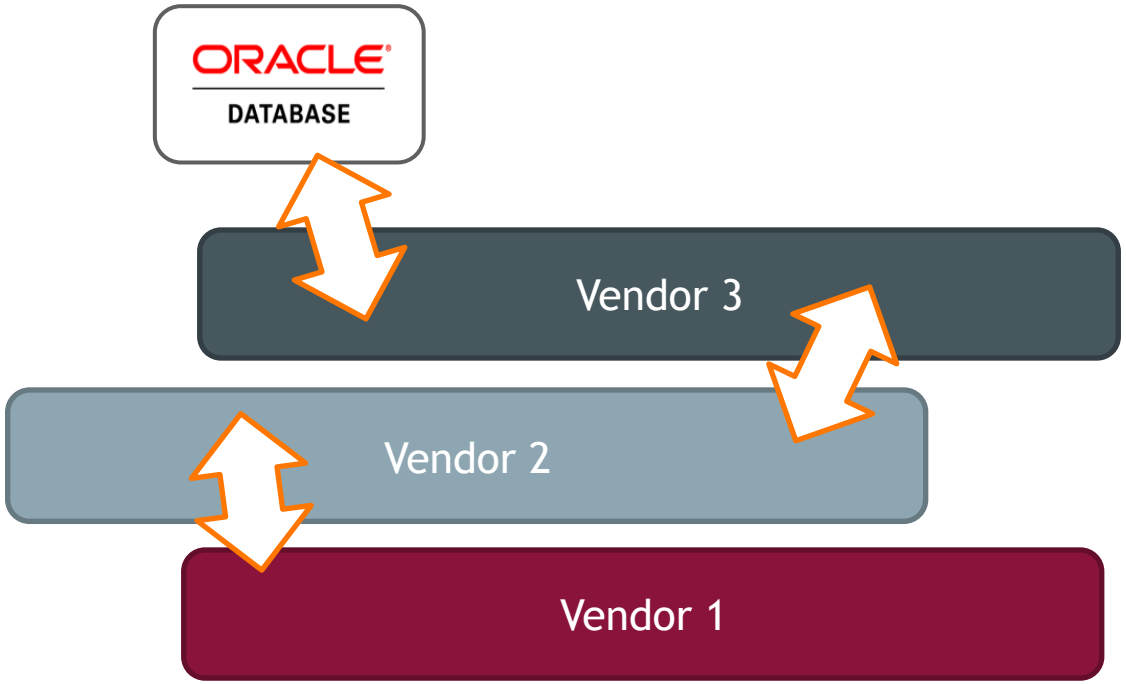
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Solaris Virtualization vs. the Competition

OS and Virtualization - Engineered Together

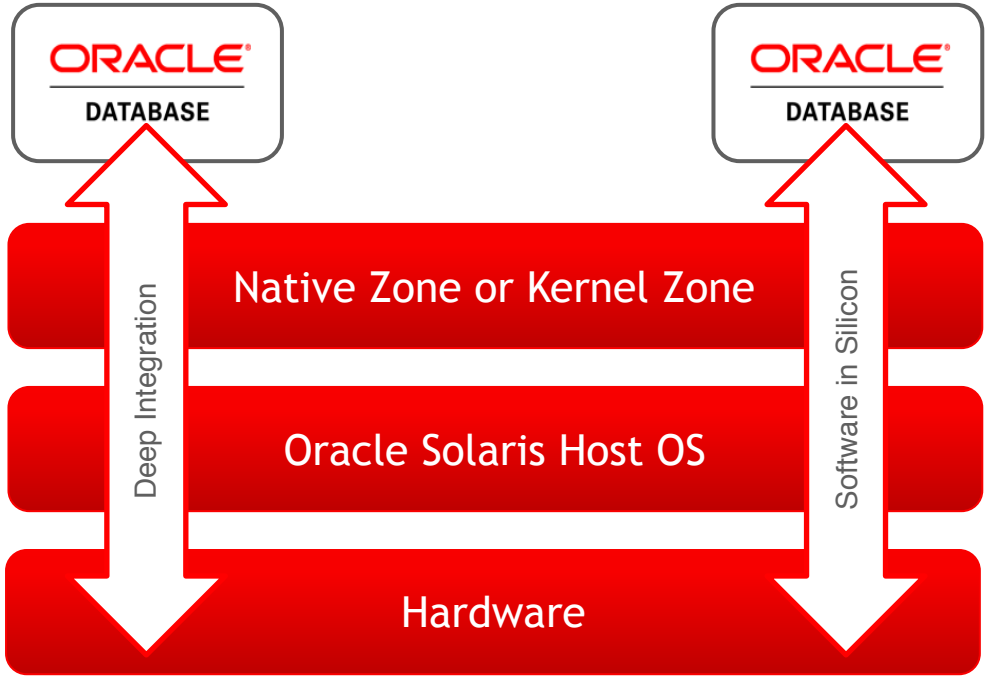
Traditional Hypervisors

Separate, isolated, slow

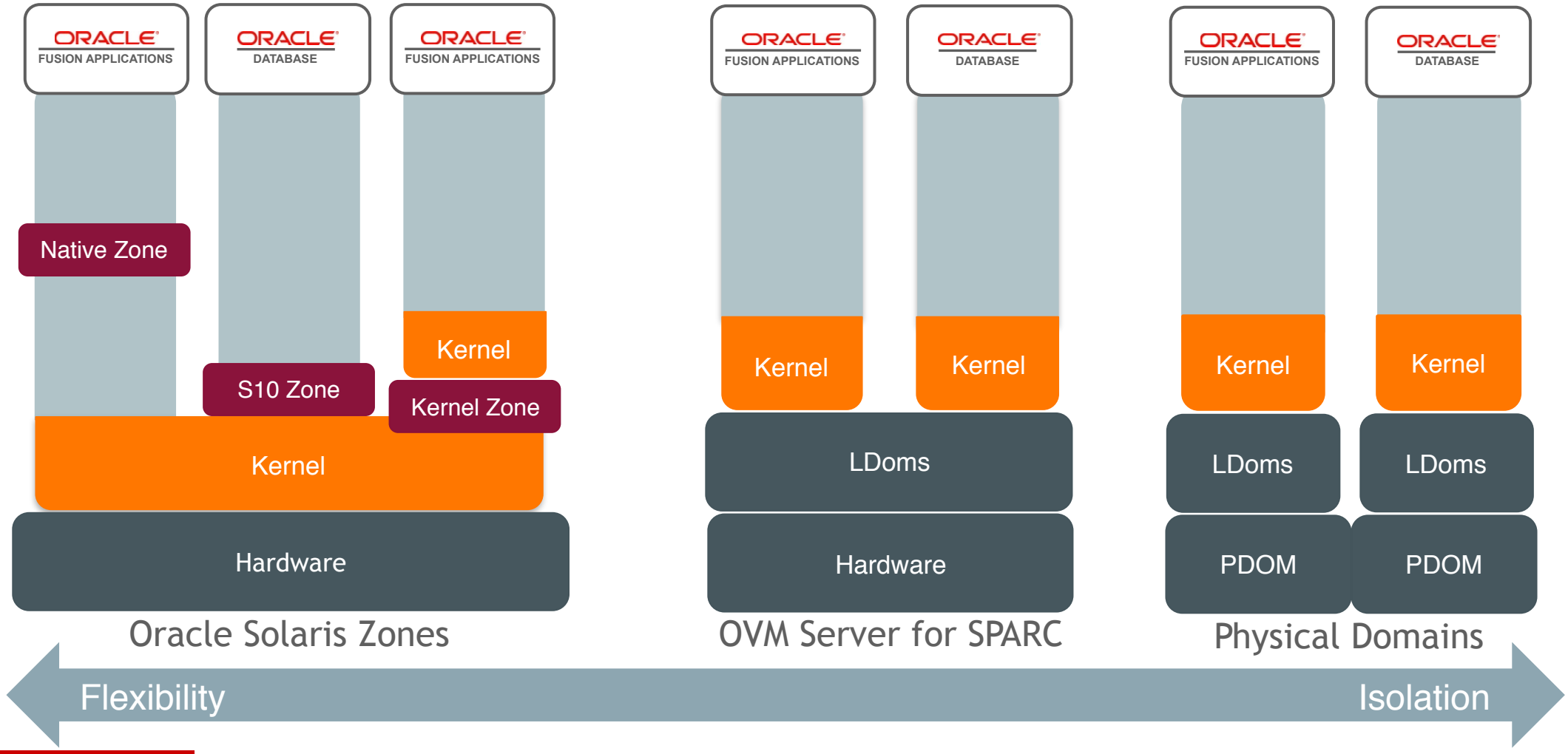


Native Zones, Kernel Zones

Engineered, performant, robust, secure

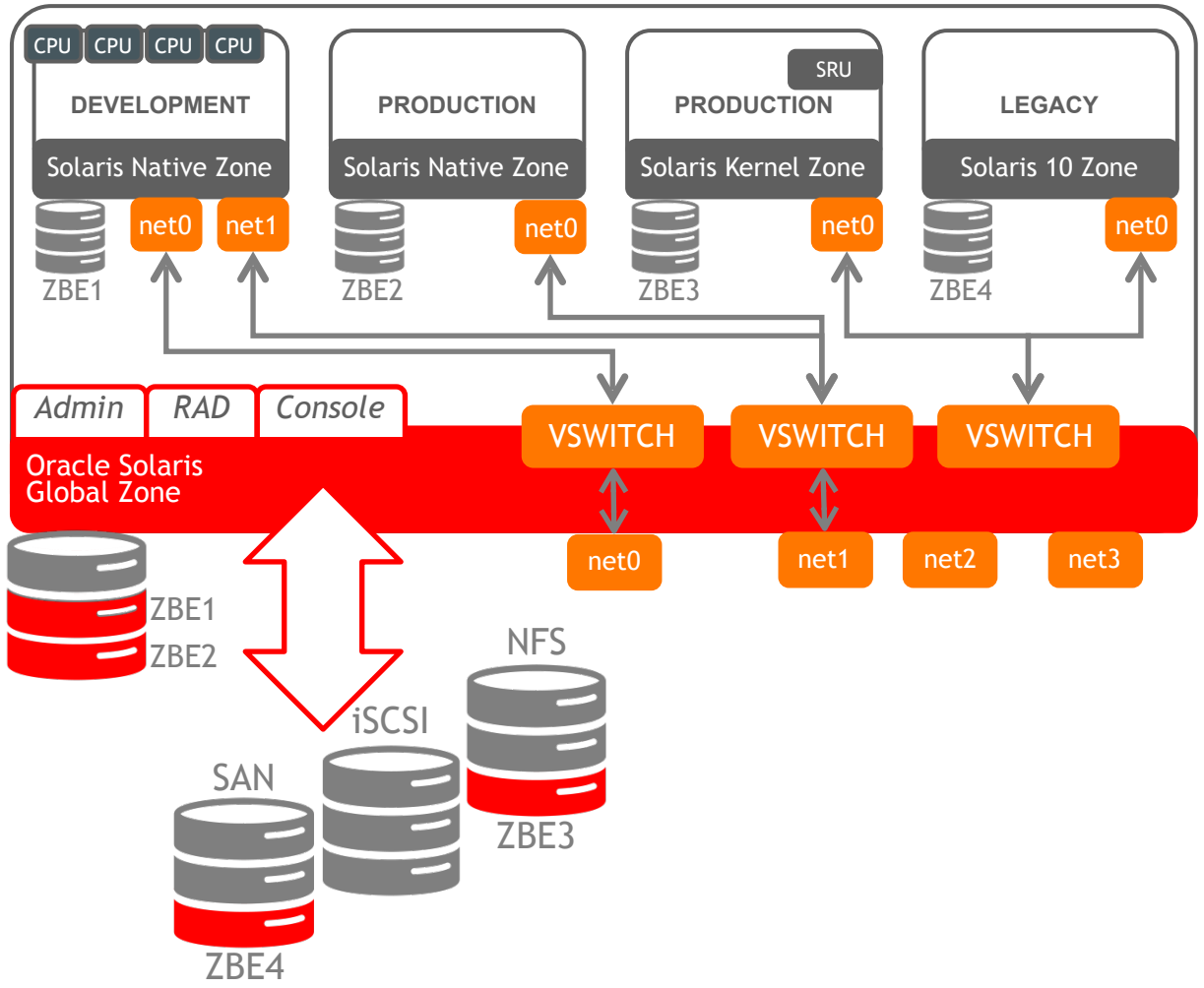


Virtualization Types



Oracle Solaris Zones

- Used by almost every Solaris customer
- Direction: the “cloud space” moving towards OS virtualization
- From 11.2 all apps should be running inside a zone of some kind



Evolution of Zones in Oracle Solaris

Solaris 10 (2005)

- First widely adopted OS-level virtualization technology
- Container technology
- Shared kernel

Solaris 11 (2012)

- Exclusive IP/Virtual networking
- Zones on shared storage
- Parallel updates
- Immutable zones
- Mature technology

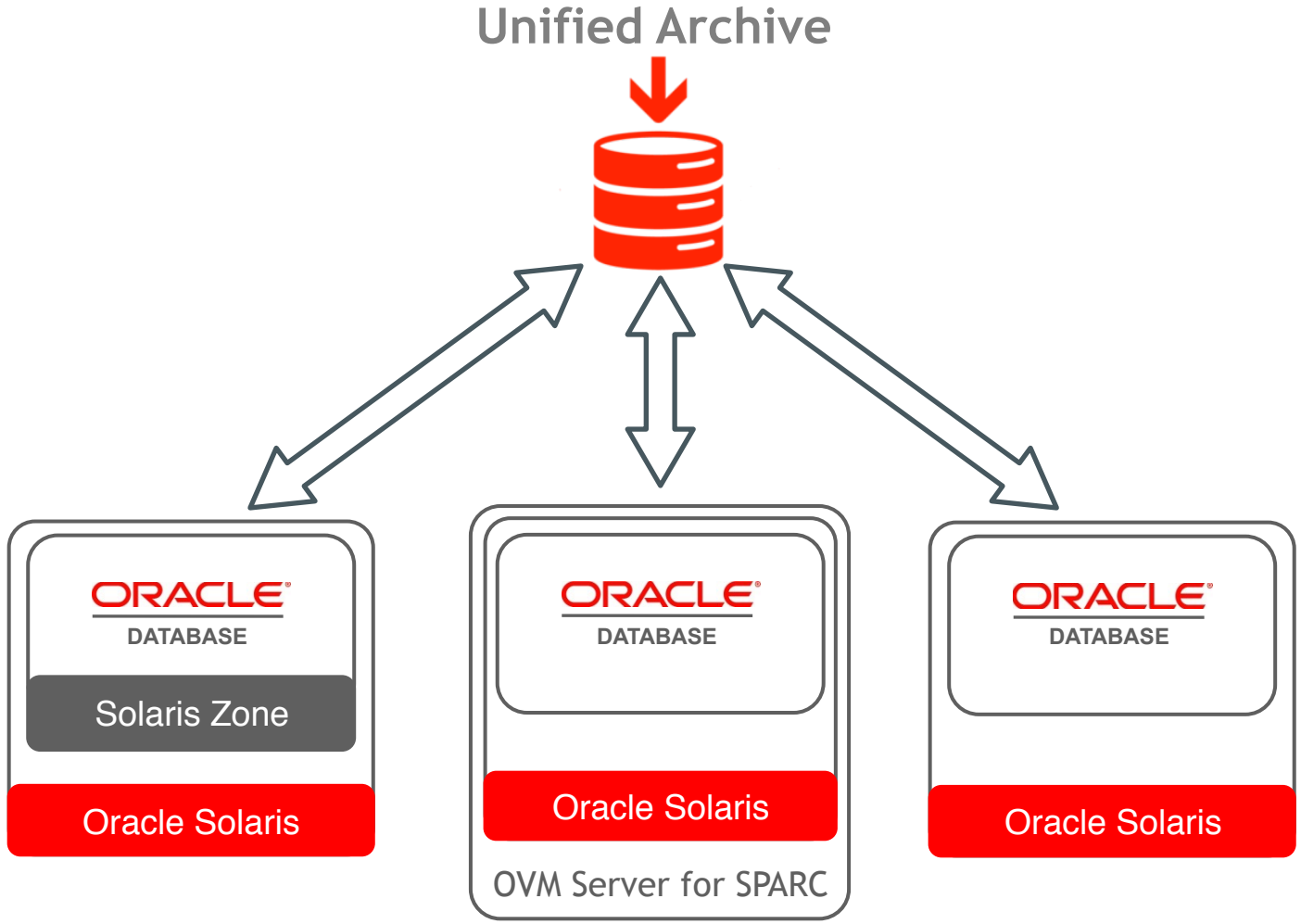
Solaris 11.2 (2014)

- New type of zone: Kernel Zone
- Live zone reconfiguration
- CMT awareness in zones
- OpenStack integration

ORACLE

Deploy And Move Between Virtualization Types

- No Virtualization “lock-in”
- Even move back to bare metal
- Move from Development to Test to Production with confidence



11.3 Highlights for Native Zones

- Virtualized clock
  - across reboots
- **`rcapd`** improvements
  - better performance
  - simpler configuration
- NPIV support
- Immutable zones enhancements
- **`zonecfg`** usability improvements

11.3: Read-Only Virtual Machines

Protect the Application Infrastructure

- Immutable Zones
  - Read only virtualization
  - Also possible with global zone and OVM Server for SPARC guest
  - Access via a trusted path
- New - Dynamic Zones setting, allows creation of zones
- Ready made “templates” via **`file-mac-profile`** property

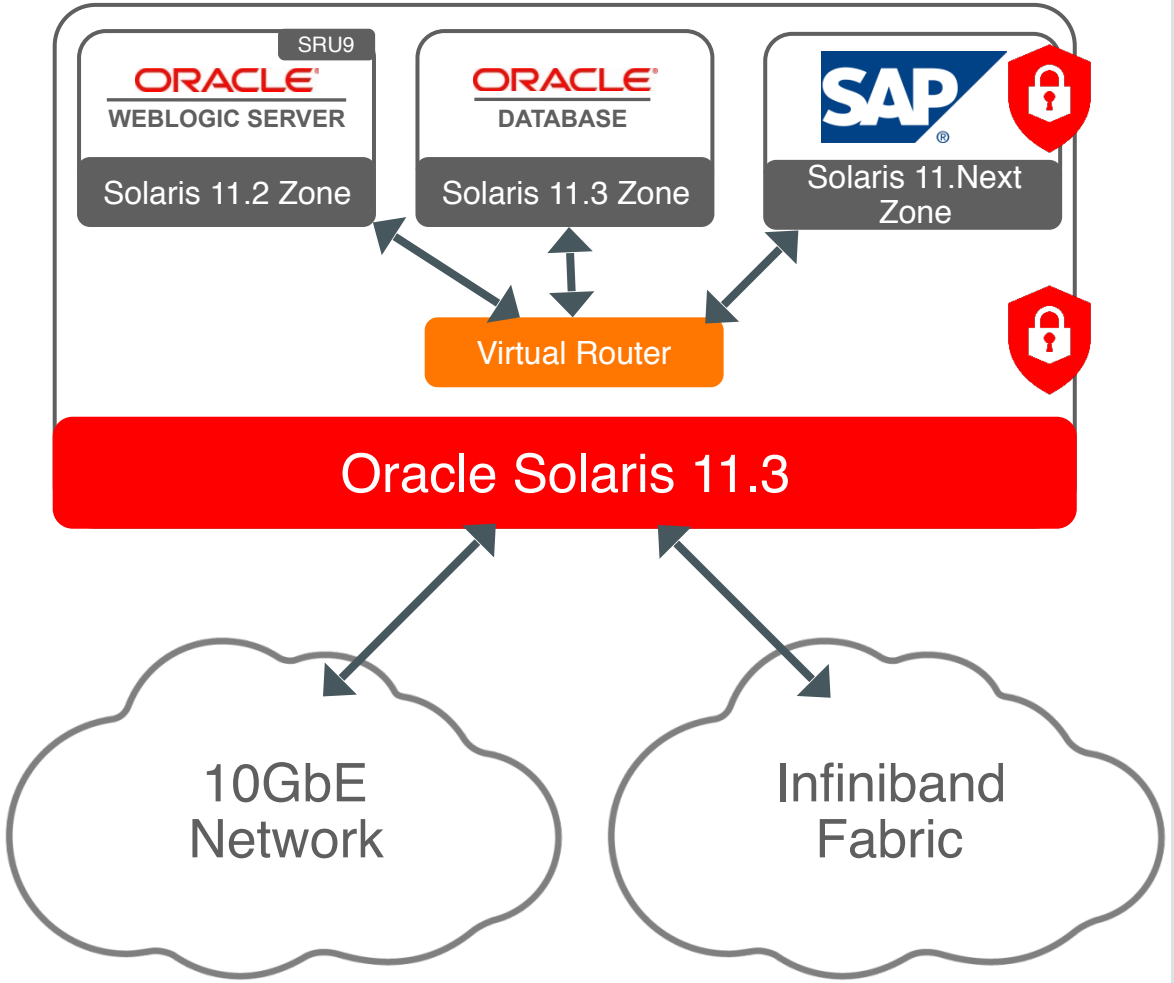
	None	Flexible	Fixed	D-Zone	Strict
/, /usr, /lb, …	Writeable	Read Only	Read Only	Read Only	Read Only
/etc	Writeable	Writeable	Read Only	Read Only	Read Only
/var	Writeable	Writeable	Writeable	Writeable	Read Only
other	Writeable	Read Only	Read Only	Read Only	Read Only
Create Zone	Yes	No	No	Yes	No

11.3: `zonecfg(8)` usability improvements

- Templates
  - To provide default values for properties
  - Just read-only zone configuration files
  - They also reside in /etc/zones
- Unique IDs for **`zonecfg`** resources
  - Previously, long selection string possibly needed when multiple instances of the same resource existed.

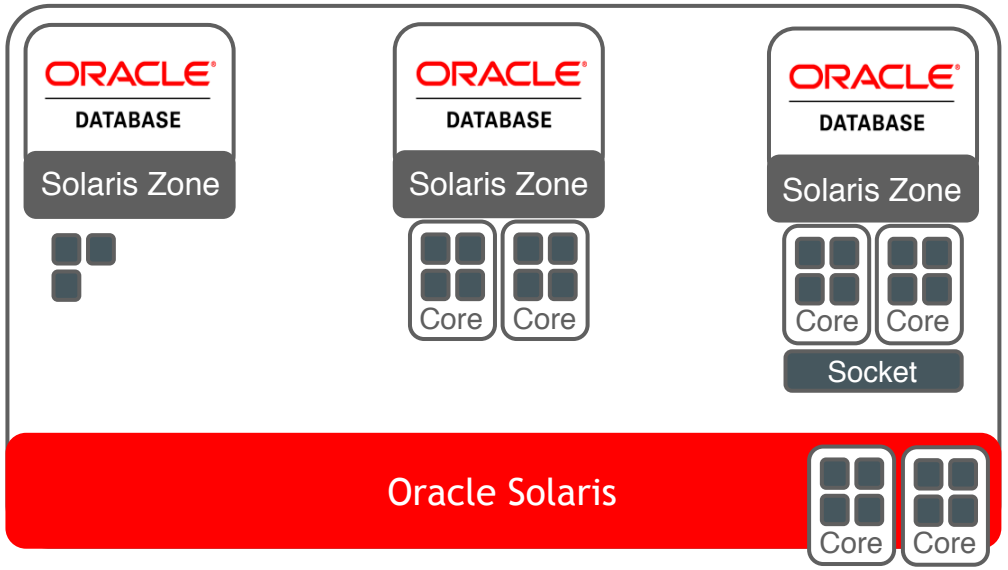
11.3 Highlights for Kernel Zones

- Live migration
  - secure; multi-threaded; migration classes for SPARC (for x86 in 11.4)
- NFS ZOSS support
  - not supported for native zones
- Live zone reconfiguration
- ADI support
- SR-IOV NIC and Infiniband support
- Verified boot
  - kernel modules verified before loading and executing



11.3: Zone Resource Management Improvements

- Assign Zone CPU resources by CPUs, Cores, and Sockets
- Applies to Zones and Resource Pools
- Use **`psrinfo –t`** to show socket/core/cpu layout
- Makes configuring and complying with license hard partition rules much easier
- Recognized license boundary



Solaris 11.4 Highlights

Main Focus Areas

- Simplified, very flexible management for all zone brands
- Extended life-cycle management
- Kernel Zones feature parity and performance

11.4: Management Improvements

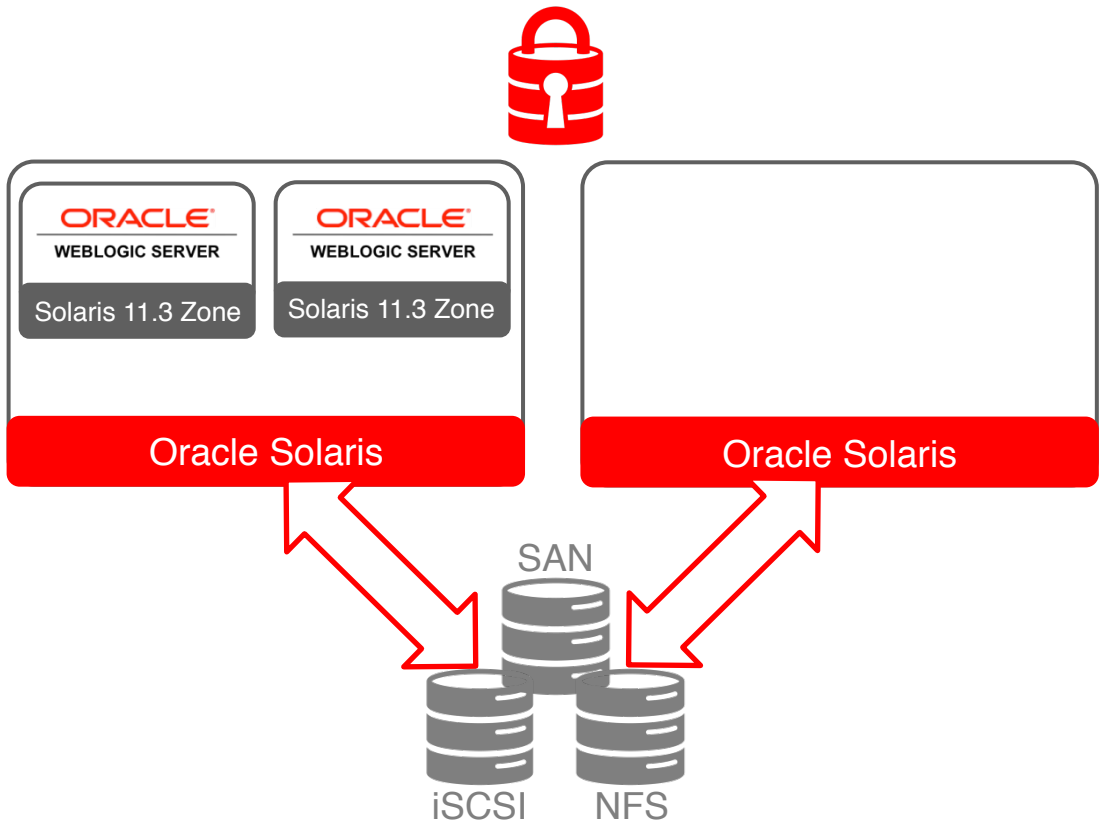
- Zones as SMF services
  - boot/shutdown throttling; boot priorities
  - FMA reporting
  - Inter-zone dependencies via goals
  - works across all zone brands
- Zone state shared across boot environments
  - previously, a zone in each BE had its own state

11.4: Management Improvements (cont.)

- Better live zone reconfiguration
  - **`virtual-cpu`** for Kernel Zones
  - datasets for native zones
- Simpler **`zonecfg info`** output
- Direct **`beadm`** of native zone BEs
  - **`beadm list -z <zone-name>`**
  - see **`fmri(7)`**, "SCHEME zbe VERSION 0" section

Secure Live Migration with Kernel Zones

- Live Migration since 11.3
- Move KZs without outage
- No downtime host maintenance
- Load balance across infrastructure
- Forward and backward compatibility for LM moves
- Migration classes for SPARC (11.3)
  - Via "**`cpu-arch`**" zone config property

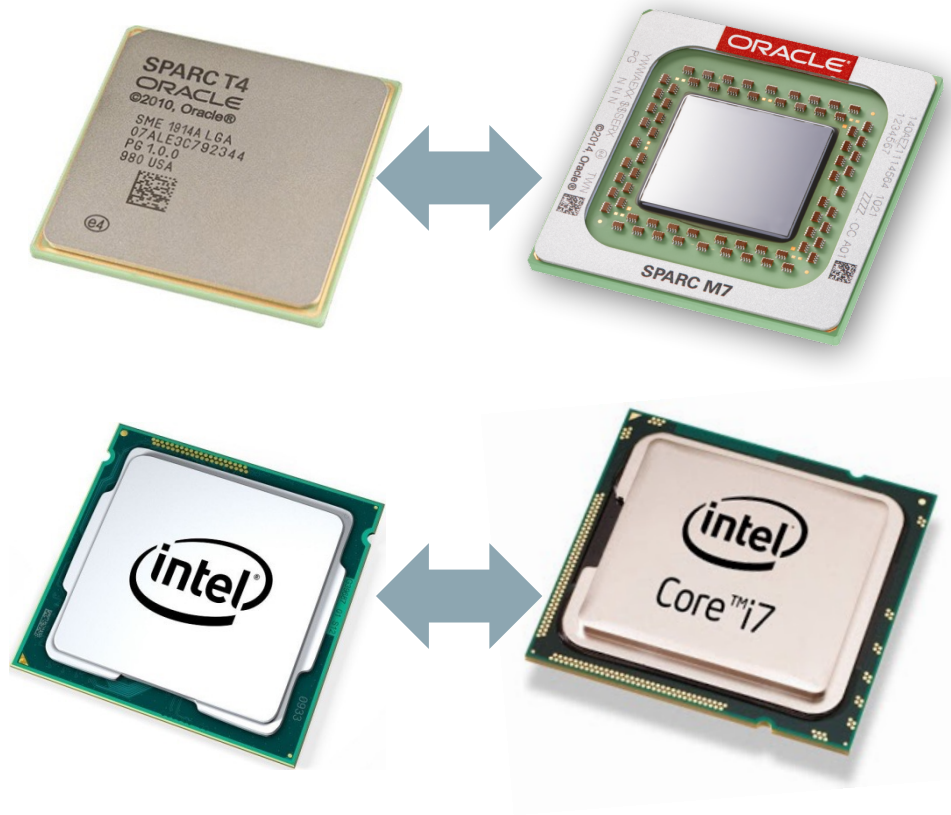


11.4: (Migration) Management Improvements

- RAD migration API
  - Finer-grained API with progress reporting
- Cold migration
  - Transfer config, automate detach/attach
  - Via **`zoneadm migrate`** (and the API)
- Improved **`virtinfo(8)`** support
  - Explicit kernel zone support info

11.4: (Migration) Management Improvements (cont.)

- Kernel zone migration flexibility improved
- Migration classes for x86
- New **`host-compatible`** setting
  - Software equivalent of **`cpu-arch`**
  - Choose feature/flexibility trade-off
  - Eg. migrating from a host supporting ADI to one that does not



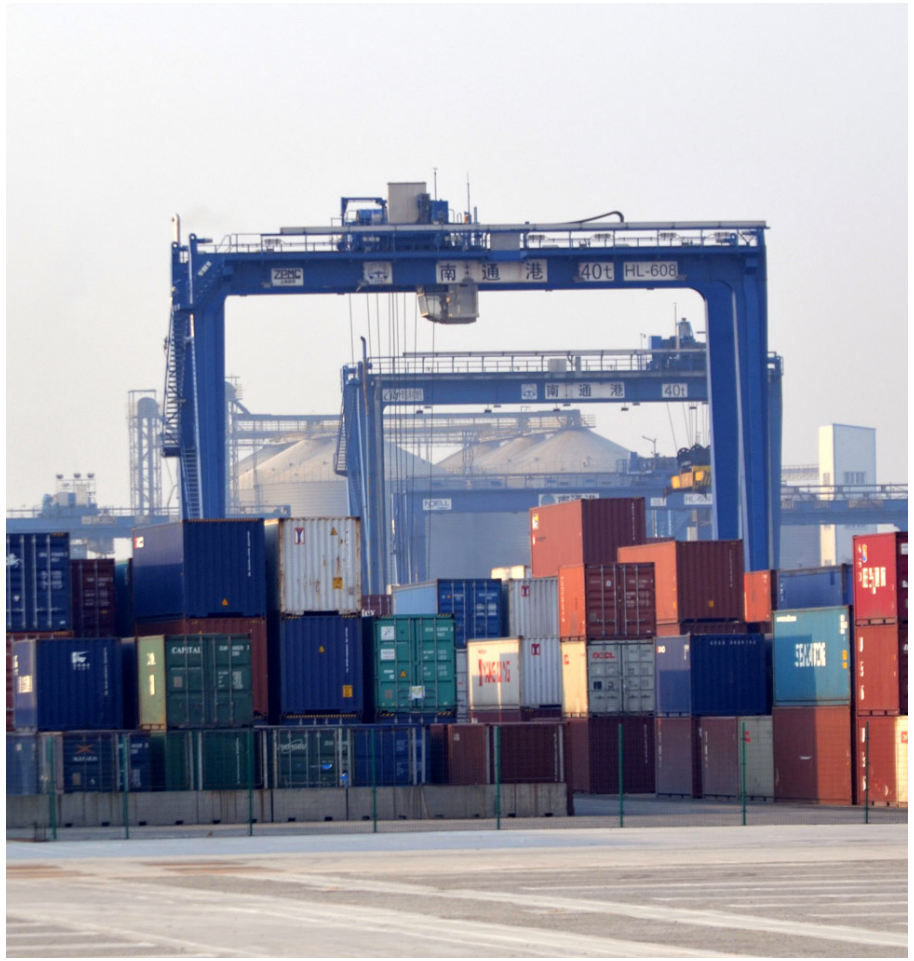
11.4: Zone Evacuation

- A single command to evacuate running Kernel Zones because of HW maintenance or failure
- Must manually specify targets
- **`sysadm(8)`** to facilitate maintenance state and evacuation capabilities
- Each KZ with a configured target is migrated to that destination
  - Installed NGZs can be evacuated as well
- Evacuate “return” to migrate back

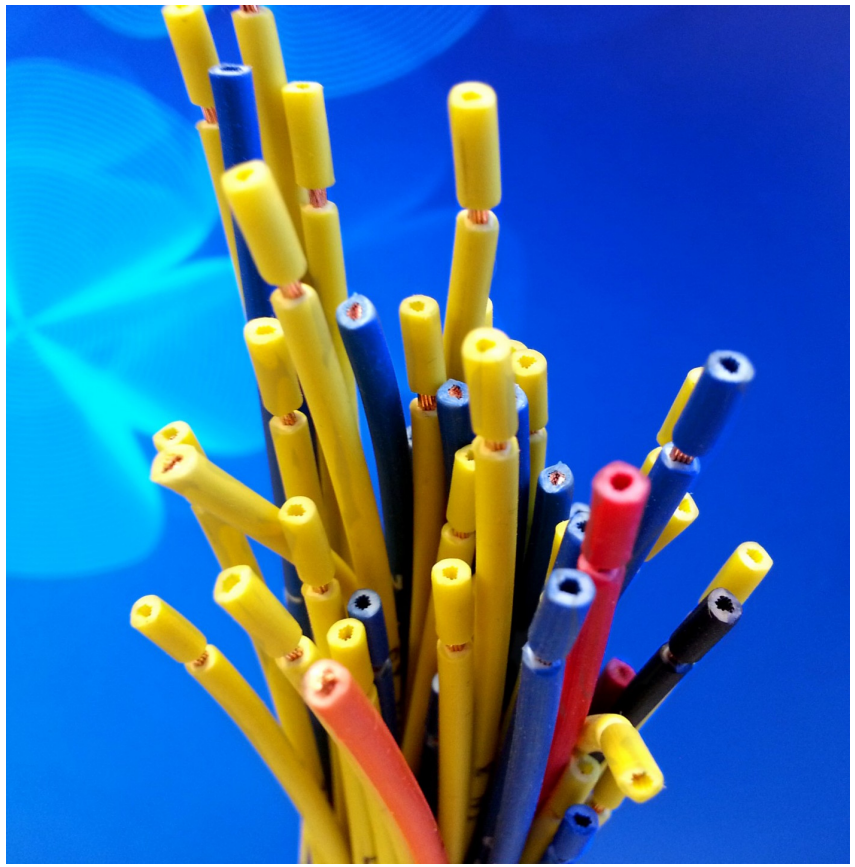
ORACLE

11.4: Storage

- SCSI-3 Persistent Group Reservation support for Kernel Zones boot disks
- NPIV support for Kernel Zones
- **`zoneadm move`**
  - Migrate native zones to/from Zones on Shared Storage (ZOSS)



11.4: Networking



- Live migration of SR-IOV Kernel Zones
  - Automatic switchover to PV interface via DLMP
  - Switch back on destination, if SR-IOV available
- VNICs over paravirtualized IPoIB datalinks for Kernel Zones
  - allows creation a NGZ inside a KZ using IPoIB PV as a lower-link

11.4: Security

- Immutable GZ management improvements
  - Avoids intermediate reboots on reconfiguration
- Per-zone RNG pools
- Comprehensive auditing of kernel (and native) zone operations
  - Auditing the lifecycle of NGZs/KZs
  - look for **`AUE_zone_*`** events

11.4: Resource management

- Per-zone statistics (kstats)
  - subset of system CPU stats
  - also memory and usage cap stats
- Improved **`zonestat(1)`** scalability
- Multi-CPU binding for projects
- Side-by-side pools and psets
- Allocation by cores/sockets

Demos

- Throughout the presentation, demos were shown.
- Those were collected and put in the following slides with short explanations for each.
- They are not meant to be complete due to limited space.

Demos (cont.)

- To show how to move between Solaris virtualization types, we created a solaris branded Zone clone archive and used it to install a Kernel Zone on NFS (see `suri(7)` for more on storage URIs).

```
# archiveadm create -z tzone1 --root-only /data/uar/tzone1.uar

# zonecfg -z kz-nfs info device
device:
        storage: nfs://pechanec:staff@bjork/data/kz-nfs.image
        create-size: 16G
        id: 0
        bootpri: 0

# zoneadm -z kz-nfs install -x storage-create-missing -a /data/uar/tzone1.uar
```

Demos (cont.)

- To assign Zone CPU resources by Cores and Sockets, you can do it like this:

```
# psrinfo -t
socket: 0
  core: 0
    cpus: 0,28
  core: 1
    cpus: 1,29
  core: 2
    cpus: 2,30
  core: 3
…….

# zonecfg -z tzone1 "add dedicated-cpu; set cores=0-2,4; end"
```

Demos (cont.)

- After the hosts were configured for the Live Migration, the installed KZ could be migrated to another host like this:

```
# zoneadm -z kz-nfs migrate ssh://pechanec@bjork
zoneadm: zone 'kz-nfs': Using existing zone configuration on destination.
zoneadm: zone 'kz-nfs': Attaching zone.
zoneadm: zone 'kz-nfs': Booting zone in 'migrating-in' mode.
zoneadm: zone 'kz-nfs': Checking live migration compatibility.
zoneadm: zone 'kz-nfs': Performing initial copy (total 4096MB).
zoneadm: zone 'kz-nfs': 0.00% done: 0MB copied @ 0.0MB/s, skipped 0MB
zoneadm: zone 'kz-nfs': 36.29% done: 512MB copied @ 102.4MB/s, skipped 974MB
zoneadm: zone 'kz-nfs': 75.19% done: 1024MB copied @ 102.4MB/s, skipped 2055MB
zoneadm: zone 'kz-nfs': 98.01% done: 1600MB copied @ 115.2MB/s, skipped 2414MB
zoneadm: zone 'kz-nfs': 100.00% done: 1681MB copied @ 16.2MB/s, skipped 2414MB
zoneadm: zone 'kz-nfs': Performing copy of recently modified memory.
zoneadm: zone 'kz-nfs': Suspending zone on source host.
zoneadm: zone 'kz-nfs': Waiting for migration to complete.
zoneadm: zone 'kz-nfs': Migration successful.
zoneadm: zone 'kz-nfs': Halting and detaching zone on source host.
```

Demos (cont.)

- Direct beadm with solaris branded Zones

```
# beadm list -z tzone1
FMRI                    Parent BE FMRI      Flags Mountpoint                Space
---------------------- ------------------- ----- ------------------------- -------
zbe://tzone1/solaris-0  -                   O     -                         271.50K
zbe://tzone1/solaris-1  be://rpool/st_012_1 -     -                         130.09M
zbe://tzone1/solaris-10 be://rpool/st_012_1 -     -                         157.50K
zbe://tzone1/solaris-12 be://rpool/st_012_1 NR    /system/zones/tzone1/root 1.97G
zbe://tzone1/solaris-2  -                   RO    -                         274.50K
zbe://tzone1/solaris-8  be://rpool/st_012_1 -     -                         523.00K
```

- To remove orphaned ZBEs (O), you can re-attach with an -x option:

```
# zoneadm -z tzone1 detach
# zoneadm -z tzone1 attach -x destroy-orphan-zbes
```

Demos (cont.)

- Solaris branded Zones can be cold migrated now, if they are installed on shared storage. Cold means in the installed state.

```
# zoneadm -z tzone1 list -v
  ID NAME             STATUS      PATH                         BRAND      IP
   - tzone1           installed   /system/zones/tzone1         solaris    excl

# zonecfg -z tzone1 info rootzpool
rootzpool:
        storage: iscsi://xstorage/luname.naa.600144f0dbf8af1900005582f1c90007

# zoneadm -z tzone1 migrate ssh://jpechane@bjork
zoneadm: zone 'tzone1': Using existing zone configuration on destination.
zoneadm: zone 'tzone1': Attaching zone.
zoneadm: zone 'tzone1': Migration successful.
```

Demos (cont.)

- Explicit kernel zone support info in `virtinfo(8)` may look like the following.

```
kz# virtinfo -c unsupported get status
NAME        CLASS       PROPERTY VALUE
kernel-zone unsupported status   not supported in kernel-zone

gz# virtinfo -c unsupported get status
NAME        CLASS       PROPERTY VALUE
kernel-zone unsupported status   cannot load the zvmm kernel module
```

Demos (cont.)

- To evacuate zones, you need to set the target first.

```
# svccfg -s system/zones/zone:evac1 listprop evacuation/target
evacuation/target astring     ssh://root@bjork
# : …set evacuation/target for other zones as well

# zoneadm list -cv
  ID NAME             STATUS      PATH                         BRAND      IP
   0 global           running     /                            solaris    shared
  44 evac1            running     -                            solaris-kz excl
  45 kz-nfs           running     -                            solaris-kz excl
   - tzone1           installed   /system/zones/tzone1         solaris    excl
   - evac3            installed   -                            solaris-kz excl
   - evac4            configured  -                            solaris-kz excl
```

Demos (cont.)

- Then, start the maintenance mode, and evacuate running zones (default), and installed zones as well (-a).

```
# sysadm maintain -s -m "Evacuation demo"
# sysadm evacuate -av
sysadm: preparing 4 zone(s) for evacuation ...
sysadm: initializing migration of tzone1 to bjork ...
sysadm: initializing migration of evac3 to bjork ...
sysadm: initializing migration of kz-nfs to bjork ...
sysadm: initializing migration of evac1 to bjork ...
sysadm: evacuating 4 zone(s) ...
sysadm: migrating tzone1 to bjork ...
sysadm: migrating evac1 to bjork ...
sysadm: migrating evac-nfs to bjork ...
sysadm: migrating evac3 to bjork ...
sysadm: evacuation completed successfully.
sysadm: evac-nfs: evacuated to ssh://root@bjork
sysadm: evac1: evacuated to ssh://root@bjork
sysadm: evac3: evacuated to ssh://root@bjork
sysadm: tzone1: evacuated to ssh://root@bjork
```

Demos (cont.)

- After the maintenance is done, end it and return the zones.

```
# sysadm maintain -e
# sysadm evacuate -rav
sysadm: preparing 4 zone(s) for return ...
sysadm: initializing return of tzone1
sysadm: initializing return of evac3
sysadm: initializing return of evac-nfs
sysadm: initializing return of evac1
sysadm: returning 4 zone(s) ...
sysadm: migrating evac3
sysadm: migrating evac-nfs
sysadm: migrating tzone1
sysadm: migrating evac1
sysadm: return completed successfully.
sysadm: evac-nfs: returned
sysadm: evac1: returned
sysadm: evac3: returned
sysadm: tzone1: returned
```

Demos (cont.)

- We can move solaris branded Zones to and out of shared storage.

```
# zoneadm -z zlocal move \
    -p iscsi://10.99.99.75/luname.naa.600144f0dbf8af19000055d2653b0001
Configured storage resource(s) from:
        iscsi://10.99.99.75/luname.naa.600144f0dbf8af19000055d2653b0001
Created zpool: zlocal_rpool
Copying from rpool/VARSHARE/zones/zlocal to zlocal_rpool: please be patient
# zonecfg -z zlocal info rootzpool
rootzpool:
        storage: iscsi://10.99.99.75/luname.naa.600144f0dbf8af19000055d2653b0001
# : now move it back out of shared storage back to a local dataset
# zoneadm -z zlocal move -x remove-rootzpool -x force-storage-destroy-all
The following ZFS file system(s) have been created:
    rpool/VARSHARE/zones/zlocal
Copying from zlocal_rpool to rpool/VARSHARE/zones/zlocal: please be patient
Exported zpool: zlocal_rpool
Unconfigured storage resource(s) from:
        iscsi://10.99.99.75/luname.naa.600144f0dbf8af19000055d2653b0001
```

Any Questions?

Integrated Cloud

Applications & Platform Services


ORACLE®