7

Advanced Administration of Zones

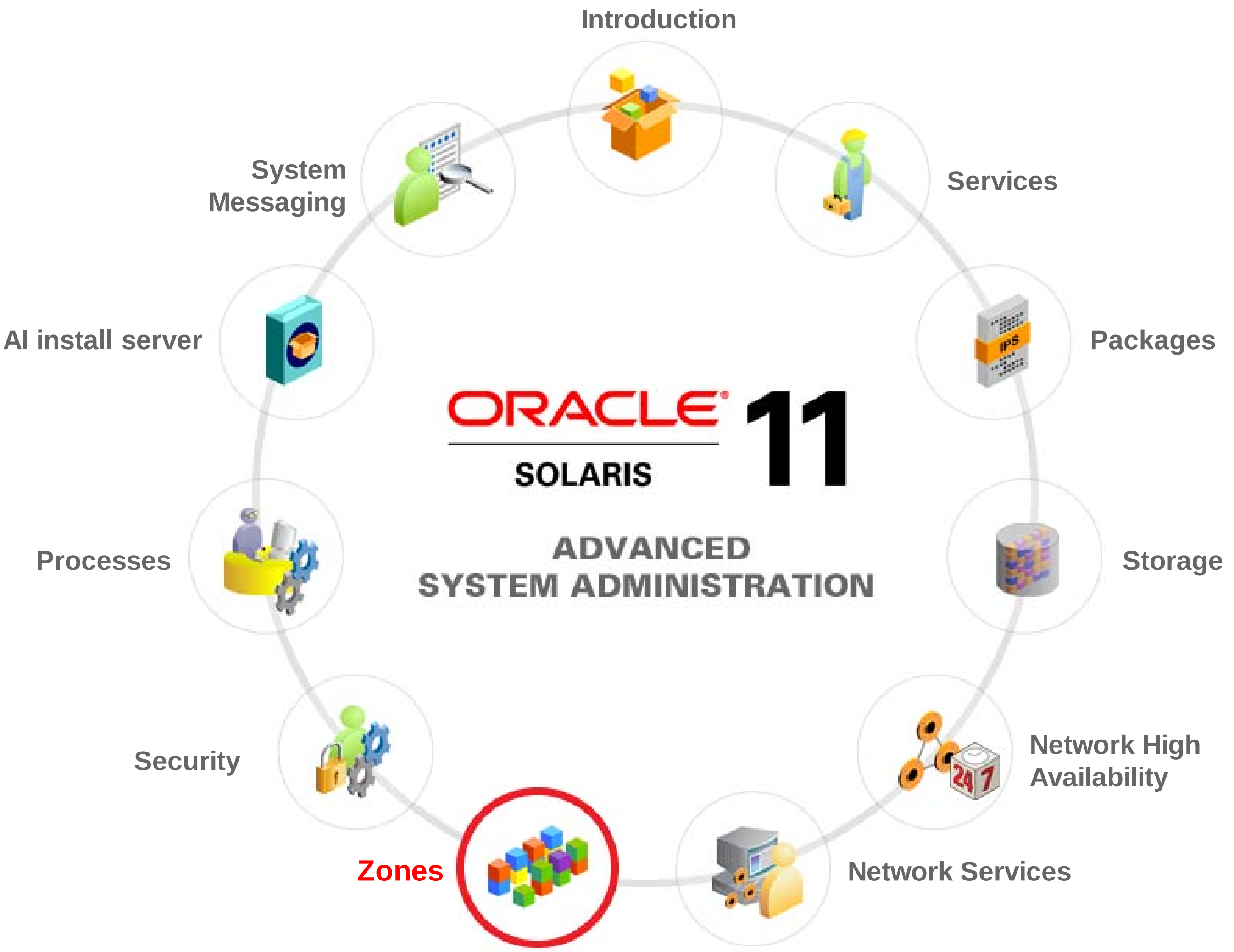
ORACLE

Objectives

After completing this lesson, you should be able to:

- Explain the fundamentals of Oracle Solaris 11 zones
- Manage system resources in a zone
- Migrate Oracle Solaris 10 systems
- Configure kernel zones
- Work with Unified Archives

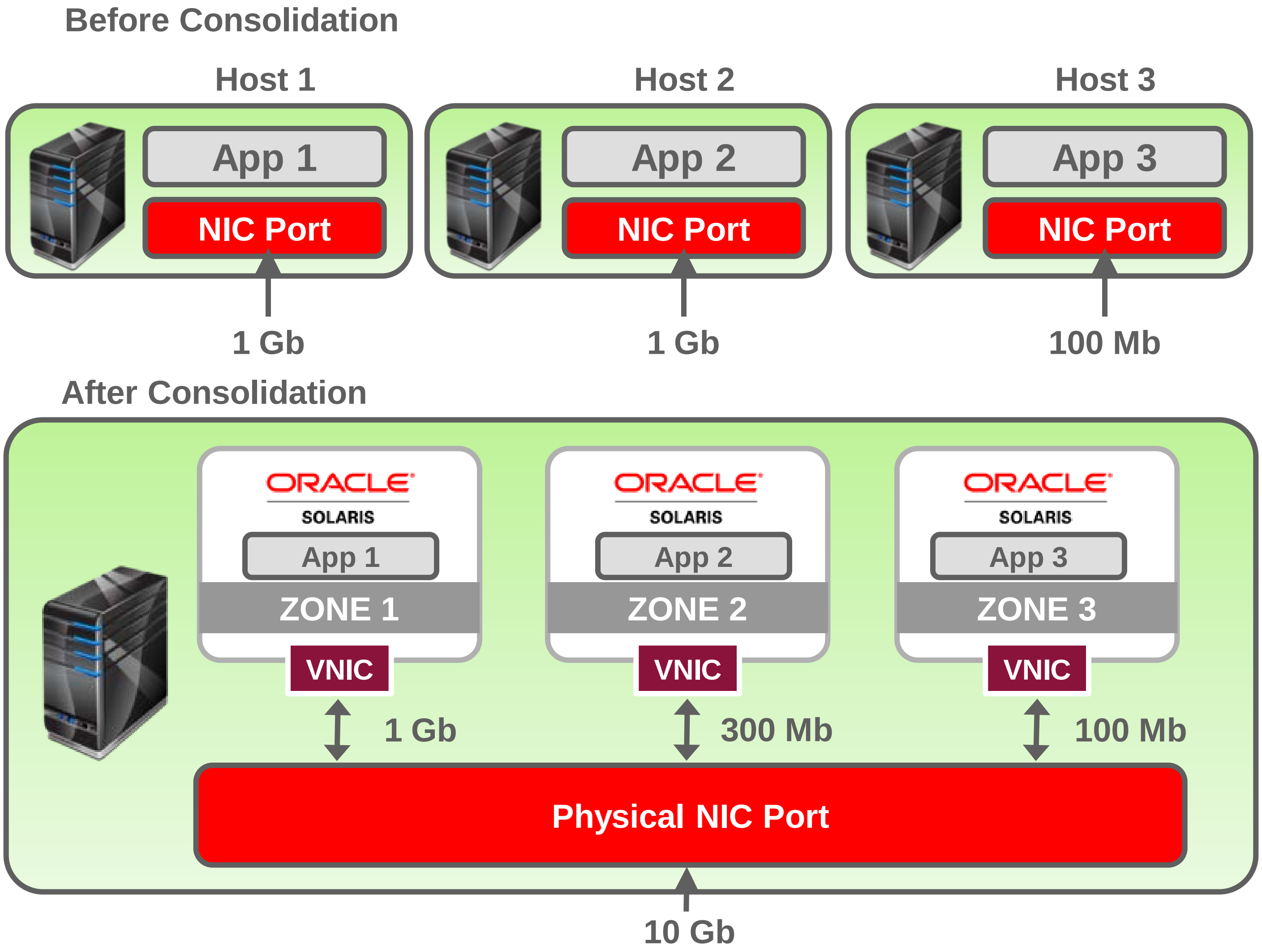
Workflow Orientation



Agenda

- Explaining the fundamentals of Oracle Solaris 11 zones
- Managing system resources in a zone
- Migrating Oracle Solaris 10 systems
- Configuring Kernel Zones
- Working with Unified Archives

Oracle Solaris Zones



Oracle Solaris Zone Brands

Oracle Solaris 11.3 supports the following zone brands:

- Oracle Solaris Zones (`solaris`):
  - Introduced in Oracle Solaris 10
  - Commonly called nonglobal zones, native, or just zones
  - A complete runtime environment for applications
- Oracle Solaris 10 Zones (`solaris10`):
  - Known as `solaris10` branded, nonglobal zones
  - Uses the BrandZ technology to run Oracle Solaris 10 applications on the Oracle Solaris 11 OS
- Oracle Solaris Kernel Zones (`solaris-kz`):
  - Introduced in Oracle Solaris 11.2
  - A feature enhancement that extends the functionality of the existing zones technology

Oracle Solaris 11 Zones: Features

Zones Feature	Description
Solaris 10 Zones	Solaris 10 Zones host Solaris 10 user environments inside zones on Oracle Solaris 11.
Boot environments for zones	Boot environments are integrated with Oracle Solaris Zones.
IPS integration	Oracle Solaris Zones have been integrated with the new IPS package management tools in Oracle Solaris 11.
Zone resource monitoring	Oracle Solaris 11 features a robust zones resource monitoring utility, `zonestat`.
Delegated administration	Delegate common zone administration tasks for specific zones to different administrators by using Role-Based Access Control.
Zones on shared storage	Configure, install, and run Oracle Solaris Zones hosted directly on arbitrary storage device objects, such as Fibre Channel, iSCSI targets, or kernel zone storage devices.
NFS server in nonglobal zones	Nonglobal zones now support NFS servers.
`unavailable` zone state	This state indicates that the zone has been installed, but cannot be verified, made ready, booted, attached, or moved.

Oracle Solaris 11 Zones: Features

Zones Feature	Description
Live zone reconfiguration	This feature enables configuration changes in a running nonglobal zone or kernel zone without the need to reboot, eliminating down time in service availability within the zone when configuration changes are made.
Live migration for kernel zones	This feature enables you to migrate a running kernel zone to a target system, without causing an outage.
Datalinks creation in nonglobal zones from the global zone	This feature enables administrators to dynamically create VNICs, VLANs, and IP-over-InfiniBand partitions directly in the nonglobal zone's namespace from the global zone.
Virtualized clocks	Set time values in nonglobal zones that are different from the value in the global zone. If you change the time in the global zone, the nonglobal zone time is offset by the same amount.

Agenda

- Explaining the fundamentals of Oracle Solaris 11 zones
- Managing system resources in a zone
- Migrating Oracle Solaris 10 systems
- Configuring Kernel Zones
- Working with Unified Archives

Importance of Resource Management

With the Oracle Solaris resource management facility, you can manage your system’s application workloads by:

- Restricting access to a specific resource
- Offering resources to workloads on a preferential basis
- Isolating workloads from each another
- Preventing an application from consuming resources indiscriminately
- Changing an application’s priority based on external events

Zone Resource Management

- Align resource management control boundaries with those of the zones.
- Zone resources that can be controlled include:
  - Resource pools or assigned CPUs
  - Resource controls
  - Scheduling classes

Managing System Resource Allocation to a Zone

System resource allocation to a zone can be controlled by:

- **Resource pools:** Used primarily to manage CPU usage
- **Resource capping:** Used to regulate physical memory consumption
- **Process scheduling:** Used to control the allocation of available CPU resources to processes

Resource Pool Allocation

- SMF supports two resource pool services:
  - Default resource pool service `svc:/system/pools:default`
  - Dynamic resource pool service `svc:/system/pools/dynamic:default`
- Resource pool services are disabled by default.
- To allocate a resource pool to a zone, you must:
  - Enable the two resource pool services
  - Create a pool configuration file and save it in the default configuration file `/etc/pooladm.conf`
  - Modify the pool configuration file to specify a subset of the system’s processors that should be dedicated to a zone
  - Bind the resource pool to the zone

How Resource Pools Work

- Oracle Solaris software boots.
- The initialization SMF service checks for the `/etc/pooladm.conf` file.
- If the file exists, `pooladm` makes the configuration the active pools’ configuration.
- The system creates the dynamic configuration.
- Resources are allocated and monitored by the pools’ resource controller (`poold`).

Memory Resource Capping

- Resource capping is controlled by the `rcapd` daemon.
- The `rcapd` daemon repeatedly samples the resource utilization of projects that have physical memory caps.
- The sampling interval is specified by the administrator.
- When physical memory utilization thresholds are exceeded, the daemon reduces the resource consumption with memory caps.

Specifying Resource Capping Within a Zone

- The `capped-memory` resource sets limits for physical, swap, and locked memory.
- At least one limit must be set.
- The `rcapd` daemon and `rcap` service must be up and running.

Allocating System Resources to a Zone

To allocate system resources to a zone, perform the following:

- Specify a subset of the system's processors that should be dedicated to a zone while it is running.
- Limit the amount of CPU resources that can be consumed by a zone.
- Control the allocation of available CPU resources among zones, based on their importance.
- Limit the amount of physical memory.

Live Zone Reconfiguration

- It allows you to reconfigure or report on the live configuration of `solaris` brand and `solaris10` brand nonglobal zones while the zones are running, without requiring to reboot.
- You can use this feature to make the following changes in running nonglobal zones:
  - Change resource controls
  - Change network configuration
  - Change the CPU resource pool
  - Add or remove file systems
  - Add or remove virtual and physical devices

Determining a Zone’s Resource Utilization

To determine a zone’s resource utilization, use the `zonestat` utility.

```
# zonestat -r summary 5
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
SUMMARYInterval: 3, Duration: 0:00:15
SUMMARY                   Cpus/Online: 1/1   PhysMem: 1023M  VirtMem: 2047M
                    ---CPU----  --PhysMem-- --VirtMem-- --PhysNet--
               ZONE  USED %PART  USED %USED  USED %USED PBYTE %PUSE
            [total]  1.00  100%  658M 64.3%  839M 41.0%  1431 0.00%
           [system]  0.18 18.9%  373M 36.5%  521M 25.4%     -     -
           choczone  0.68 68.8% 44.0M 4.30% 49.6M 2.42%     0 0.00%
             global  0.11 11.0%  133M 13.0%  167M 8.16%  1431 0.00%
                 QA  0.00 0.40% 53.5M 5.23% 50.3M 2.46%     0 0.00%
        grandmazone  0.00 0.81% 53.3M 5.21% 51.4M 2.51%     0 0.00%


...
...
...
```

Determining a Zone’s Kernel File System Statistics

```
# fsstat -z s10 -z s10u9 zfs tmpfs
  new name   name  attr attr lookup rddir  read  read write write
 file remov  chng   get  set    ops   ops   ops bytes   ops bytes
   93    82     6  163K  110   507K   148 69.7K 67.9M 4.62K 13.7M zfs:s10
  248   237   158  188K  101   612K   283 70.6K 68.6M 4.71K 15.2M zfs:s10u9
12.0K 1.90K 10.1K 35.4K   12  60.3K     4 25.7K 29.8M 36.6K 31.0M tmpfs:s10
12.0K 1.90K 10.1K 35.6K   14  60.2K     2 28.4K 32.1M 36.5K 30.9M tmpfs:S10u9
# fsstat -A -Z zfs tmpfs
  new name   name  attr  attr lookup rddir  read  read write write
 file remov  chng   get   set    ops   ops   ops bytes   ops bytes
 360K 1.79K 20.2K 4.20M 1.02M  25.0M  145K 5.42M 2.00G 1.07M 8.10g zfs
 359K 1.48K 20.1K 4.04M 1.02M  24.5M  144K 5.31M 1.88G 1.06M 8.08G zfs:global
   93    82     6 74.8K   107   250K   144 54.8K 60.5M 4.61K 13.7M zfs:s10
  248   237   158 90.2K   101   336K   283 53.0K 58.3M 4.71K 15.2M zfs:s10u9
60.0K 41.9K 17.7K  410K   515   216K   426 1022K 1.02G  343K  330M tmpfs
49.4K 38.1K 11.0K  366K   489   172K   420  968K  979M  283K  273M tmpfs:global
5.28K 1.90K 3.36K 21.9K    12  21.7K     4 25.7K 29.8M 29.9K 28.3M tmpfs:s10
5.25K 1.90K 3.34K 22.1K    14  21.6K     2 28.4K 32.1M 29.8K 28.2M tmpfs:s10u9
```

Allocating and Managing System Resources in a Zone

This section covers allocating and managing the following:

- CPU resources with resource pools
- Physical memory resources with resource capping

Allocating and Managing CPU Resources with Resource Pools

To manage CPU consumption in a zone, perform the following steps by using resource pools:

1. Enabling services for resource pools
2. Configuring a persistent resource pool
3. Binding the zone to a persistent resource pool
4. Removing the resource pool configuration

Enabling Services for Resource Pools

To activate the resource pool services, run `svcadm enable -r pools/dynamic`.

```
# svcadm enable -r pools/dynamic
```

To verify that the service pools and the `poold` daemon are up, run `svcs *pools*` and `pgrep -lf poold`, respectively.

```
# svcs *pools*
STATE          STIME    FMRI
online         16:08:10 svc:/system/pools:default
online         16:08:11 svc:/system/pools/dynamic:default
# pgrep -lf poold
 2283 /usr/lib/pool/poold
```

Configuring a Persistent Resource Pool

To create the pool configuration file, use `pooladm -s`.

```
# pooladm –s
```

To verify that the file has been created, use `ls -l /etc/pool*`.

```
# ls -l /etc/pool*
-rw-r--r-- 1 root root 1298 Dec 14 16:13 /etc/pooladm.conf
# file /etc/pooladm.conf
/etc/pooladm.conf:      XML document
```

Displaying the Resource Pool Configuration File

To display the resource pool configuration file, use `poolcfg -c info`.

```
# poolcfg -c info
system default
        string  system.comment
        int     system.version 1
        boolean system.bind-default true
        string  system.poold.objectives wt-load
```

<Complete output presented in the Notes>

Modifying the Resource Pool Configuration File

To create the pset, use poolcfg -c 'create pset pset_*psetname* (uint pset.min = x; uint pset.max = x)'.

```
# poolcfg -c 'create pset pset_1to2 (uint pset.min = 1; uint
pset.max = 2)'
```

To create the pool, use poolcfg -c 'create pool pool_*poolname’*.

```
# poolcfg -c 'create pool pool_hrzone'
```

To associate the pset with the pool, use poolcfg -c 'associate pool pool_*poolname* (pset pset_*psetname*) '.

```
# poolcfg -c 'associate pool pool_hrzone (pset pset_1to2)'
```

Displaying and Committing the Modified Resource Pool Configuration File

To display the modified resource pool configuration, use `poolcfg -c info`.

```
# poolcfg -c info

<Output presented in the Notes>
```

To validate and commit the modified configuration, use `pooladm -n -c`, and then `pooladm -c`.

```
# pooladm -n -c
# pooladm -c
```

Displaying the Resource Pool Configuration That Is Currently in Use

To display the resource pool configuration that is currently in use, use `poolcfg -dc info`.

```
# poolcfg -dc info

system default
        string  system.comment
        int     system.version 1
        boolean system.bind-default true
        string  system.poold.objectives wt-load

        pool pool_hrzone
                int     pool.sys_id 1
                boolean pool.active true
                boolean pool.default false
                int     pool.importance 1
                string  pool.comment
                pset    pset_1to2
```

Displaying All Active Resource Pools

To display all the active resource pools on the system, use `poolstat -r all`.

```
# poolstat -r all
 id pool            type rid rset                  min  max size used load
  1 pool_hrzone     pset   1 pset_1to2               1    2    1 0.00 0.00
  0 pool_default    pset  -1 pset_default            1  66K    1 0.00 0.17
```

Binding the Zone to a Persistent Resource Pool

This section covers the following steps:

1. Listing the current state of the zones
2. Allocating the pool to the zone and confirming the allocation
3. Rebooting the zone to activate the resource pool binding
4. Confirming the availability of the resource pool

Listing the Current State of the Zones

To list the current state of the zones on the system, use `zoneadm list -iv`.

```
# zoneadm list -iv
   ID NAME      STATUS     PATH                BRAND    IP
   0 global     running    /                   solaris  shared
   1 hrzone     running    /zones/hrzone       solaris  excl
   2 itzone     running    /zones/itzone       solaris  excl
```

Allocating the Pool to the Zone and Confirming the Allocation

To allocate the pool to the zone, use `zonecfg -z` followed by the zone name and set `pool=pool_poolname`.

```
# zonecfg -z hrzone set pool=pool_hrzone
```

To confirm that the allocation has been made, use `zonecfg -z zonename info pool`.

```
# zonecfg -z hrzone info | grep pool
pool: pool_hrzone
```

Confirming the Availability of the Resource Pool

To confirm the availability of the resource pool, log in by using `zlogin zonename`, and then use `poolcfg -dc info`.

```
# zlogin hrzone
[Connected to zone 'hrzone' pts/2]
Oracle Corporation  SunOS 5.11    11.3   September 2015

# poolcfg -dc info

<Output presented in the Notes>

# exit
Logout
[Connection to zone 'hrzone' pts/2 closed]
```

Removing the Resource Pool Configuration

This section covers the following steps:

1. Removing the pool configuration from the zone
2. Rebooting the zone
3. Checking the resource pool configuration for the zone
4. Deleting the resource pool
5. Displaying all active resource pools

Removing the Pool Configuration from the Zone

To remove the resource pool configuration from the zone, use `zonecfg -z zonename clear pool`.

```
# zonecfg -z hrzone clear pool
```

Rebooting the Zone

To reboot the zone, use `zlogin hrzone init 6` command.

```
# zlogin hrzone init 6
```

Checking the Resource Pool Configuration for the Zone

To check the resource pool configuration for the zone, log in by using `zlogin zonename`, and then use `poolcfg -dc info`.

```
# zlogin hrzone
[Connected to zone 'hrzone' pts/2]
Oracle Corporation  SunOS 5.11    11.3   September 2015

# poolcfg -dc info
           <Output presented in the Notes>
# exit
Logout
[Connection to zone 'hrzone' pts/2 closed]
```

Deleting the Resource Pool

To delete the resource pool, use `pooladm -x`.

```
# pooladm -x
```

Displaying All Active Resource Pools

To display all the active resource pools on the system, use `poolstat -r all`.

```
# poolstat -r all
 id pool           type rid rset          min  max size used load
  0 pool_default   pset  -1 pset_default  1    66K    2 0.00 0.73
```

Allocating and Managing Physical Memory Resources with Resource Capping

To add a memory cap to a zone, perform the following steps:

1. Configure the zone by using `zonecfg -z zone`.
2. Add the memory cap resource and set each memory cap type as appropriate: physical, swap, and locked.
3. Verify, commit, and exit the zone.

```
# zonecfg -z itzone
zonecfg:itzone> add capped-memory
zonecfg:itone:capped-memory> set physical=50m
zonecfg:itzone:capped-memory> set swap=100m
zonecfg:itzone:capped-memory> set locked=30m
zonecfg:itone:capped-memory> end
zonecfg:itzone> verify
zonecfg:itzone> commit
zonecfg:itzone> exit
```

Quiz

To display the resource pool configuration that is currently in use, which of the following commands will you use?

a. poolstat -r all

b. poolcfg -c info

c. poolcfg –r all

d. poolcfg -dc info

e. poolstat –c info

f. poolstat –dc info

Quiz

The `pooladm –c` command validates and commits the modified resource pool configuration file.

a. True

b. False

Practice 7-1 Overview: Allocating Resources to Zones

This practice covers the following topics:

- Enabling services for resource pools
- Configuring a persistent resource pool
- Binding the zone to a persistent resource pool
- Removing the resource pool configuration

Agenda

- Explaining the fundamentals of Oracle Solaris 11 zones
- Managing system resources in a zone
- Migrating Oracle Solaris 10 systems
- Configuring Kernel Zones
- Working with Unified Archives

Migrating Oracle Solaris 10 Systems

- Migrating Oracle Solaris 10 zones
- Migrating Oracle Solaris 10 global zones

Migrating Oracle Solaris 10 Zones

1. Assess the Oracle Solaris 10 zone to be migrated.
2. Create an archive of the Oracle Solaris 10 zone to be migrated.
3. Prepare the Oracle Solaris 11 target system.
4. Migrate the Oracle Solaris 10 zone.

Migrating Oracle Solaris 10 Global Zones

1. Assess the global zone to be migrated.
2. Create an archive of the global zone to be migrated.
3. Prepare the Oracle Solaris 11 target system.
4. Migrate the Solaris 10 global zone.

Practice 7-2: Overview

- 7-2: Migrating an Oracle Solaris 10 Global Zone to Oracle Solaris 11

Agenda

- Explaining the fundamentals of Oracle Solaris 11 zones
- Managing system resources in a zone
- Migrating Oracle Solaris 10 systems
- Configuring Kernel Zones
- Working with Unified Archives

Kernel Zones Overview

- New in Oracle Solaris 11.2 release
- Is managed by from the global zone by using the existing tools
- Is a new zone brand with brand name as "`solaris-kz`" and pronounced as "kernel zones".
- Is managed by using existing tools such as `zonecfg`, `zoneadm`, and `zlogin`
- Provides a full kernel and user environment within a zone
- Increases kernel separation between the host and the global zone
- Provides greater flexibility in configuring and managing the zone than a default `solaris` zone
- Requires installing the `brand-solaris-kz` package

Why Kernel Zones?

From a technology perspective, in a nonglobal zone:

- The OS software layer must match across all instances on the same host
- Differing SRU/update levels for zones on the same host is not possible
- Suspending, resuming, and live migrating is not possible
- Managing ZFS pools and running a Common Internet File System (CIFS) server is not possible

From a customer perspective, in nonglobal zones:

- Aligning outage window to virtual environments is problematic
- Consolidating applications that require different OS versions is not possible
- Performing hardware maintenance on a system is not possible

Capabilities of Kernel Zones

The capabilities of kernel zones are derived from the following major design principles:

- Full zones integration
- Cross-platform implementation
- Emulation free
- Security
- Isolation
- Virtualization
- Granularity
- Transparency

System Requirements for Using Oracle Solaris Kernel Zones

To use kernel zones, your host hardware and host operating system must meet the following requirements:

- Operating system version of Oracle Solaris 11.2 or higher.
  - For SPARC systems:
    - A T4 SPARC system with System Firmware 8.5.0 or later
    - A T5, M5, or M6 system with System Firmware 9.2.0 or later
  - For X86 systems:
    - Nehalem+(Intel) or Barcelona+(AMD) based machines with:
      - CPU virtualization (for example, VT-x) enabled in the BIOS
      - Extended/Nested Page Table support, also referred to as EPT, NPT, or Rapid Virtualization Indexing (RVI)

In addition, both SPARC and x86 systems require:

- Minimum of 8 GB of RAM
- Kernel zone brand package, `brand/brand-solaris-kz`
- ZFS ARC tuning on the host

Auxiliary States

- Kernel zones use auxiliary states to communicate supplementary state information to the global zone.
- A kernel zone does not have an auxiliary state set by default.
- Auxiliary states are set only when you initiate debugging and kernel maintenance operations.
- The available kernel zone auxiliary states are:
  - `suspended`
  - `debugging`
  - `panicked`
- To view the global zone current state and the kernel zone auxiliary states, use the `zoneadm list –s` command.

```
global# zoneadm list -s
NAME    STATUS                 AUXILIARY        STATE
global  running
kzone1  running
kzone2  running
kzone3  running                debugging
```

Configuring Kernel Zones

1. Install and verify whether kernel zone packages are installed on the global zone.

```
# pkg install brand/brand-solaris-kz
# pkg list brand/brand/brand-solaris-kz
```

2. Create a new kernel zone configuration.
3. Add any additional kernel zone resources such as CPUs, memory, storage devices, and network devices.
4. Install the kernel zone by using the `zoneadm -z <zonename> install` command.
5. Boot the kernel zone by using the `zoneadm -z <zonename> boot` command.
6. Log in to the zone by using the `zlogin -C <zonename>` command.

Configuring Kernel Zones

```
globalzone# zonecfg -z kzone1
Use ’create’ to begin configuring a new zone.
zonecfg:kzone1> create -t SYSsolaris-kz
zonecfg:kzone1> add virtual-cpu
zonecfg:kzone1:virtual-cpu> set ncpus=8
zonecfg:kzone1:virtual-cpu> end
zonecfg:kzone1> info virtual-cpu
virtual-cpu:
ncpus: 8
zonecfg:kzone1> add device
zonecfg:kzone1:device> set match=/dev/rdsk/c9t0d0
zonecfg:kzone1:device> set id=4
zonecfg:kzone1:device> end
zonecfg:kzone1> info device
device:
match not specified
    storage.template: dev:/dev/zvol/dsk/%{global-
      rootzpool}/VARSHARE/zones/%{zonename}/disk%{id}
    storage: dev:/dev/zvol/dsk/rpool/VARSHARE/zones/kzone1/disk0
    id: 0
```

Configuring Kernel Zones

```
(continued...)

    bootpri: 0
device:
    match: /dev/rdsk/c9t0d0
    storage not specified
    id: 4
    bootpri not specified
zonecfg:kzone1> select capped-memory
zonecfg:kzone1:capped-memory> set physical=2048m
zonecfg:kzone1:capped-memory> end
zonecfg:kzone1> add anet
zonecfg:kzone1:anet> set id=3
zonecfg:kzone1:anet> end
zonecfg:kzone1> verify
zonecfg:kzone1> commit
zonecfg:kzone1> exit
# zoneadm –z kzone1 install
# zoneadm –z kzone1 boot
# zlogin –C kzone1
```

Host Data and Host ID

- Each kernel zone bootable device contains state information known as host data.
- A kernel zone's host data monitors kernel zone state information including:
  - Zone usage
  - Zone suspends
  - Time of day offset between the kernel zone clock and the global zone clock
  - OpenBoot variables (SPARC only)

Host Data and Host ID

- If the boot storage is in use, the kernel zone enters the `unavailable` state and displays an error message.
- If it is not in use, the kernel zone can be repaired by using the `zoneadm attach -x force-takeover` command.

```
global# zoneadm -z kzone1 attach
zone 'kzone1': error: ERROR: zone kzone1 is in use by host with
hostid 848611d4
zone 'kzone1': error: last known state: installed
zone 'kzone1': error: hostname: global2
zone 'kzone1': error: boot environment name: solaris-1
zone 'kzone1': error: boot environment uuid:
69ed2e6a-e25a-6d36-e022-ed7261ed8899
zone 'kzone1': error: last update time: Sun Apr 13 20:08:13 2014
zone 'kzone1': error: To fix, detach the zone from the other host then
attach it to this host
zone 'kzone1': error: If the zone is not active on another host,
attach it with
zone 'kzone1': error: zoneadm -z kzone1 attach -x force-takeover
```

Boot Loader

- The kernel zone boot loader manages booting operations on the kernel zone.
- To invoke the boot loader, the kernel zone must be in the `ready` or `installed` state.
- You can use the kernel zone boot loader to perform the following operations:
  - List alternative BEs
  - Boot the zone to an alternative BE
- To invoke the kernel zone boot loader, use the `zoneadm boot` command.

Boot Loader

To specify an alternative BE in a kernel zone:

1. Log in to the zone console.

```
# zlogin -C zonename
```

2. In a separate terminal window, list the available kernel zone BEs.

```
# zoneadm -z zonename boot -- -L
```

3. Boot to a selected BE.

```
# zoneadm -z zonename boot -- -Z boot-environment
```

Cloning a Kernel Zone

- Cloning allows you to make copies of an existing configured and installed zone.
- The cloned zone includes any changes that were made to customize the existing zone.
- Cloning a zone is an efficient way to add additional zones with a similar customized zone configuration.
- You can clone a kernel zone in the following ways:
  - By using the `zoneadm clone` command
  - By using Unified Archives

Cloning by Using `zoneadm clone`

In the following example, the kernel zone `kzone1` is cloned to the kernel zone `kzone2` on the host `global`:

1. To create a clone, you need to first halt that zone.

```
global# zoneadm -z kzone1 halt
```

2. Create a new zone configuration.

```
global# zonecfg -z kzone2 create -t kzone1
```

3. Clone the zone by using the `zoneadm clone` command.

```
global# zoneadm -z kzone2 clone kzone1
```

Cloning by Using Unified Archives

Cloning by using Unified Archives is recommended for large deployments.

Note: You will learn more about cloning by using Unified Archives in the next section “Working with Unified Archives.”

Suspending and Resuming a Kernel Zone

- The suspend operation is unique to kernel zones.
- You can suspend a kernel zone to disk by means of the `zoneadm suspend` command.
- The `zoneadm suspend` command places a kernel zone in the `suspended` state.
- You can resume a suspended zone by invoking the `zoneadm boot` command.
- Suspend and resume are supported only if a kernel zone has the `suspend` resource property in its configuration.

```
global# zonecfg -z kzone1 info suspend
suspend:
path: /system/zones/kzone1/suspend
storage not specified
```

Warm Migrating a Kernel Zone

The following example warm migrates the `kzone1` zone onto the new host:

1. On the global zone, suspend the file system on the kernel zone to be migrated.

```
global# zoneadm -z kzone1 suspend
```

2. Detach the kernel zone file system on the global zone.

```
global# zoneadm -z kzone1 detach
```

3. Export the zone configuration and transfer the file onto the new host.

```
global# zonecfg -z kzone1 export | ssh root@global2 \
zonecfg -z kzone1 -f -
```

Warm Migrating a Kernel Zone

4. Attach the zone on the new host.

```
global2# zoneadm -z kzone1 attach
```

5. Boot the kernel zone on the new host to resume the migrated zone.

```
global2# zoneadm -z kone1 boot
```

Quiz

To migrate a Kernel zone, you need to first halt it, detach it, and then attach it on the remote machine.

a. True

b. False

Quiz

Which are the following are the available kernel zone auxiliary states?

a. `panicked`

b. `detached`

c. `configured`

d. `suspended`

e. `debugged`

Practice 7-3 Overview: Configuring Kernel Zones (Demonstration)

This practice provides steps to access a demonstration on the Kernel zones feature as part of which you will get to observe the following:

- Configuring Kernel zones
- Cloning a Kernel zone
- Warm migration of a Kernel zone

Note: This practice is only a demonstration.

Agenda

- Explaining the fundamentals of Oracle Solaris 11 zones
- Managing system resources in a zone
- Migrating Oracle Solaris 10 systems
- Configuring Kernel Zones
- Working with Unified Archives

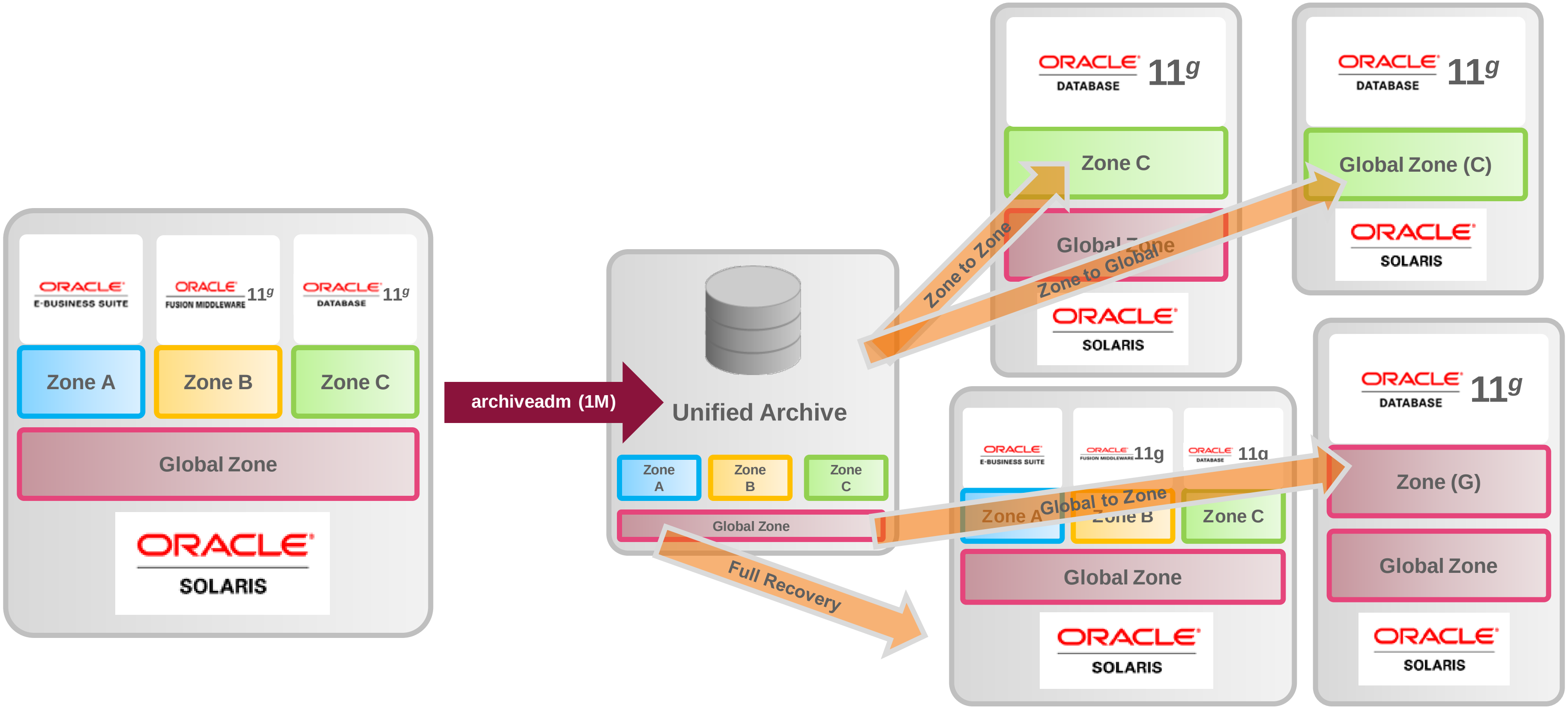
Archival Before Oracle Solaris 11

- Before Oracle Solaris 11, the default archive solution was the Oracle Solaris Flash Archive.
- Flash archives were designed to create and deploy OS instances of and to bare metal systems.
- With the introduction of BEs, IPS, and virtualization technologies, a more robust archive solution became possible.
- Unified Archives employs these technologies and provides a much more flexible and extensible solution.

Oracle Solaris Unified Archives Overview

- A new native archive type for Oracle Solaris, which may contain a global zone, a nonglobal zone, or a kernel zone
- Allows multiple system instances to be archived in a single unified file format
- Allows deploying the archives to recover a system by using one of the following methods:
  - Automated Installer (AI)
  - Zones utilities
  - Bootable media
- Provides `archiveadm` command-line utility to create the archive
- Provides the following main functions:
  - System recovery
  - System cloning
  - System migration

Oracle Solaris Unified Archives Overview



ARCHIVE

DEPLOYMENT

Key Terms

Terms	Description
System	A single deployed instance of an operating environment.
Archive	Set of system data and metadata which describes it and the system(s)
System Image	A specific system’s data within an archive or elsewhere
Unified Archive	One or more system images bundled into a single archive. More specifically, an OVF-compliant OVA file with a nontraditional stream-based payload.

Unified Archives Types

Following are the two types of Unified Archives:

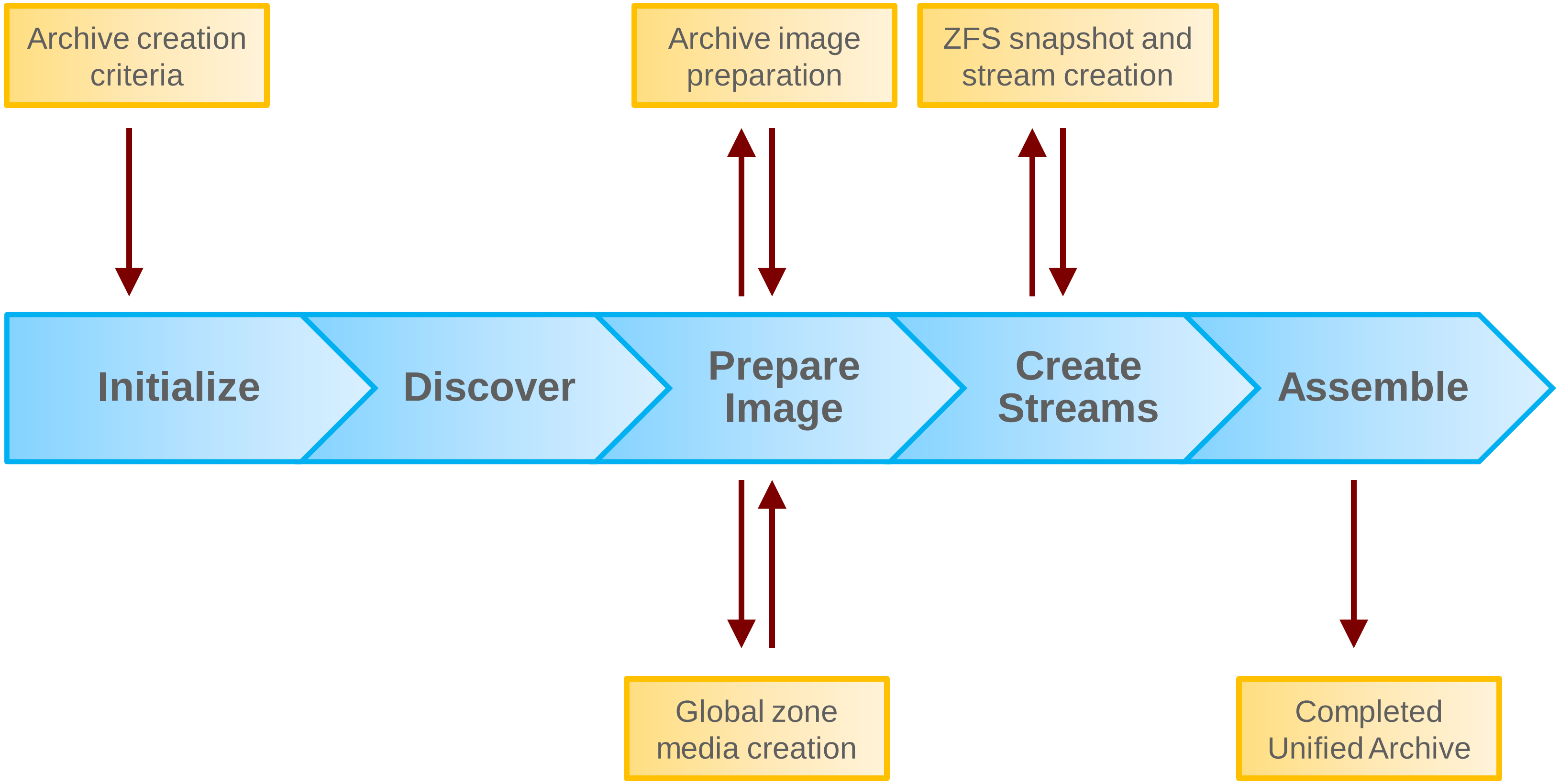
Unified Archive Types	Description
Clone archive	Allows archiving only one BE at a time
Recovery archive	Allows creating a full-system archive containing all boot environments from all included instances

Clone Archive	Recovery archive
Individually deployable systems	Single deployable system
Active BEs only	Active and Inactive BEs
Used for cloning	Used for recovery and migration

Platform Portability

- The requirement for platform compatibility is that the systems be of the same instruction set architecture (ISA).
- Unified Archives also enable image transforms.
- However, there are some systems that cannot be transformed, such as:
  - Oracle Solaris 10 brand zones
  - Trusted Solaris or labeled zone archives

Unified Archives Creation


Archive creation criteria

Archive image preparation

ZFS snapshot and stream creation

Initialize

Discover

Prepare Image

Create Streams

Assemble

Global zone media creation

Completed Unified Archive

Creating a Clone Archive

- By default, a clone archive is created.
- Use the `archiveadm create <archive-name>` command to create a clone archive.

```
# archiveadm create -z kzone1 /var/tmp/kzone1.uar
Initializing Unified Archive creation resources...
Unified Archive initialized: /var/tmp/kzone1.uar
Logging to: /system/volatile/archive_log.29929
Executing dataset discovery...
Dataset discovery complete
Creating install media for zone(s)...
Media creation complete
Preparing archive system image...
Beginning archive stream creation...
Archive stream creation complete
Beginning final archive assembly...
Archive creation complete
# archiveadm info /var/tmp/kzone1.uar
Archive Information
          Creation Time:  2014-05-28T10:16:01Z
            Source Host:  t4-1ou
           Architecture:  sparc
       Operating System:  Oracle Solaris 11.3 SPARC
      Deployable Systems: kzone1
```

Creating a Recovery Clone

Use the `archiveadm create -r <archive-name>` command to create a recovery clone.

```
# archiveadm create -r -z global /var/tmp/globalrecovery.uar
Initializing Unified Archive creation resources...
Unified Archive initialized: /var/tmp/globalrecovery.uar
Logging to: /system/volatile/archive_log.11234
Executing dataset discovery...
Dataset discovery complete
Preparing archive system image...
Beginning archive stream creation...
Archive stream creation complete
Beginning final archive assembly...
Archive creation complete
# archiveadm info globalrecovery.uar
Archive Information
             Creation Time: 2013-10-03T18:13:21Z
               Source Host: example
              Architecture: i386
          Operating System: Oracle Solaris 11.3 X86
        Deployable Systems: global
```

Deploying Unified Archives

- After an archive is created, you can store it as a file until it needs to be deployed.
- Deployment scenarios include:
  - System recovery
  - System cloning
  - System migration
- Archive deployment modes are:
  - Oracle Solaris Automated Installer (AI)
  - Oracle Solaris Zones utilities
  - Bootable media

Deploying a Zone from Unified Archives by Using Zones Utilities

1. Select a zone configuration to use or create one.
   – If you have already created a zone configuration, skip to Step 2.
   – If you need to create a zone configuration from an existing archive, use the following steps.
      A. Clone the zone configuration using the archive.

```
# zonecfg -z new-zone create -a archive
```

      B. Verify that the zone configuration was created.

```
# zoneadm list -cv
```

2. Install and boot the zone.

```
# zoneadm -z new-zone install –a archive
# zoneadm -z new-zone boot
```

Deploying an Archive Using AI

To deploy an archive by using AI:

- You must have a compatible AI service and AI manifest
  - For example, if the archived system is Oracle Solaris 11.3 FCS, any Oracle Solaris 11.3 version AI service is compatible.
- You have to modify the AI manifest to include the following information:
  - The location of the archive
  - The name of the systems the archive should be deployed to

Note: Each archive does not require its own service. The specific archive deployed is determined by the manifest used for a specific deployment.

Deploying a System from Unified Archives by Using AI

1. If required, create a new install service.

```
# installadm create-service -n new-service -s <ISO_source_or_FMRI_location>
```

2. Edit the default manifest XML file to reference the Unified Archive or create a new manifest with the updated manifest XML file.

```
# installadm create-manifest -n svcname -m manifest -f file
```

3. Confirm whether the service was created and the new manifest was applied to the service.

```
# installadm list -n new-service
# installadm list –m manifest-file
```

Deploying an Archive Using Bootable Media

1. Create a media image from an existing archive.

```
# archiveadm create-media archive-name
```

   - For example, to create an ISO image, you must include the `-f` option.

```
# archiveadm create-media -f iso archive.uar
```

2. Create bootable media.
   - ISO images: Burn the `.iso` file to a CD or DVD.
   - USB images: Use the `usbcopy` utility to copy the image to a USB flash drive.
3. Boot from the media.

Troubleshooting Cloning and Archiving

- Issues may be encountered when creating and deploying Unified Archives.
- Most issues related to archive creation involve system configuration, available space for archive storage, and connectivity to the package repositories, which are used to prepare the image for redeployment.
- Common deployment issues include target configuration and selection problems and failure to configure the deployed system.
- In the case of archive creation or deployment errors, logging files are created. These files contain detailed data regarding the errors encountered.

Note: With Solaris Kernel Zones, some log files may be within the specific zone.

Practice 7-4 Overview: Working with Unified Archives (Demonstration)

Note: This practice is only a demonstration.

Summary

In this lesson, you should have learned how to:

- Explain the fundamentals of Oracle Solaris 11 zones
- Manage system resources in a zone
- Migrate Oracle Solaris 10 systems
- Configure kernel zones
- Work with Unified Archives