

D82701GC10

Edition 1.0

December 2013

D85050

ORACLE®

Oracle Solaris 11 System Administration for Experienced UNIX/Linux Administrators

Activity Guide

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Authors

Uma Sannasi, J S Raghavendra

Technical Contributors and Reviewers

Geetha Nazare, Shripad Patki, John Hathaway, Harry Burks, Rosemary Martinak, Todd Lowry, Rajesh Rajasekharan, Gary Riseborough, David S Maxwell, Glynn Foster, Alex Barclay, Deepak Dhanukodi, Yousuf Mohammed, Niveditha Sananth, Sharvani Nagamalli, Pranamya Jain, Sreejith Mohan, Manish Pawar

Editors

Daniel Milne, Richard Wallis, Smita Kommini

Graphic Designers

Prakash Damodaran, Seema Bopaiah

Publishers

Syed Ali, Srividya Rameshkumar

This book was published using: **oracle***tutor*

Table of Contents

Practices for Lesson 1: Course Introduction	1-1
Practices for Lesson 1: Overview.....	1-2
Practices for Lesson 2: Administering System Software by Using IPS.....	2-1
Practices for Lesson 2: Overview.....	2-2
Practice 2-1: Configuring a Local IPS Repository	2-4
Practice 2-2: Configuring a Client to Access the Local IPS Server	2-9
Practice 2-3: Managing Software Packages by Using the Command-Line Interface.....	2-12
Practice 2-4: Managing Software Packages by Using Package Manager	2-15
Practice 2-5: Managing Multiple Boot Environments	2-24
Practices for Lesson 3: Administering Services by Using SMF	3-1
Practices for Lesson 3: Overview.....	3-2
Practice 3-1: Configuring SMF Services.....	3-3
Practice 3-2: Modifying a Service Profile.....	3-8
Practice 3-3: Restoring a Service.....	3-10
Practices for Lesson 4: Administering ZFS.....	4-1
Practices for Lesson 4: Overview.....	4-2
Practice 4-1: Configuring ZFS Storage Pools	4-4
Practice 4-2: Administering ZFS File Systems.....	4-9
Practice 4-3: Configuring ZFS Properties	4-11
Practice 4-4: Configuring ZFS Shadow Migration	4-25
Practice 4-5: Administering ZFS Snapshots and Clones	4-28
Practice 4-6: Performing ZFS Data Backup and Recovery	4-35
Practices for Lesson 5: Configuring the Network.....	5-1
Practices for Lesson 5: Overview.....	5-2
Practice 5-1: Initializing the Network Setup	5-4
Practice 5-2: Creating a Private Virtual Network	5-11
Practice 5-3: Configuring Link Aggregation in the Global Zone.....	5-13
Practice 5-4: Regulating Bandwidth by Using the Datalink Properties.....	5-17
Practice 5-5: Regulating Bandwidth by Using Flows	5-20
Practices for Lesson 6: Administering: Oracle Solaris Zones.....	6-1
Practices for Lesson 6: Overview.....	6-2
Practice 6-1: Configuring zgateway as a Whole Root Zone.....	6-4
Practice 6-2: Configuring dbzone as a Branded Zone.....	6-12
Practice 6-3: Configuring IPMP for Network High Availability	6-18
Practice 6-4: Delegating ZFS Datasets to Zones	6-22
Practice 6-5: Allocating Resources to Zones	6-27
Practices for Lesson 7: Administering Privileges and RBAC.....	7-1
Practices for Lesson 7: Overview.....	7-2
Practice 7-1: Creating User Accounts	7-3
Practice 7-2: Setting User and Group Quotas on a ZFS File System.....	7-8
Practice 7-3: Configuring RBAC.....	7-11
Practices for Lesson 8: Installing the Oracle Solaris 11.1 Operating System	8-1
Practices for Lesson 8: Overview.....	8-2
Practice 8-1: Customizing the Automated Installation Service.....	8-4

Practices for Lesson 9: Monitoring System Resources..... 9-1

Practices for Lesson 9: Overview..... 9-2

Practice 9-1: Evaluating System Performance Levels..... 9-3

Practice 9-2: Monitoring Zone Resource Utilization 9-14

Practice 9-3: Analyzing Applications at Run Time by Using DTrace One-Liners..... 9-17

Practice 9-4: Analyzing Applications at Run Time by Using the DTrace Toolkit..... 9-23

Practices for Lesson 1: Course Introduction

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

This practice introduces you to your course assignment and the infrastructure, which you will use for performing the practices. The practices map to the respective lessons.

Practices Infrastructure

This section presents an architectural view of the equipment and the platforms for the practices. Three virtual machines (VMs) are configured on a private internal network (192.168.0). Each VM can communicate with other VMs only on the same private network (see Figure 1).

The VMs are configured to communicate with the host machine only through the share directory.

Note: Internet access is not configured from these VMs.

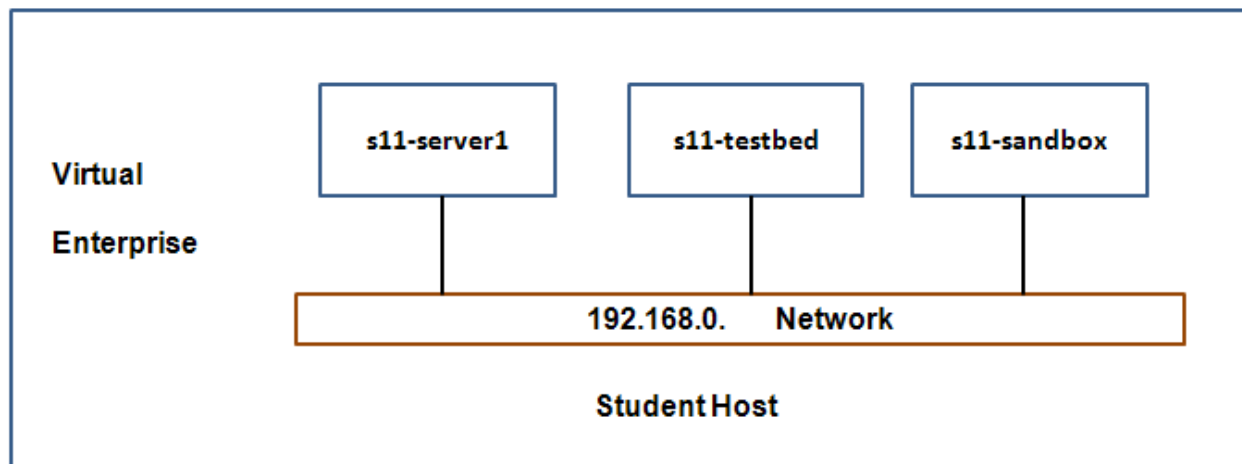


Figure 1: Virtual Pod Network Schematic

Your lab environment is based on the Oracle VM VirtualBox virtualization software. The VirtualBox is a cross-platform virtualization application. Figure 2 shows the configured virtual machines. The Oracle Solaris 11.1 OS is installed in the virtual machines with the exception of **s11-sandbox**, which is an empty VM.

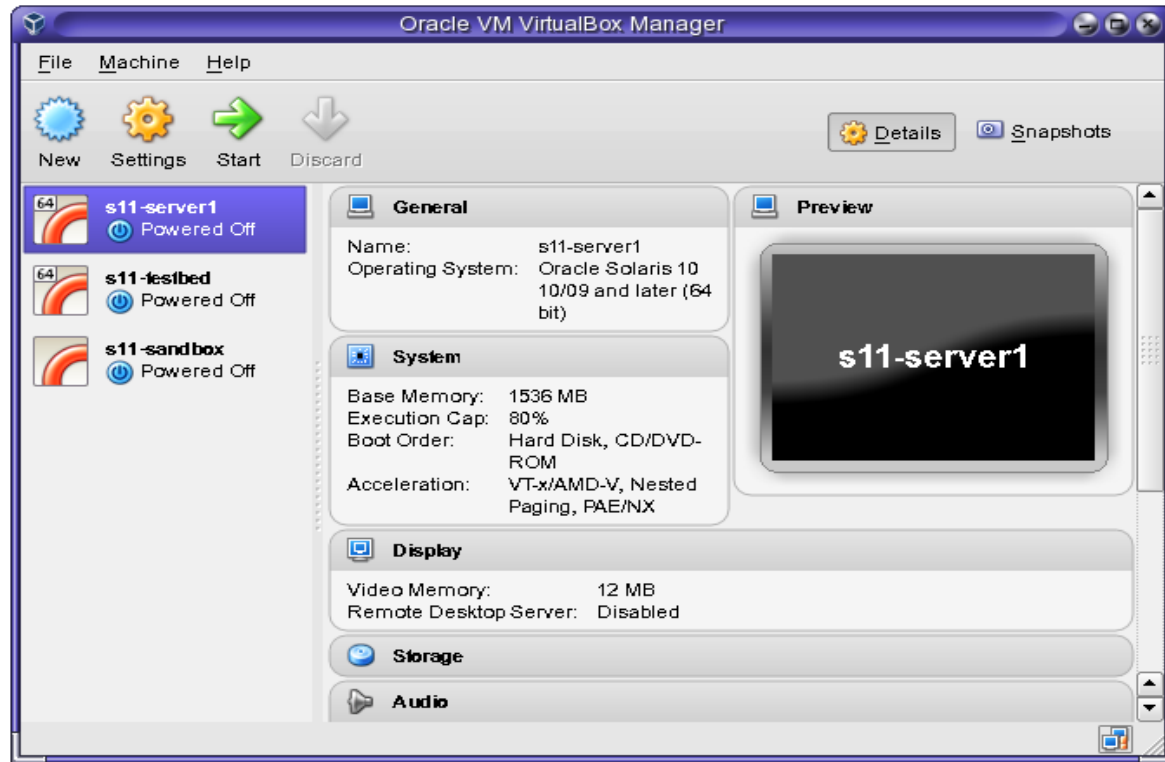


Figure 2: Oracle VirtualBox VMs

The VirtualBox environment consists of the following VMs:

VM	Description
s11-server1	This VM provides network services, such as DNS, DHCP, and IPS that are used by other VMs in this virtual network. This VM should always be up and running. You use the command-line tools here.
s11-testbed	This is a general-purpose user machine with the GUI and other features normally available on a network client machine. Most of the facilities available in s11-server1 are available in this VM.
s11-sandbox	This is the VM for Oracle Solaris 11.1 installation that uses Automated Install mode. After performing the practice, switch off this VM. It will not be needed for any other practice.

The following are the memory configuration for the Virtual Machines. Most of the host machines have a total of 8 GB to work with.

VM	Memory
s11-server1	2 GB
s11-testbed	2 GB
s11-sandbox	2 GB

All the student files are located in the `/opt/ora/` directory. This directory contains mostly scripts that you may be directed to use to establish the start or end state of a particular practice.

Logging In to the Practice Environment

When you first log in to the practice environment, you are prompted to provide a login and password for the host system:

- **User ID:** `root`
- **Password:** `oracle`

After you have gained access to the host system, the user account and password for each virtual machine is:

- **User account:** `oracle`
- **Password:** `oracle1`
- **Administrator privileges:** As the `oracle` user, use `su -` to switch to the primary administrator (`root`) role. The password is `oracle1`. The `oracle` user switches to `root` because `root` is configured as a role by default. The first username created on the system (during the OS installation) is the initial privileged user who can assume the administrator role. This can be verified in the `/etc/user_attr` file.

Note: The `s11-server1` VM must be started before any additional VMs are started. The `s11-server1` VM must always be running to perform the practices in this guide.

Best Practices

- When required to perform the steps from `s11-server1` VM, establish secure remote connection with the `s11-server1` by using `ssh` and switch to `root` role from `s11-testbed` VM.

```
oracle@s11-testbed:~$ ssh s11-server1
Password:
Oracle Corporation      SunOS 5.11      11.1      September 2012
oracle@s11-server1:~$ su -
Password:
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-server1:~#
```

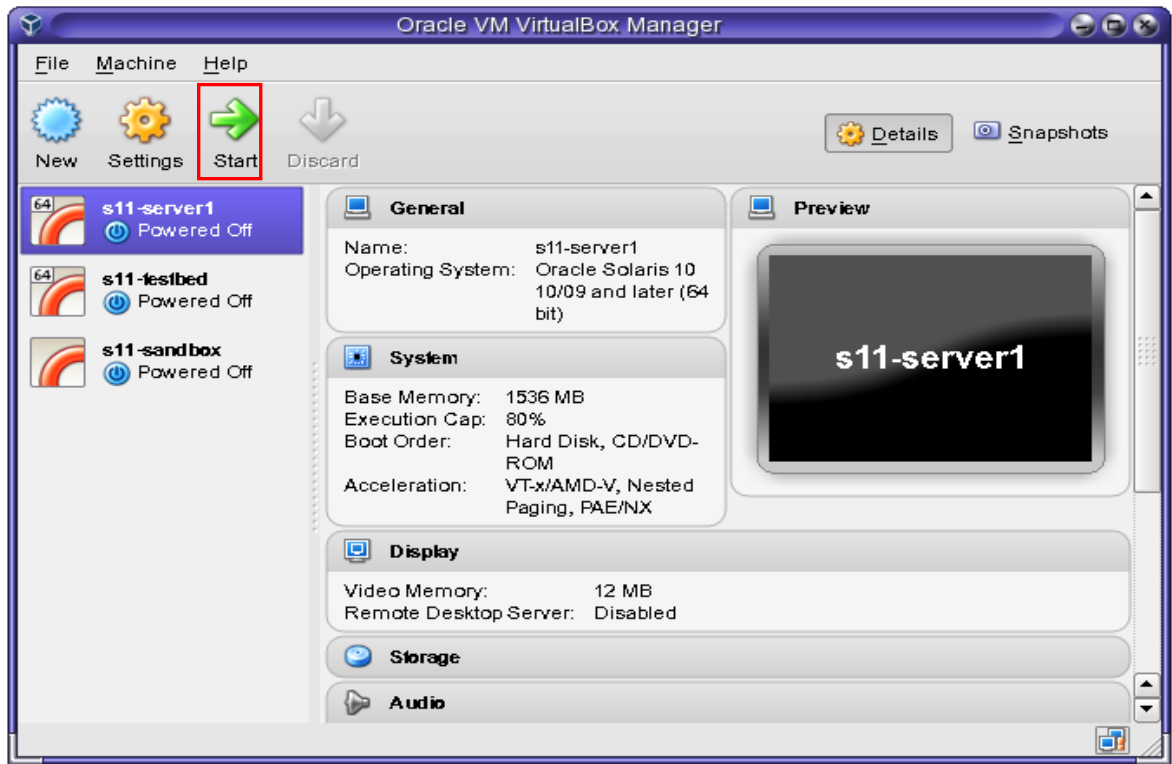
- Your system performance depends on the network speed and network load. If you find your VM too slow to proceed with, it is suggested that you restart the VM.
- Follow the instructions in the practices for a smooth learning experience.
- Close all the terminals in the `s11-testbed` VM after completing the individual practices.

Task: Becoming Familiar with Your Practice Environment

1. On your host system, start the Oracle VM VirtualBox Manager by double-clicking its icon on your desktop.



- In the Oracle VM VirtualBox Manager window, double-click the **s11-server1** VM to start it. Alternatively, you can simply select the **s11-server1** VM and click the **Start** button.



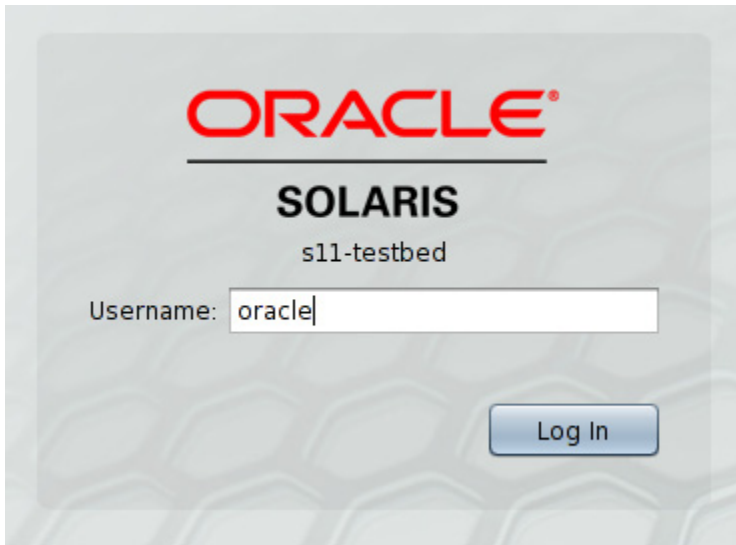
- After the **s11-server1** VM is powered on, at the command prompt, log in as the user **oracle** with the password **oracle1**.

```
s11-server1 console login: oracle
Password: oracle1
Last Login: Mon Nov 12 03:59:49 on console
Oracle Corporation SunOS 5.11 11.1 September 2012
Or

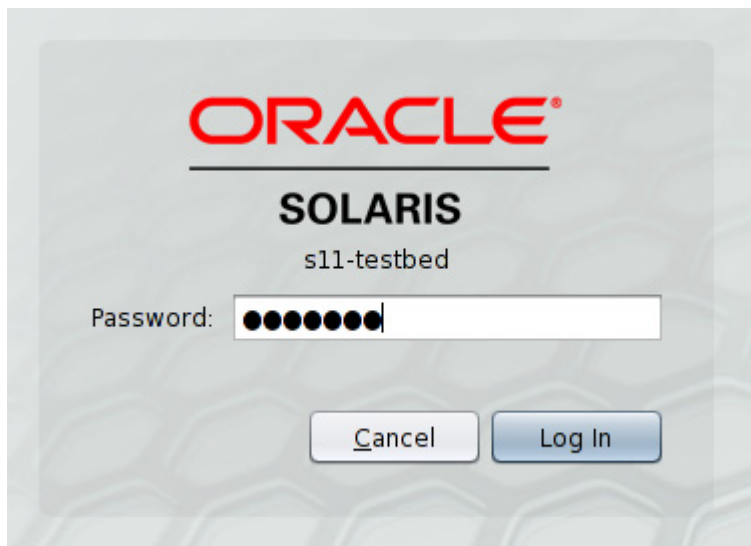
oracle@s11-server1:~$
oracle@s11-server1:~$ su -
Password: oracle1
...
root@s11-server1:~#
```

4. Start the **s11-testbed** VM. When the Username login screen appears, enter `oracle` for the username and click the **Log In** button.

Note: It might take a few minutes for the **Username** login screen to appear.



5. When the password login screen appears, enter the password `oracle1` and click the Log In button.

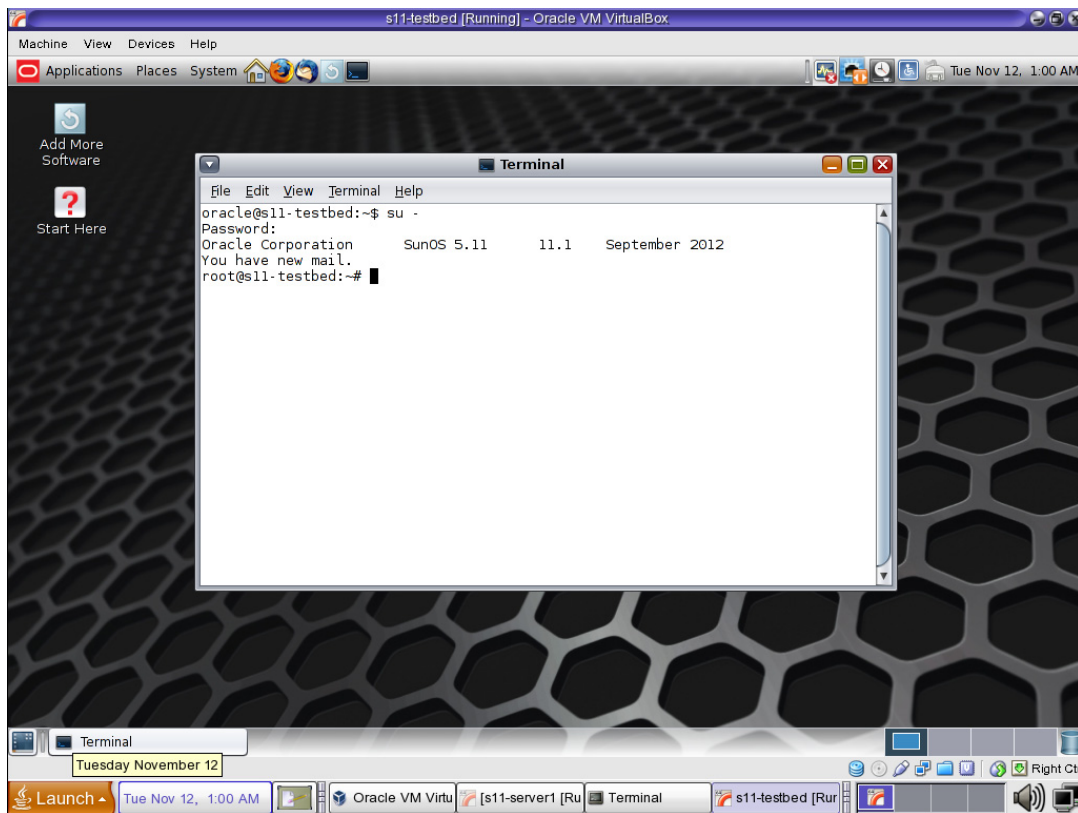


- Open a terminal window by right-clicking the desktop and selecting **Open Terminal**. In the terminal window, run the `su -` command to assume the administrator privileges. The password is `oracle1`.

```
oracle@s11-testbed:~$ su -  
Password: oracle1  
Oracle Corporation      SunOS 5.11 11.1      September 2012  
root@s11-testbed:~#
```

- At times, you may need to power off a VM and close its window. You may also need to shut down a VM to comply with the maximum recommended number of VMs running simultaneously, which is currently limited to three VMs.

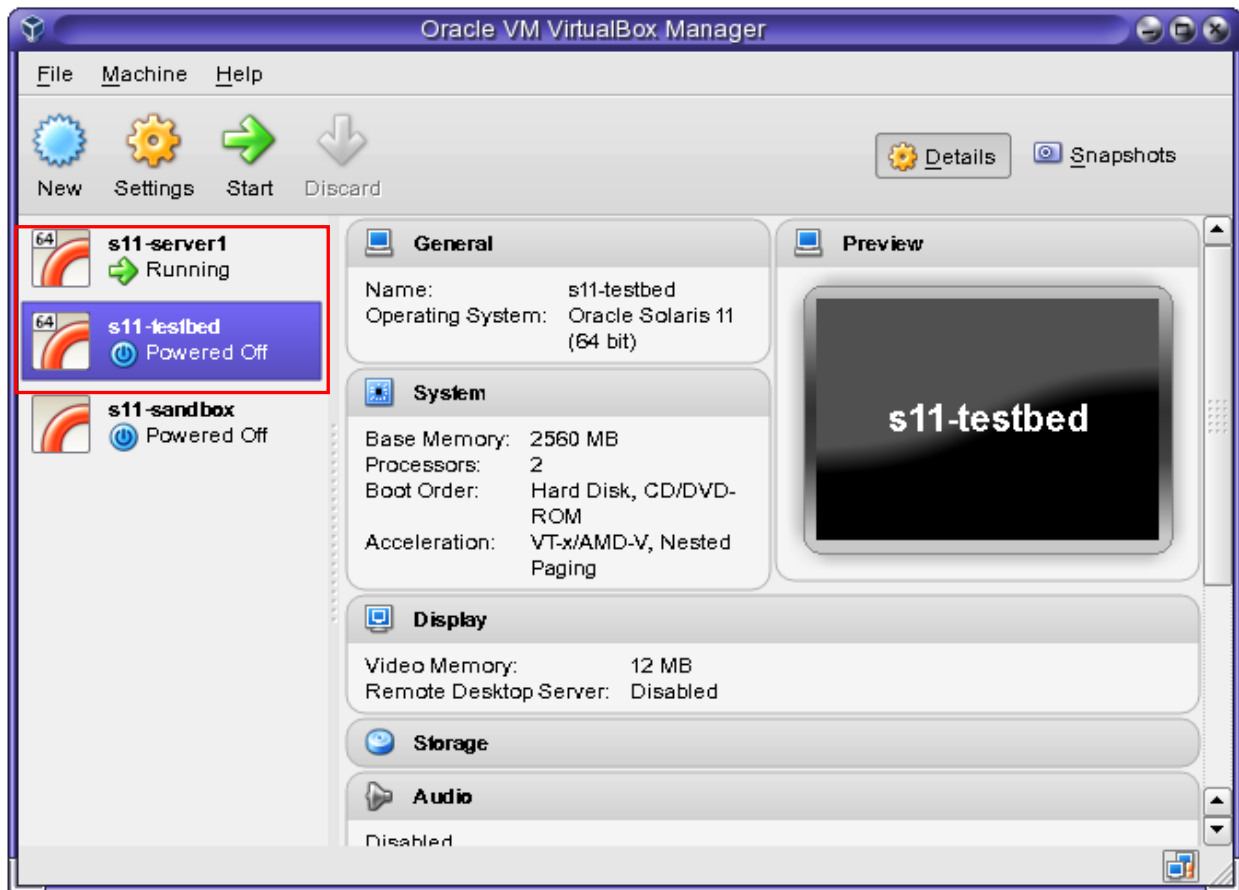
Now, practice shutting down a VM by using the **s11-testbed**. To shut down the VM, click the “close” button (x) in the top-right corner of the VM window.



- When the Close Virtual Machine dialog box appears, select **Power off the machine** and click OK.



Note: You can verify that the VM is shut down by checking the status that appears under the VM's name in the Oracle VM VirtualBox Manager. The status for the **s11-testbed** should be "Powered Off." The status for the **s11-server1** VM should be "Running."



Introducing the Practices

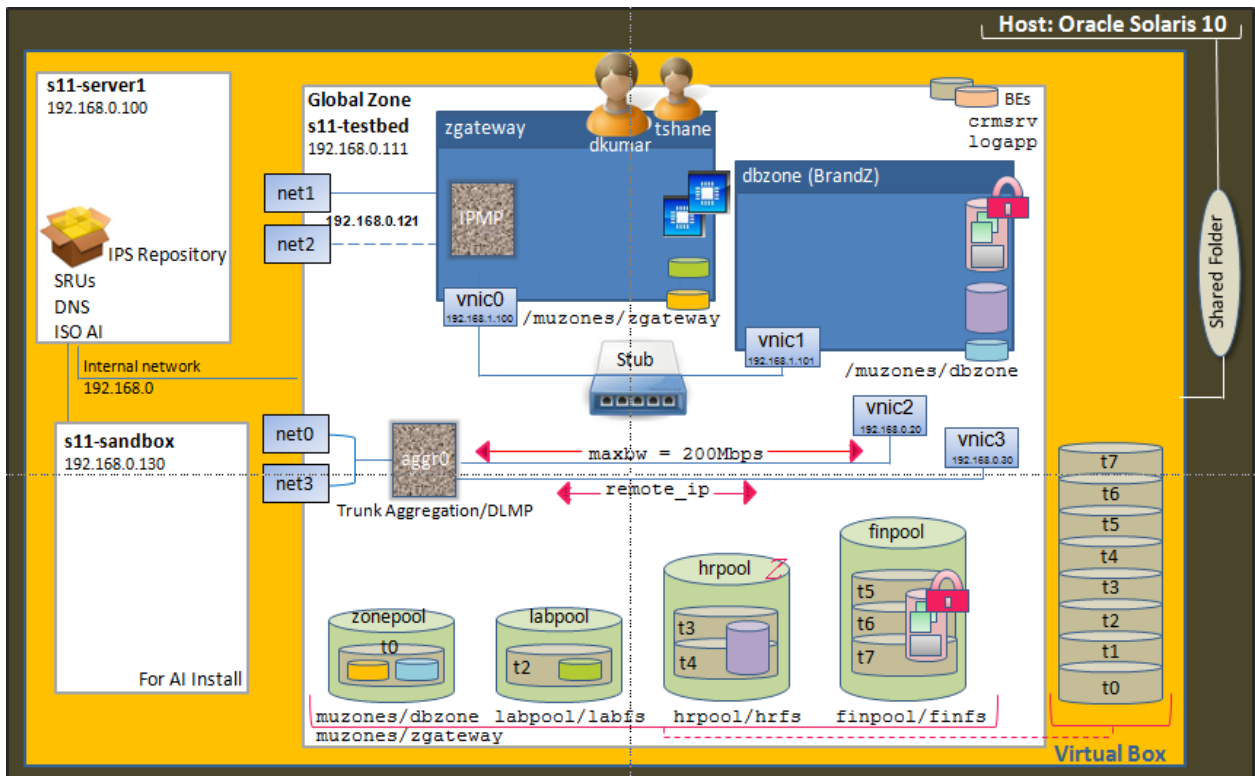
The practices in this course are meant to provide a consolidated experience of deploying the various features and technologies of Oracle Solaris 11.1. To that extent, the practices are driven by certain requirements explained in each exercise.

Scenario:

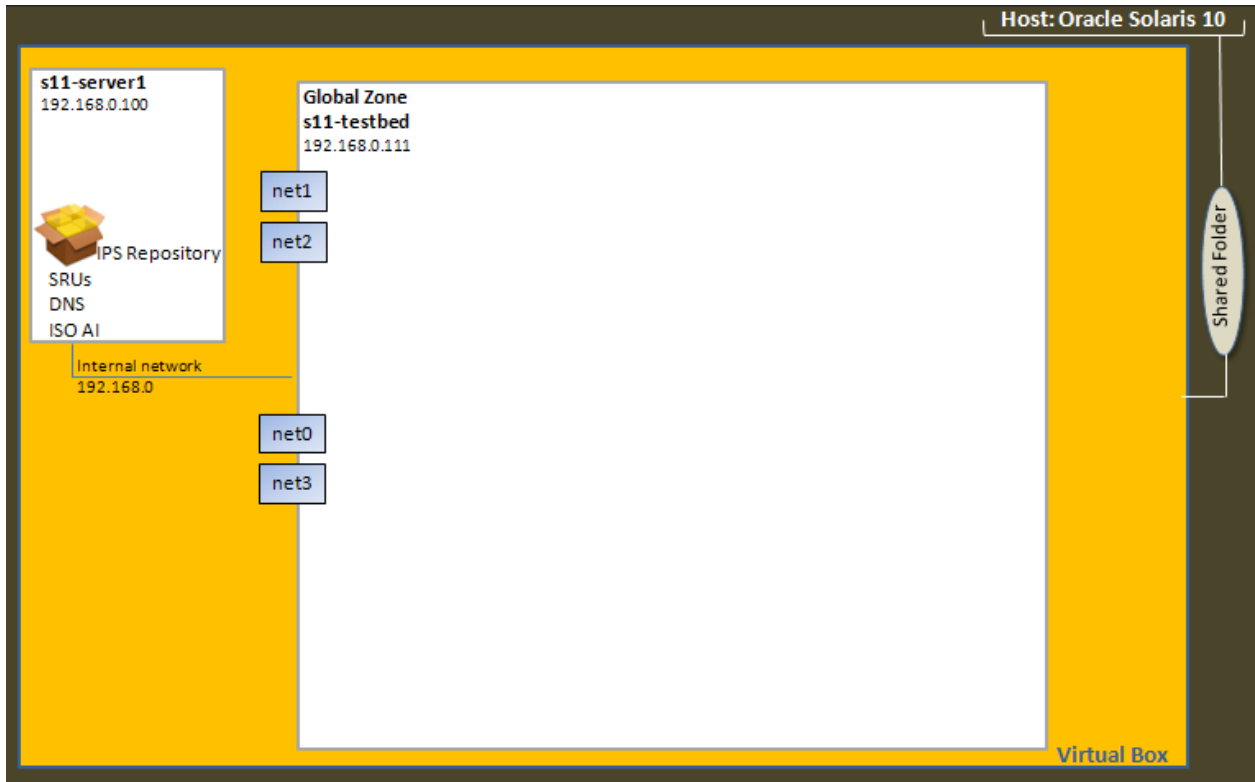
Murraya Inc., a worldwide freight company, has considered phasing in Oracle Solaris 11.1 at its data center. You are part of a larger team of system administrators at Murraya that is responsible for configuring and administering a Solaris testbed—an infrastructure for validating and tuning custom and Oracle applications on Oracle Solaris 11 before migrating to a production environment. More importantly, the testbed will be a platform where you and your team members can experiment with various Oracle Solaris 11.1 features and technologies before implementing them on a live system.

Implementation Layout:

Starting with “Practices for Lesson 2,” you will be involved in addressing specific requirements until the testbed implementation evolves into a consolidated infrastructure, as depicted in the schematic representation below:



You will start “Practices for Lesson 2” with a bare infrastructure as illustrated below:



This is your starting point.

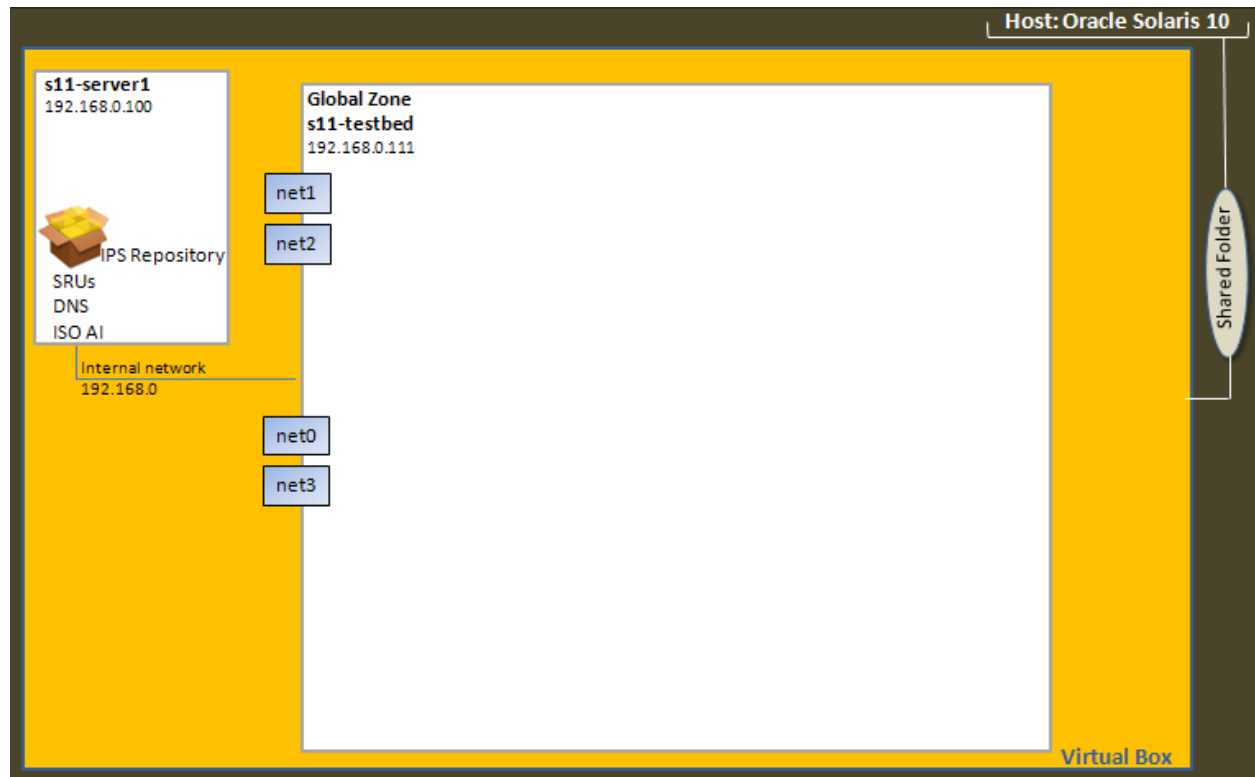
Practices for Lesson 2: Administering System Software by Using IPS

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

Below is the schematic representation of the environment that you are provided with. Observe that the testbed infrastructure at this stage is at its most minimal configuration. From here on, you will be engaged in building the infrastructure so that it meets the requirements of a validation and tuning environment.



Your first activity is to set up a local IPS repository. A local IPS repository is meant to supply Oracle Solaris packages to the clients.

Also, recall that boot environments (BEs) provide risk-free upgrades. During package update or system upgrade, if a system were to get corrupted, you can revert to the previous environment. As the testbed environment evolves, it is a good practice to maintain multiple BEs with each delta change that takes place in the infrastructure.

In this practice, you will perform the following:

- Configure a local IPS repository.
- Configure the client to access the IPS repository.
- Manage software packages.
- Manage boot environments.

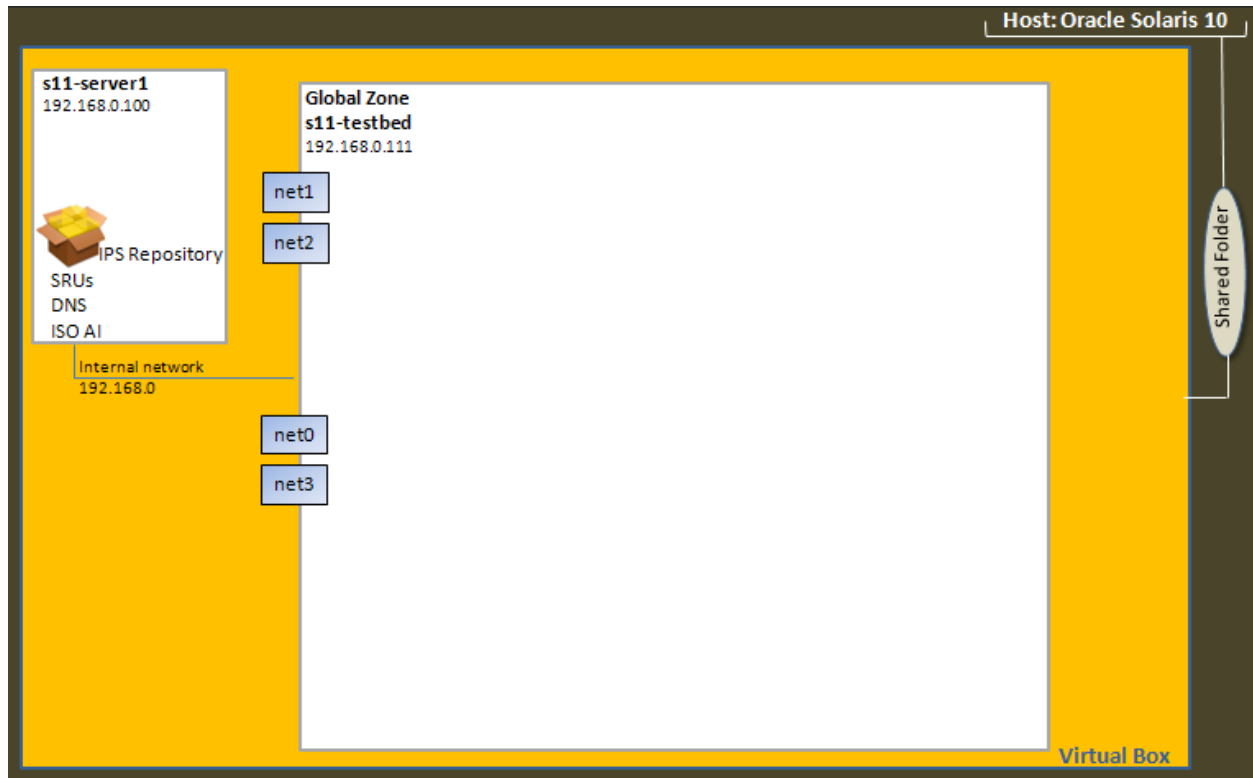
Notes

- Some command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.
- Do not perform the steps given under the additional note section. These are only for your reference.

Practice 2-1: Configuring a Local IPS Repository

Overview

In your lab environment, the **s11-testbed** Virtual Machine (VM) system cannot access the default publisher URL to download the Oracle Solaris packages. Recall that Oracle Solaris 11.1 has only one publisher configured: the `solaris` publisher. Therefore, your first task is to create your local package repository on the **s11-server1** system and make it the default so that the clients on the network can be serviced by IPS.



In this practice, you will configure a local IPS repository on the **s11-server1** VM.

Tasks

1. Verify that the **s11-server1** VM is running.
2. Log in to the **s11-server1** VM as the `oracle` user. Use the password `oracle1`.
3. Run the `su` command to assume administrator privileges.

```
oracle@s11-server1:~$ su -
Password: oracle1
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-server1:~#
```

4. Determine the host name and domain of this server.

```
root@s11-server1:~# hostname
s11-server1
root@s11-server1:~# domainname
mydomain.com
```

5. Verify that this server can access DNS services.

```
root@s11-server1:~# nslookup s11-server1
Server:      192.168.0.100
Address:     192.168.0.100#53

Name:       s11-server1.mydomain.com
Address:    192.168.0.100
```

6. Verify that the rpool/export/IPS file system is mounted on the /export/IPS directory.

```
root@s11-server1:~# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool    39.5G  10.3G  29.2G  26%  1.00x  ONLINE  -

root@s11-server1:~# zfs list
NAME                                           USED  AVAIL  REFER  MOUNTPOINT
rpool                                         10.4G  28.5G   4.58M  /rpool
rpool/ROOT                                   2.22G  28.5G    31K  legacy
rpool/ROOT/solaris                         2.22G  28.5G   2.06G  /
rpool/ROOT/solaris/var                     100M  28.5G   98.8M  /var
rpool/VARSHARE                             46K   28.5G    46K  /var/share
rpool/dump                                 1.03G  28.6G   1.00G  -
rpool/export                               6.07G  28.5G    35K  /export
rpool/export/IPS                           6.07G  28.5G   6.07G  /export/IPS
rpool/export/ZFS_data                       31K   28.5G    31K  /export/ZFS_data
rpool/export/home                          102K   28.5G    33K  /export/home
rpool/export/home/gail                      35K   28.5G    35K
rpool/export/home/oracle                    34K   28.5G    34K
rpool/swap                                 1.03G  28.6G   1.00G  -
root@s11-server1:~#
```

Note: Normally, a local IPS repository must be manually created on the local server. This involves the following steps:

1. Obtain software packages:
 - Download ISO image.
 - Copy from the default package repository.
2. Create a ZFS file system for the repository.
3. Copy the packages to the repository.
4. Set the properties like pkg/inst_root, pkg/readonly.
5. Set the preferred publisher.

The following commands are used to create the file system and copy the IPS repository from the ISO image to the local ZFS file system. **Do not run these commands here.** The repository has already been installed on the local server for you.

```
# zfs create -o compression=on rpool/export/IPS
# zfs set mountpoint=/export/IPS rpool/export
# mount -F hsfs /dev/lofi/1 /mnt
```

```
# rsync -aP /mnt/repo /export/IPS
```

The package repository is very large (approximately 7 gigabytes). Depending on the speed of your host machine, the `rsync` command can take a few hours to complete.

7. Assess the current IPS configuration on the **s11-server1** VM.

```
root@s11-server1:~# svcs application/pkg/server
STATE      STIME      FMRI
disabled   17:00:56   svc:/application/pkg/server:default
root@s11-server1:~# svcprop -p pkg/inst_root application/pkg/server
/var/pkgrepo
```

The **s11-server1** system is not currently configured as an IPS server (the service is disabled). Also, observe the default location of the IPS repository as determined by the `pkg/inst_root` property. The `/var/pkgrepo` directory is not the correct location of your local repository.

8. Determine whether the IPS service is currently available.

```
root@s11-server1:~# pkg search entire
pkg: Some repositories failed to respond appropriately:
solaris:
Unable to contact valid package repository
Encountered the following error(s):
Unable to contact any configured publishers.
This is likely a network configuration problem.
Framework error: code: 6 reason: Couldn't resolve host 'pkg.oracle.com'
URL: 'http://pkg.oracle.com/solaris/release' (happened 4 times)
```

Note: Searching for a package is a quick way of determining whether the IPS service is available. This step is especially useful because you can see the displayed URL. Based on the results shown here, the **s11-server1** system has no access to the IPS service. In this case, your publisher URL points to **s11-server1**.

9. Set the `pkg/inst_root` property to `application/pkg/server` SMF service to the location `'/export/IPS/repo'`.

```
root@s11-server1:~# svccfg -s application/pkg/server setprop
pkg/inst_root=/export/IPS/repo
root@s11-server1:~#
```

10. Set the `pkg/readonly` property of `application/pkg/server` service to `true`.

```
root@s11-server1:~# svccfg -s application/pkg/server setprop
pkg/readonly=true
```

11. Verify the `application/pkg/server` service `inst_root` property.

```
root@s11-server1:~# svcprop -p pkg/inst_root application/pkg/server
/export/IPS/repo
```

12. Refresh the application/pkg/server service.

```
root@s11-server1:~# svcadm refresh application/pkg/server
```

13. Enable the application/pkg/server service.

```
root@s11-server1:~# svcadm enable application/pkg/server
```

14. Verify that the application/pkg/server service is enabled.

```
root@s11-server1:~# svcs application/pkg/server
STATE          STIME          FMRI
online         17:00:56      svc:/application/pkg/server:default
```

15. Refresh the package repository by using the pkgrepo refresh command.

```
root@s11-server1:~# pkgrepo refresh -s /export/IPS/repo
Initiating repository refresh.
```

When you create a new package repository, you must refresh the repository catalog so that the package search operations work correctly. This may take several minutes to complete.

16. List the current package publishers.

```
root@s11-server1:~# pkg publisher

PUBLISHER      TYPE          STATUS P LOCATION
solaris        origin        online F http://pkg.oracle.com/solaris/release/
```

The command output shows the current *publisher*. A publisher is a forward domain name that identifies a person, group of persons, or an organization that publishes one or more packages. The repository type *origin* is the location of the package repository that contains both package metadata (package manifests and catalogs) and package content (package files). The default publisher URI is <http://pkg.oracle.com/solaris/release/>.

17. You are now going to configure the local server to use your newly created IPS repository. Map the publisher name *solaris* with the local IPS server URI by removing the current publisher URI (<http://pkg.oracle.com/solaris/release/>) and replacing it with the new URI (<http://s11-server1.mydomain.com/>). Display the results.

```
root@s11-server1:~# pkg set-publisher -G '*' -g
http://s11-server1.mydomain.com/ solaris

root@s11-server1:~# pkg publisher
PUBLISHER      TYPE          STATUS P LOCATION
solaris        origin        online F http://s11-server1.mydomain.com/
```

18. Test IPS on the local server by searching for the `entire` package.

```
root@s11-server1:~# pkg search entire
INDEX          ACTION    VALUE          PACKAGE
...
pkg.fmri       set       solaris/entire  pkg:/entire@0.5.11-
0.175.1.0.0.24.2
```

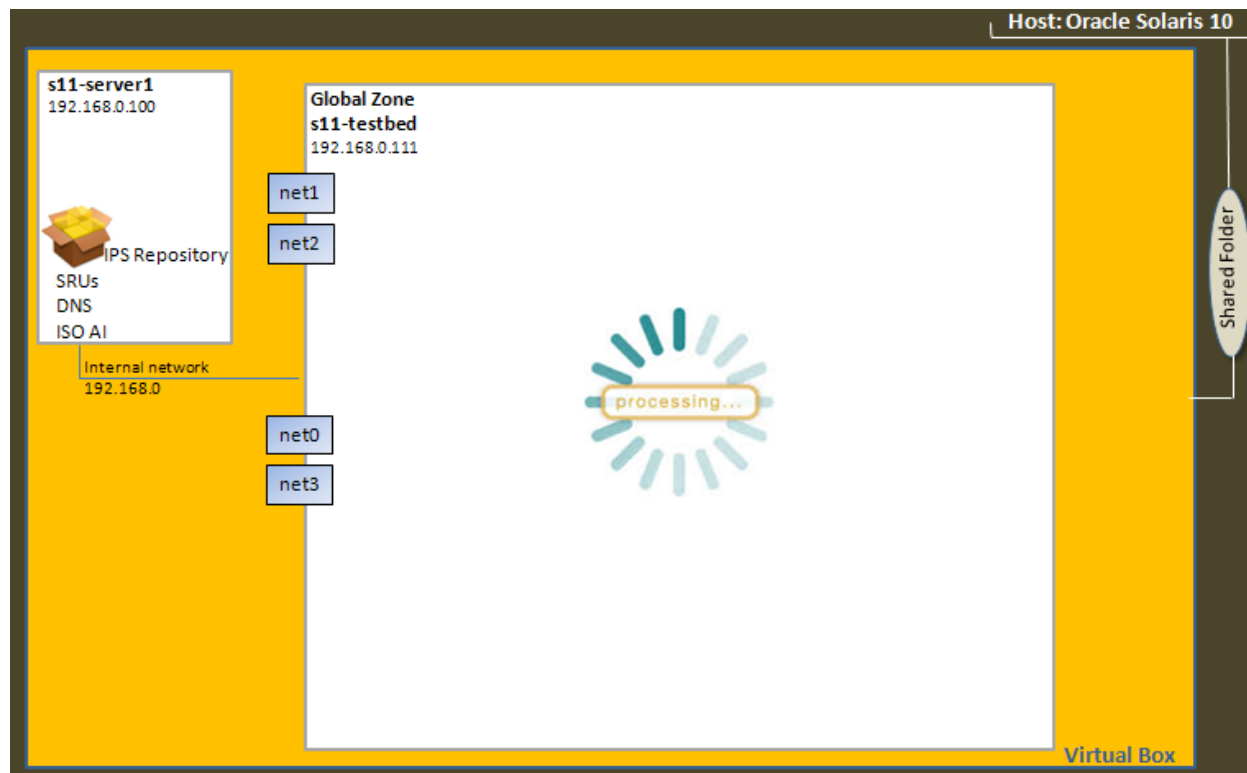
The `entire` package has been located.

Summary: You just created a local IPS repository so that the clients on the network can use this IPS repository, rather than the default repository at <http://pkg.oracle.com/>.

Practice 2-2: Configuring a Client to Access the Local IPS Server

Overview

Now that you have a local IPS repository set up on **s11-server1**, you must configure the client **s11-testbed** to access the local repository. By default, clients are configured to use the default URI <http://pkg.oracle.com/solaris/release/> for publisher **solaris**. You need to reconfigure the client to access the local IPS repository URI <http://s11-server1.mydomain.com/> for publisher **solaris**.



Tasks

1. Log in to the **s11-testbed** VM as the **oracle** user. Use the password **oracle1**.
2. Right-click the desktop and open a terminal window.
3. In the terminal window, run the **su** command to assume primary administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```

4. Verify that the client can access DNS services by resolving the IPS server host name.

```
root@s11-testbed:~# nslookup s11-server1
Server:      192.168.0.100
Address:     192.168.0.100#53

Name:       s11-server1.mydomain.com
Address:    192.168.0.100
```

Alternatively, DNS information can be obtained by using the `dig` command.

```
root@s11-testbed:~# dig s11-server1

; <<>> DiG 9.6-ESV-R7-P2 <<>> s11-server1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: SERVFAIL, id: 32476
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;s11-server1.                IN      A

;; Query time: 15 msec
;; SERVER: 192.168.0.100#53 (192.168.0.100)
;; WHEN: Sun Nov  3 22:28:18 2013
;; MSG SIZE rcvd: 29

root@s11-testbed:~#
```

5. Also verify that the client can ping the IPS server.

```
root@s11-testbed:~# ping s11-server1
s11-server1 is alive
```

6. List the current package publishers.

```
root@s11-testbed:~# pkg publisher
PUBLISHER      TYPE          STATUS P LOCATION
solaris        origin        online F http://pkg.oracle.com/solaris/release/
```

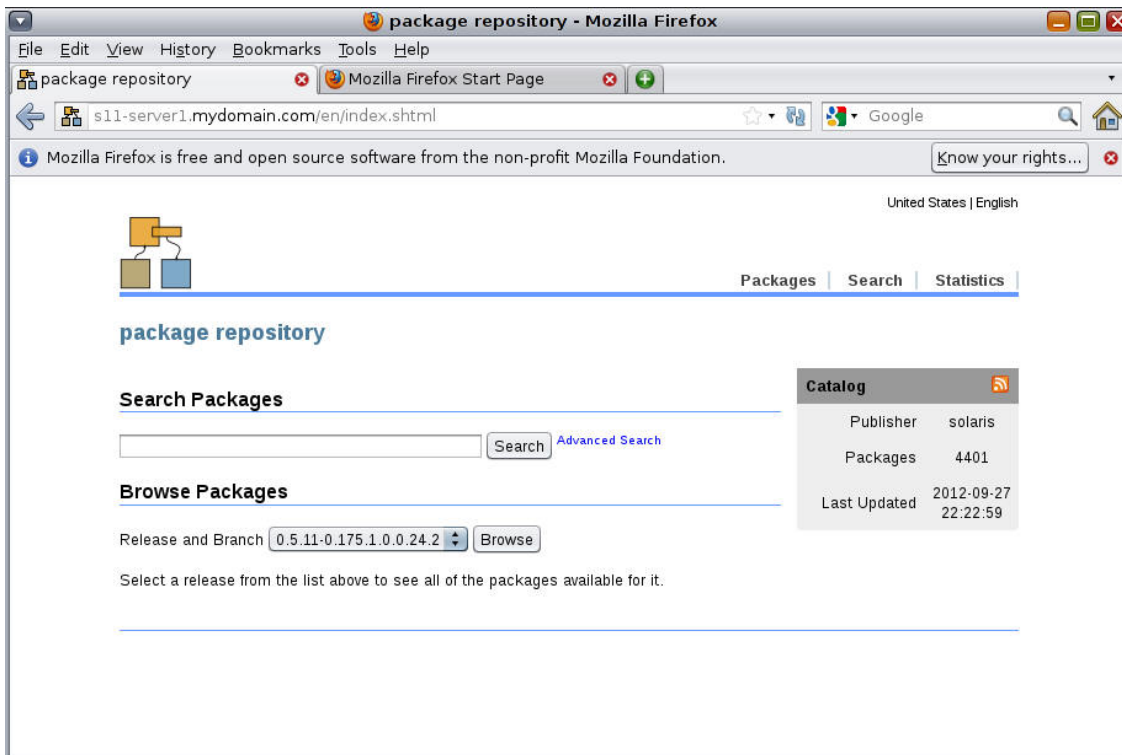
7. Remove the default publisher URI (<http://pkg.oracle.com/solaris/release/>) and add the local IPS repository URI (<http://s11-server1.mydomain.com/>) to the publisher name `solaris`.

```
root@ s11-testbed:~# pkg set-publisher -G '*' -g \
http://s11-server1.mydomain.com/ solaris
```

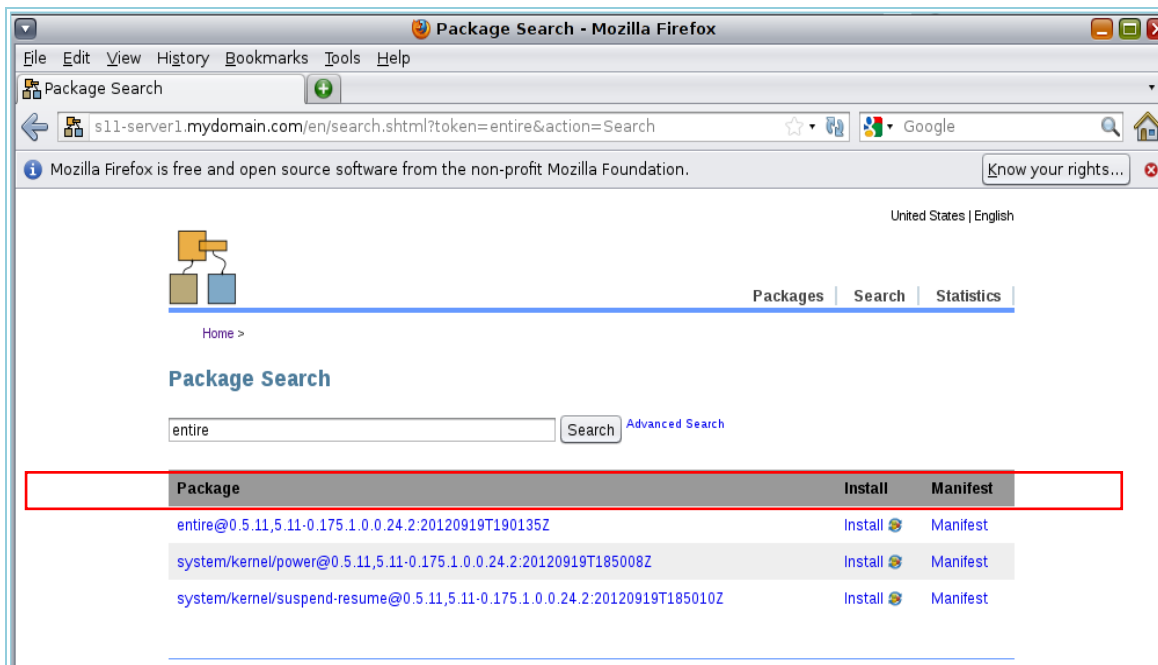
8. Verify that the publisher is set to the new URI.

```
root@s11-testbed:~# pkg publisher
PUBLISHER      TYPE          STATUS P LOCATION
solaris        origin        online F http://s11-server1.mydomain.com/
```


9. Verify client access to the IPS server by opening the `http://s11-server1.mydomain.com` URL in the Firefox browser on `s11-testbed`.



10. Using the package repository browser, search for the `entire` package.



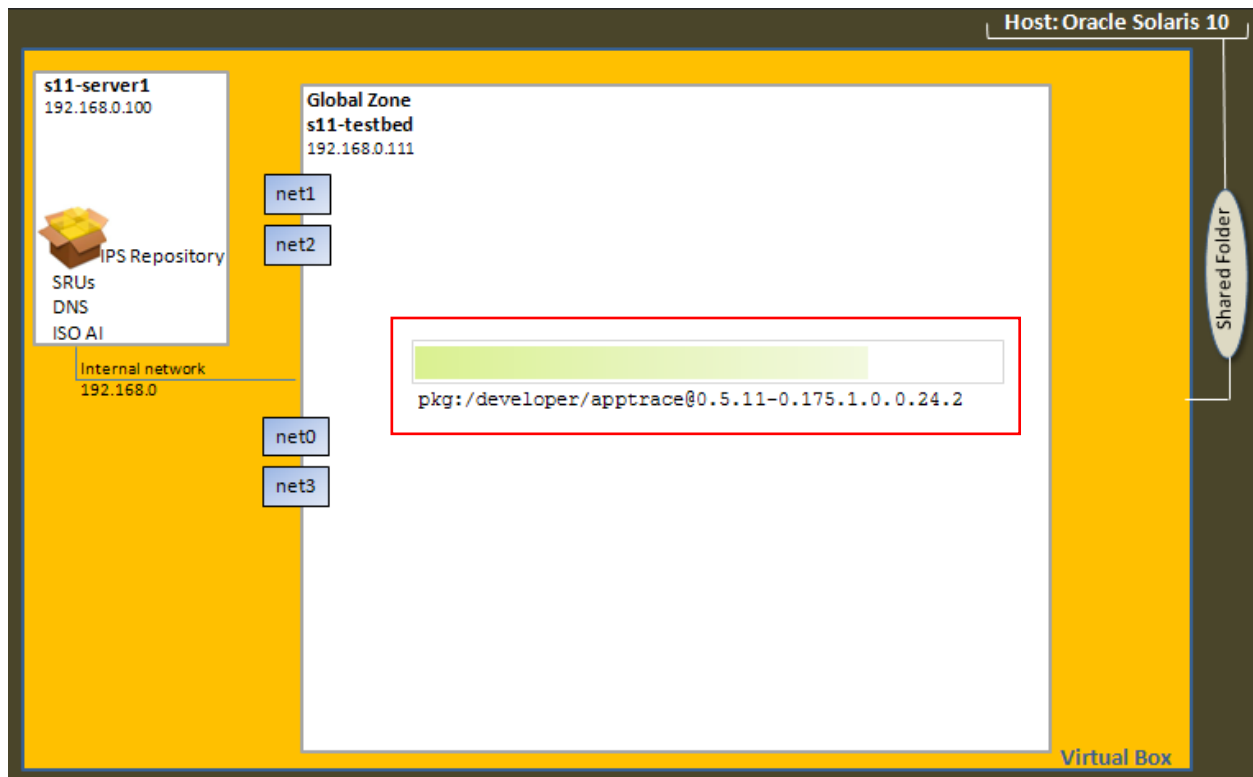
11. Close the browser.

Summary: Note that the `s11-testbed` system is able to access the `entire` package installed on the local IPS repository, `s11-server1`.

Practice 2-3: Managing Software Packages by Using the Command-Line Interface

Overview

When you install Oracle Solaris 11.1, a host of packages is installed along with the OS. However, you might require additional packages as in the **s11-testbed** environment where a range of Oracle applications will be eventually deployed and validated. For now, you need the package for the `appttrace` utility, which provides tracing facility for applications.



In this practice, you will perform the following tasks:

- Search for the `appttrace` package.
- Perform a dry run of the package installation.
- Install the package.
- Verify the package installation.
- Display information about the package.

Tasks

1. Verify that the **s11-testbed** VM is running. If not, start the VM.
2. Log in to the **s11-testbed** VM as user `oracle`, and then run the `su -` command to assume administrator privileges.

- Switch to root privileges.

```
oracle@s11-testbed:~$ su -
Password:
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```

- In a terminal window on the **s11-testbed** VM, check whether the **appttrace** software package is currently installed.

```
root@s11-testbed:~# pkg list appttrace
pkg list: no packages matching 'appttrace' installed
```

- Now, search the IPS repository for the **appttrace** software package.

```
root@s11-testbed:~# pkg search appttrace
INDEX          ACTION VALUE          PACKAGE
pkg.description set    Appttrace utility for application tracing,
including shared objects pkg:/developer/appttrace@0.5.11-
0.175.1.0.0.24.2
...
```

Note that the **appttrace** package is available on the IPS server (**s11-server1**).

- Display detailed information about the **appttrace** package from the remote repository by using the **-r** option.

```
root@s11-testbed:~# pkg info -r appttrace
      Name: developer/appttrace
      Summary: Appttrace Utility
      Description: Appttrace utility for application tracing, including
shared objects
      Category: Development/System
      State: Not installed
Publisher: solaris
...
...
...
```

Note that the package **appttrace** is available in the IPS repository but is not installed as indicated by the **State** information.

- Perform a dry run on the **appttrace** package installation.

```
root@s11-testbed:~# pkg install -nv appttrace
      Packages to install:      1
      Estimated space available: 32.16 GB
      Estimated space to be consumed: 16.62 MB
      Create boot environment:   No
      Create backup boot environment: No
      Rebuild boot archive:      No
...
...
```

The dry run shows that there is one package to be installed. The package installation will not affect the boot environment. Neither will the currently installed packages be changed. The `pkg` command uses Fault Management Resource Identifiers (FMRI), or portions of FMRI, to operate on packages. Note that the FMRI includes the package publisher, name, and version.

8. Install the `appttrace` package.

```
root@s11-testbed:~# pkg install appttrace
      Packages to install: 1
      Create boot environment: No
      Create backup boot environment: No

DOWNLOAD                                PKGS           FILES        XFER (MB)   SPEED
Completed                             1/1           10/10         0.1/0.1    548k/s

PHASE                                ITEMS
Installing new actions                29/29
Updating package state database        Done
Updating image state                   Done
Creating fast lookup database          Done

Observe that the package installation is successful.
```

9. Verify the `appttrace` package installation.

```
root@s11-testbed:~# pkg verify -v appttrace
PACKAGE                                STATUS
pkg://solaris/developer/appttrace      OK
```

Summary: Note that the package for the `appttrace` utility has been installed successfully.

Note: Do not perform the following steps. They are only for your reference.

Additional Note: If in future, you wish to remove the `appttrace` package from the system image on your host, you can perform the following steps:

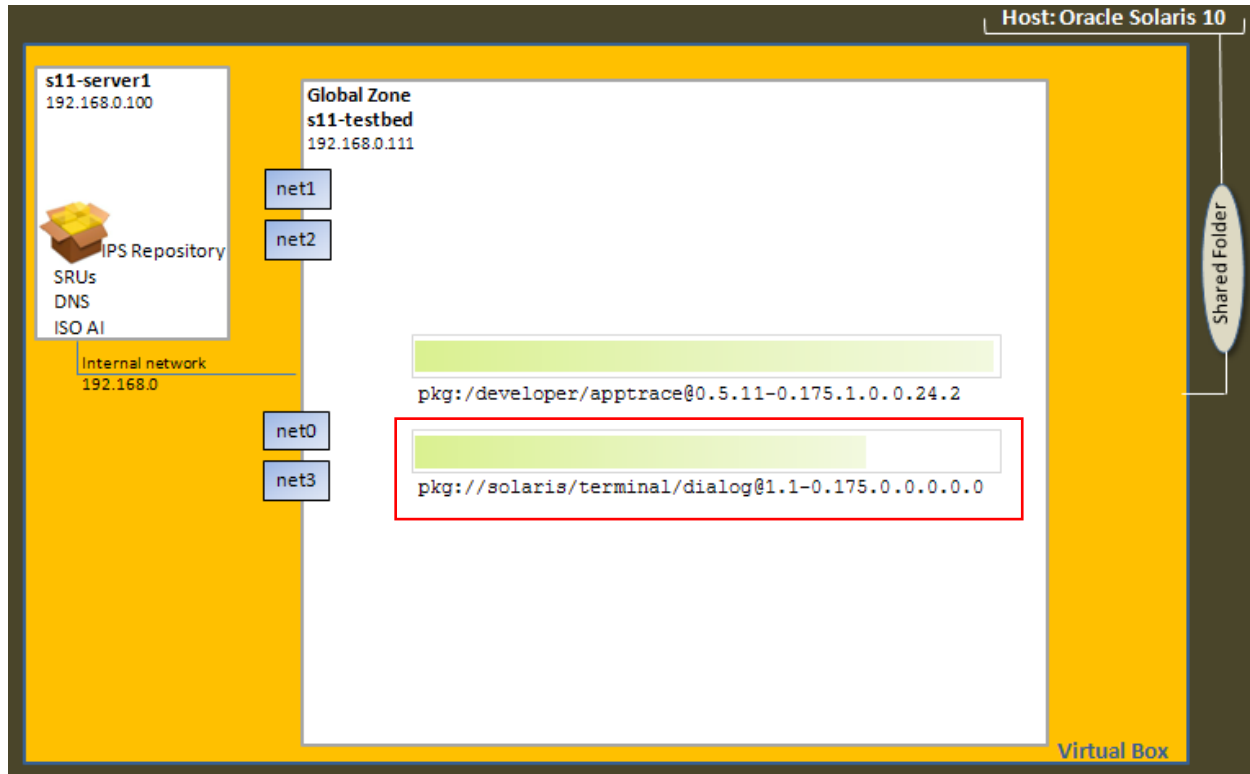
```
root@s11-testbed:~# pkg uninstall appttrace
      Packages to remove: 1
      Create boot environment: No
      Create backup boot environment: No

PHASE                                ITEMS
Removing old actions                  26/26
Updating package state database        Done
Updating package cache                 1/1
Updating image state                   Done
Creating fast lookup database          Done
root@s11-testbed:~# pkg list appttrace
pkg list: no packages matching 'appttrace' installed
```

Practice 2-4: Managing Software Packages by Using Package Manager

Overview

Just as you installed the `appttrace` package, you need to install the package for the `dialog` utility. The `dialog` utility provides a useful interface in the form of a dialog box that you can use for testing or verifying your scripts, which you could be using at a later time. Using Package Manager, start by checking whether the `dialog` package is already installed on your system. If not, check for its availability in the local IPS repository. Finally, install the package and verify that it was successfully installed.



In this practice, you will perform the following tasks:

- Launch Package Manager.
- Search for the `dialog` package.
- Install the package.
- Verify the package installation.
- Display information about the package and its contents.

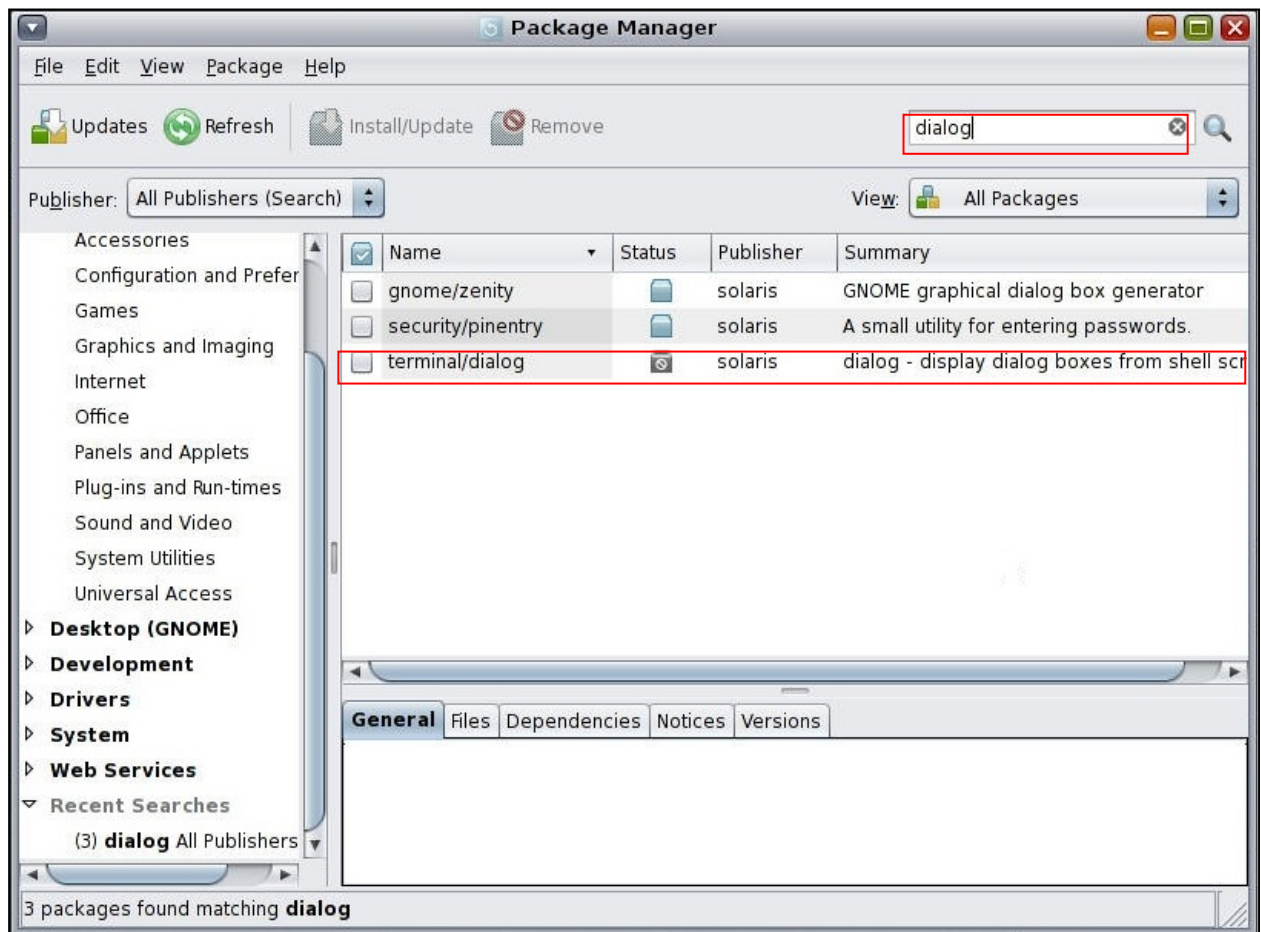
Tasks

1. Verify that the **s11-testbed** VM is running. If not, start the VM.

- Log in to the **s11-testbed** VM as user `oracle`, and then run the `su -` command to assume primary administrator privileges. Use the password `oracle1`.

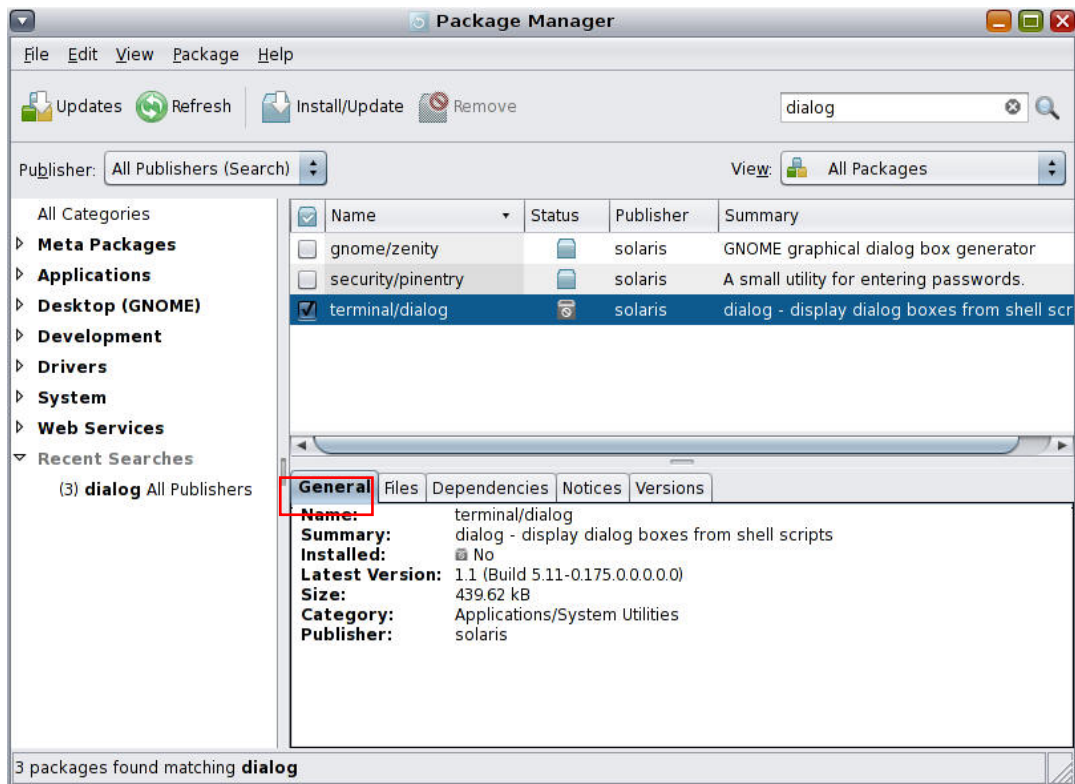
```
oracle@s11-testbed:~$ su -
password:oracle1
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```

- On the desktop background, double-click the **Add More Software** icon.
- In the **File** menu, select **Manage Publishers**.
- In the **Manage Publisher** dialog box, verify that the package publisher is enabled. Also, verify that the **Origin** points to the IPS server. Click **OK**.
- In the **Package Manager** search textbox, enter `dialog` and press **Enter**.

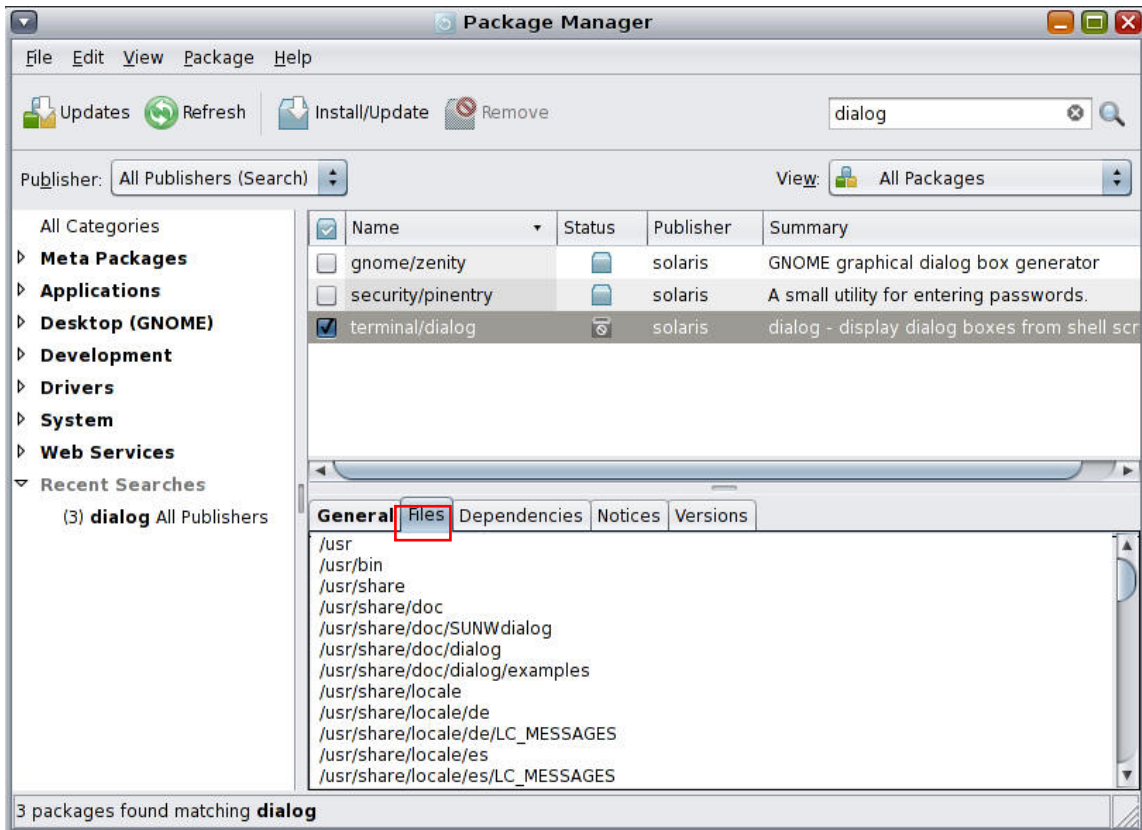


Note: The status icon indicates that the `dialog` package is not currently installed on this system.

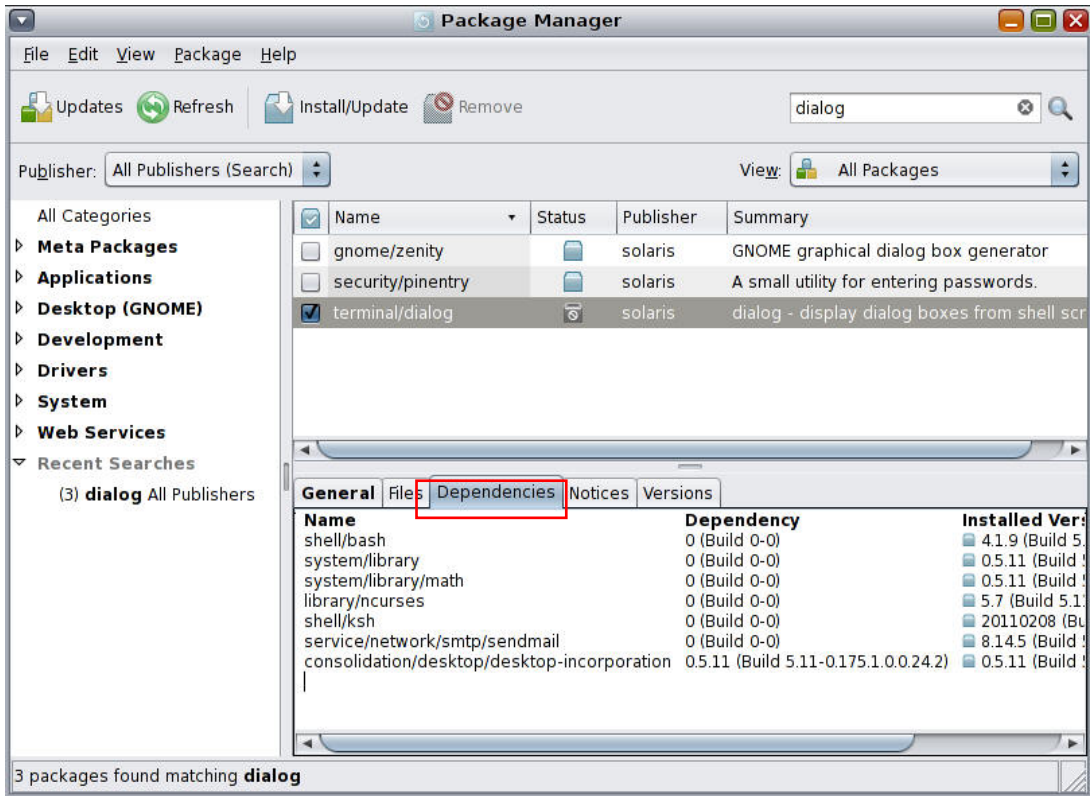
7. Select the dialog package. Note the content of the **General** tab at the bottom of the display. This information is derived from the dialog manifest.



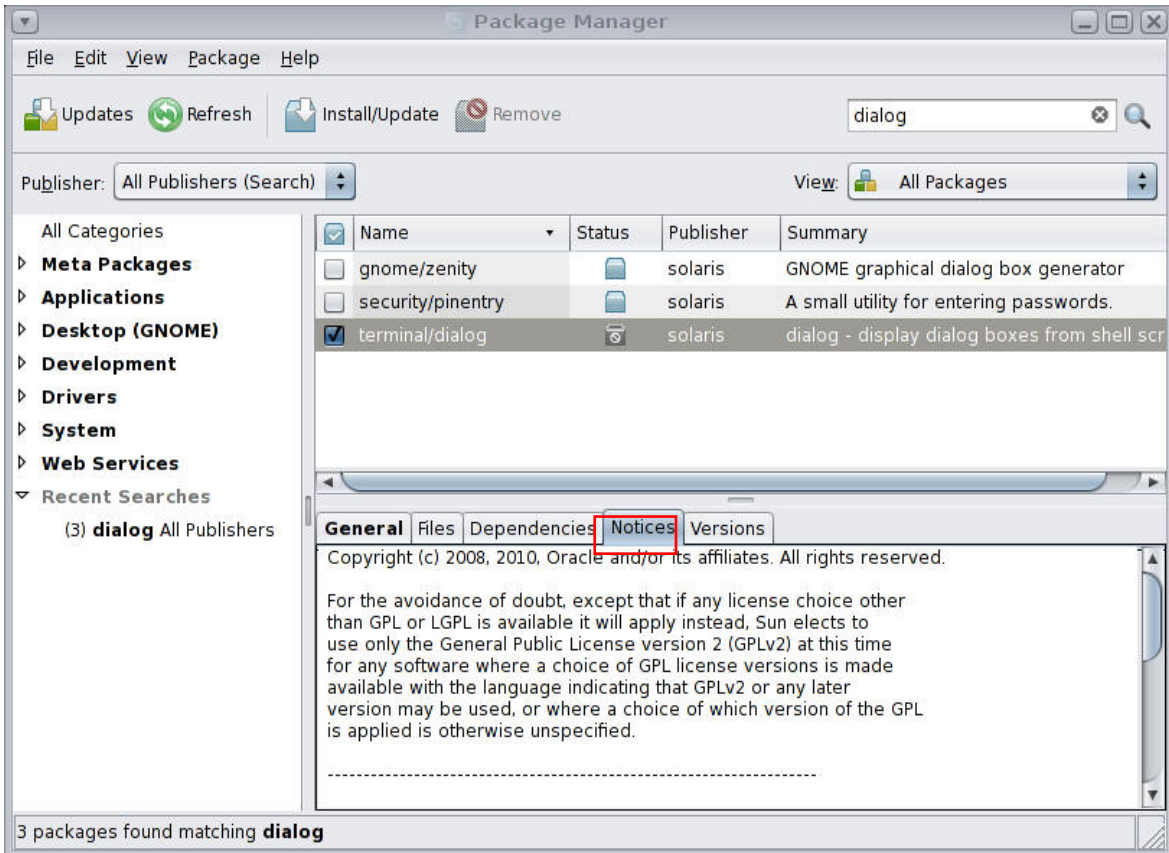
8. Click the **Files** tab to view the files called out in the dialog manifest.



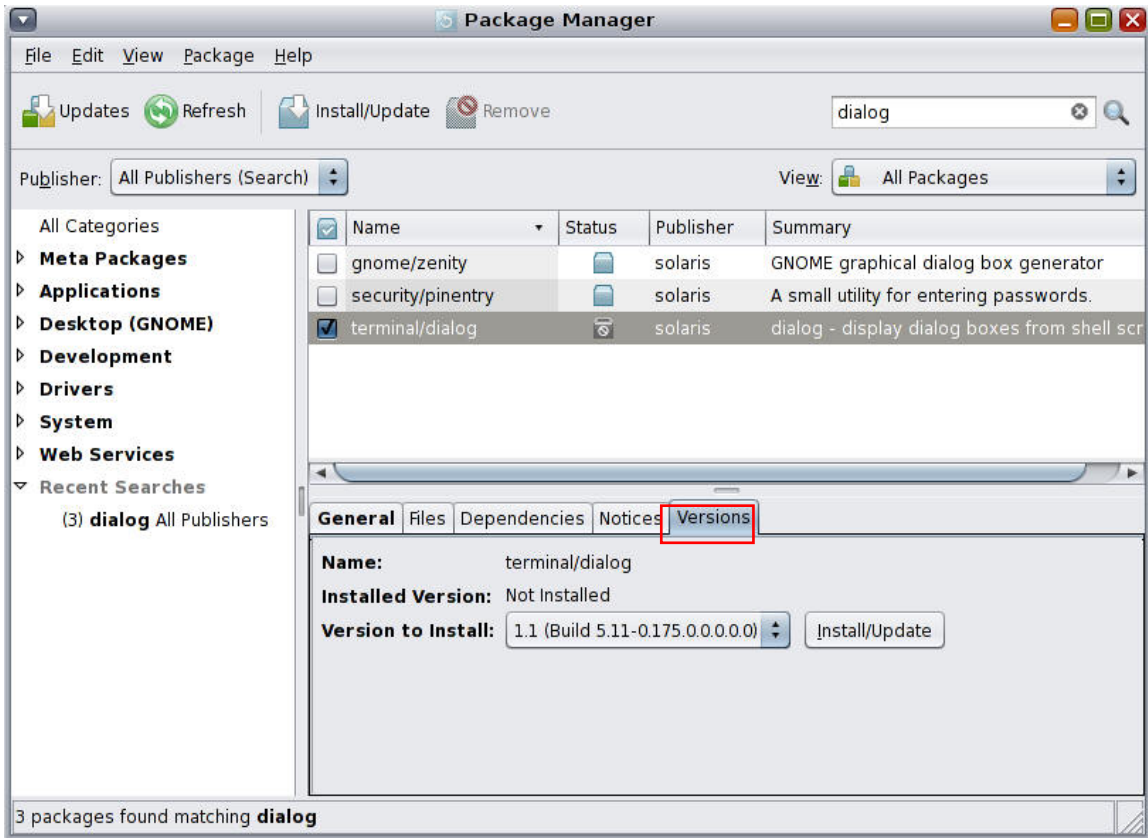
9. Click the **Dependencies** tab.



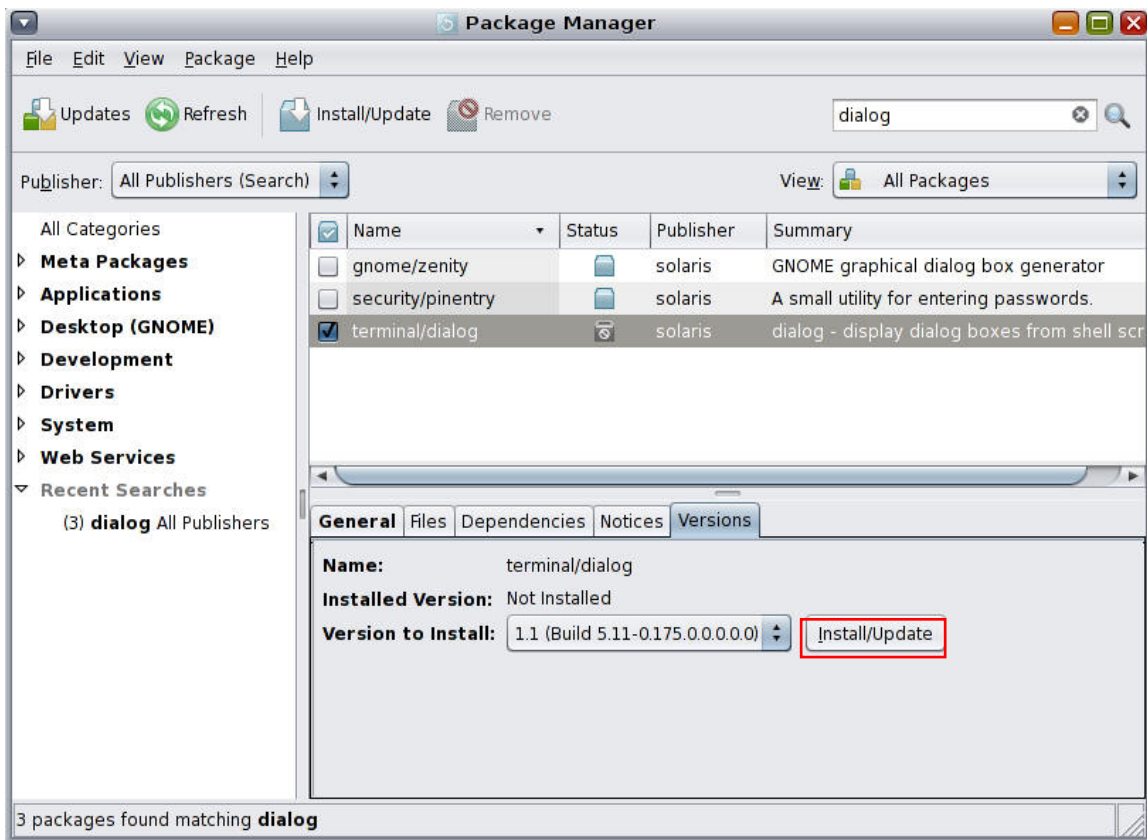
10. Click the **Notices** tab.



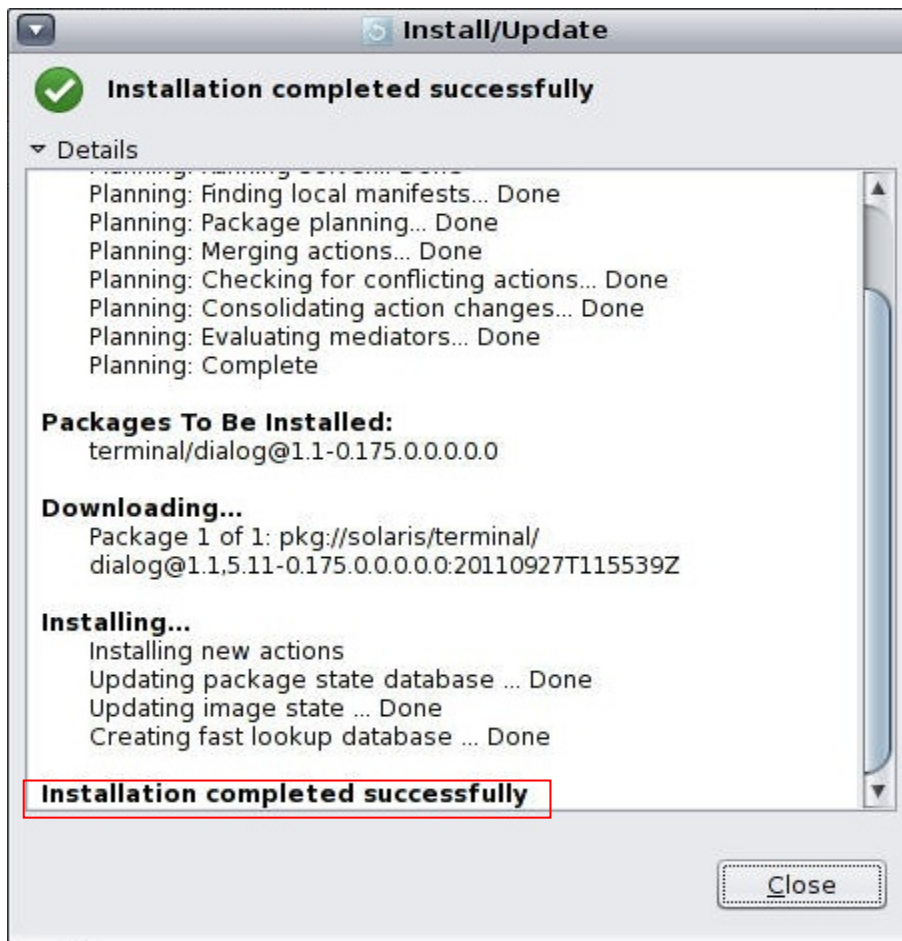
11. Click the **Versions** tab.



12. Select the package. Click the **Install/Update** button. Then click **Proceed** in the **Install Confirmation** dialog box.



13. Verify that the `dialog` package installed successfully. When done, close the **Install/Update** dialog box and **Package Manager**.



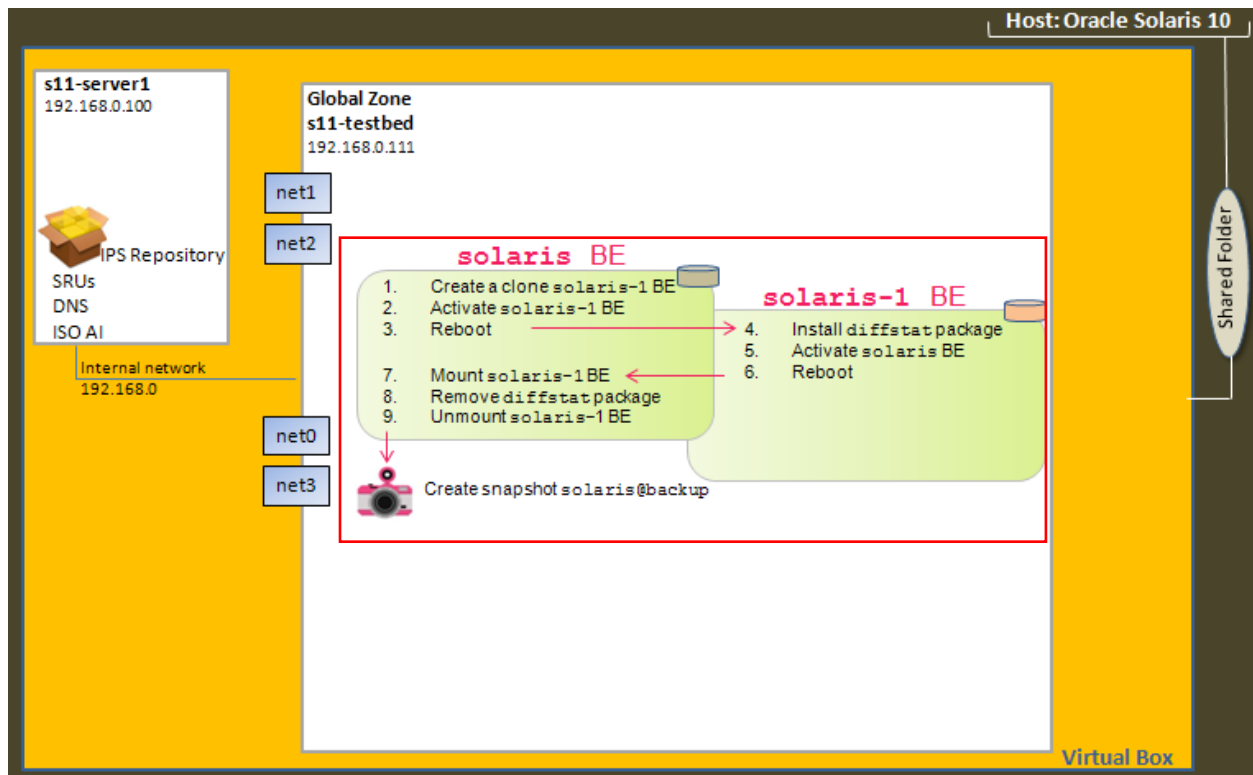
Summary: The `dialog` package has been successfully installed by using Package Manager.

Practice 2-5: Managing Multiple Boot Environments

Overview

The need to maintain multiple BEs becomes more pronounced as you continue to perform more complex activities on the system. You cannot risk system corruption following an upgrade or update activity. It is a good practice to create new BEs at the end of each significant activity so that you can revert to a previous BE if the situation so demands. A little hands-on around the various activities involved in managing BEs at this stage will be very useful in maintaining your system against any potential loss in the future.

In this practice, you will configure and manage multiple BEs.



Notes

- Some command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.

Tasks

1. Open a terminal window on the **s11-testbed** VM, and list the current BEs using the `beadm list` command.

```
root@s11-testbed:~# beadm list
BE          Active Mountpoint Space  Policy Created
--          -
solaris     NR      /           2.84G  static 2012-11-17 08:47
```

Note that the current BE (`solaris`) is the default BE that is created during the OS installation.

The `Active` field indicates whether the boot environment is active now (N) and active on reboot (R).

2. Clone the current active BE. Name the clone `solaris-1`.

```
root@s11-testbed:~# beadm create solaris-1
```

3. List the current BEs.

```
root@s11-testbed:~# beadm list
BE          Active Mountpoint Space  Policy Created
--          -
solaris     NR      /           2.84G  static 2012-11-30 08:47
solaris-1   -      -           164.0K  static 2013-08-15 07:01
```

4. Activate the `solaris-1` BE. Display the list of BEs.

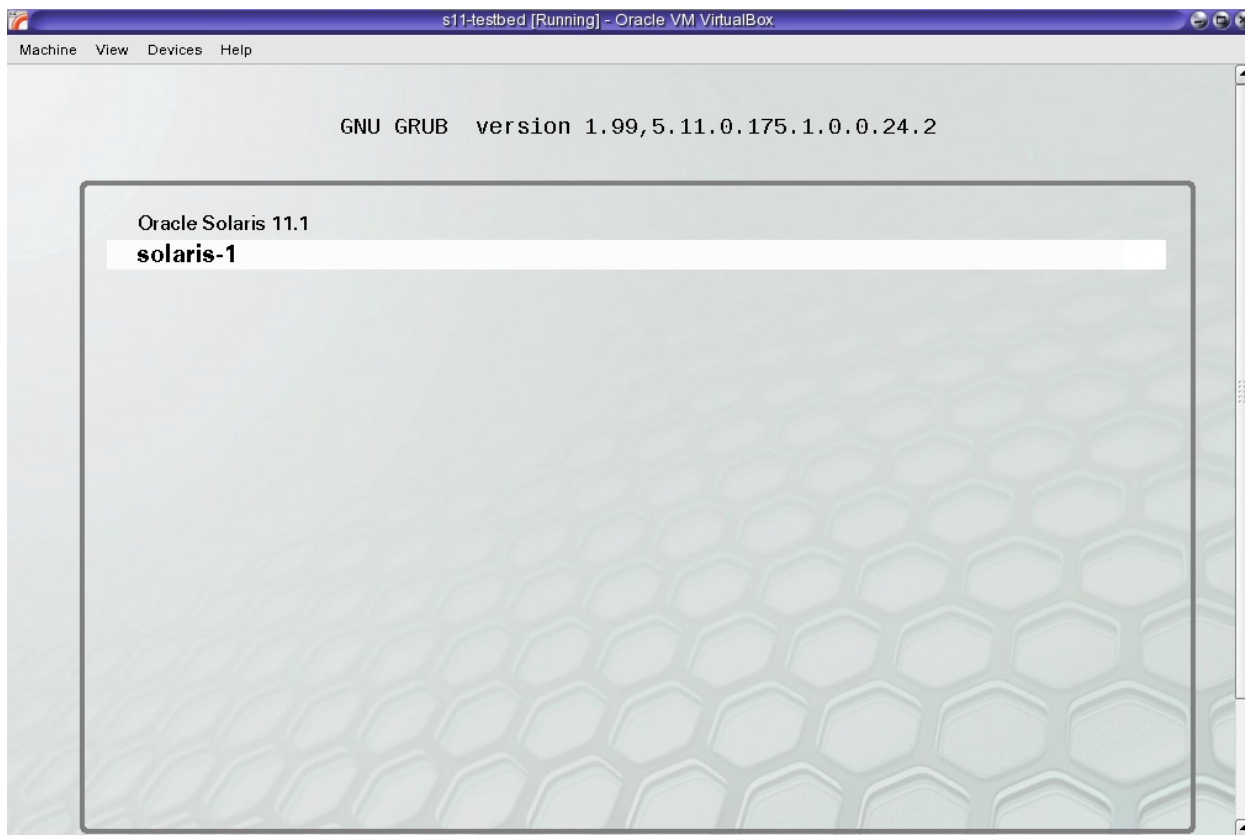
```
root@s11-testbed:~# beadm activate solaris-1
root@s11-testbed:~# beadm list
BE          Active Mountpoint Space  Policy Created
--          -
solaris     N      /           469.0K  static 2012-11-30 08:47
solaris-1   R      -           2.84G  static 2013-08-15 07:01
```

Note that `solaris-1` is pending activation on reboot.

5. Reboot the **s11-testbed** VM.

```
root@s11-testbed:~# init 6
```

The activation process takes some time to store the data in the partition. Notice that **solaris-1** is now the default boot entry in the GRUB menu.



6. After **s11-testbed** has rebooted, log in as the **oracle** user and **su** to **root**.
7. In a terminal window, list the current BEs.

```
root@s11-testbed:~# beadm list
BE          Active Mountpoint Space  Policy Created
--          -
solaris     -      -           4.60M  static 2012-11-30 08:47
solaris-1 NR /           2.89G  static 2013-08-15 07:01
```

Note that the **solaris-1** image is now active.

8. Verify that the **diffstat** package is not currently installed on the new active BE.

```
root@s11-testbed:~# pkg list diffstat
pkg list: no packages matching 'diffstat' installed
```

9. Install the **diffstat** package on the new active BE.

```
root@s11-testbed:~# pkg install diffstat

Packages to install: 1
Create boot environment: No
Create backup boot environment: No

DOWNLOAD                                PKGS          FILES        XFER (MB)   SPEED
```



```
Completed                      1/1                      6/6          0.0/0.0  978k/s

PHASE                          ITEMS
Installing new actions          24/24
Updating package state database Done
Updating image state            Done
Creating fast lookup database   Done
root@s11-testbed:~#
```

10. Activate the **solaris** BE. Display the list of BEs.

```
root@s11-testbed:~# beadm activate solaris
root@s11-testbed:~# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      R      -           2.84G  static 2012-11-30 08:47
solaris-1    N      /           72.06M  static 2013-08-15 07:01
```

Note that the **solaris** BE is pending activation on reboot.

11. Reboot the **s11-testbed** VM. After **s11-testbed** has rebooted, log in as the **oracle** user and **su** to **root**.

```
root@s11-testbed:~# init 6
```

12. Verify that the **solaris** image is now active and that the **diffstat** package is not available in this BE.

```
root@s11-testbed:~# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      NR      /           2.89G  static 2012-11-30 08:47
solaris-1    -      -           76.03M  static 2013-08-15 07:01
root@s11-testbed:~# pkg list diffstat
pkg list: no packages matching 'diffstat' installed
```

13. Mount the inactive BE, **solaris-1**.

```
root@s11-testbed:~# mkdir -p /solaris-1
root@s11-testbed:~# beadm mount solaris-1 /solaris-1
root@s11-testbed:~# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      NR      /           2.89G  static 2012-11-30 08:47
solaris-1    -      /solaris-1  76.03M  static 2013-08-15 07:01
```

14. Verify that the **diffstat** package is installed in the inactive BE, **solaris-1**.

```
root@s11-testbed:~# pkg -R /solaris-1 verify -v diffstat
PACKAGE                      STATUS
pkg://solaris/text/diffstat   OK
```

15. Remove the `diffstat` package from the mounted inactive BE, `solaris-1`.

```
root@s11-testbed:~# pkg -R /solaris-1 uninstall diffstat
Packages to remove: 1

PHASE                                ITEMS
Removing old actions                  19/19
Updating package state database       Done
Updating package cache                1/1
Updating image state                  Done
Creating fast lookup database         Done
root@s11-testbed:~# pkg -R /solaris-1 list diffstat
pkg list: no packages matching 'diffstat' installed
```

16. Unmount the inactive BE `solaris-1`.

```
root@s11-testbed:~# beadm unmount solaris-1
```

17. Create a snapshot of the `solaris` BE. Name the snapshot `solaris@backup`.

```
root@s11-testbed:~# beadm create solaris@backup
```

18. Display the list of snapshots associated with the `solaris` BE.

```
root@s11-testbed:~# beadm list -a solaris
```

BE/Dataset/Snapshot	Active	Mountpoint	Space	Policy	Created

solaris					
rpool/ROOT/solaris	NR	/	2.24G	static	2012-12-17 18:37
rpool/ROOT/solaris/var	-	/var	111.96M	static	2012-12-17 18:37
rpool/ROOT/solaris/var@2013...	-	-	718.5K	static	2013-08-15 20:48
rpool/ROOT/solaris/var@backup	-	-	23.0K	static	2013-08-15 22:16
rpool/ROOT/solaris/var@install	-	-	19.00M	static	2012-12-17 18:43
rpool/ROOT/solaris@2013...	-	-	15.01M	static	2013-08-15 20:48
rpool/ROOT/solaris@backup	-	-	0	static	2013-08-15 22:16
rpool/ROOT/solaris@install	-	-	55.87M	static	2012-12-17 18:43
...					
...					

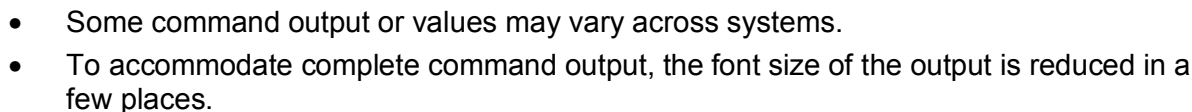
```
root@s11-testbed:~#
```

Summary: You can see the `solaris@backup` snapshot listed under the `solaris` BE. Know that you can further create a new BE from a snapshot, for example, `beadm create -e solaris@backup solaris-2`.

Practices for Lesson 3: Administering Services by Using SMF

Chapter 3

Below is the schematic representation of the environment you are now in. Note that the infrastructure thus far is configured with an IPS repository and the **s11-testbed** system has access to the repository. You will now be involved in not just configuring some user-defined services but also modifying the profile of existing services for specific needs. Finally, you will get a preview of some restore and repair situations—a glimpse of the debugging and troubleshooting activities you will encounter at the workplace.



Practice 3-1: Configuring SMF Services

Overview

Apart from the stock of services that comes with the OS, you can create your own services to address the specific needs of your infrastructure. For a start, you create a service called `crmsvc` that is meant to assist you in monitoring the CRM processes. Although for now you do not define the entire functionality of the service, you will create a fully functional service that is operational and can be later enhanced.

In addition, you will modify certain environment variables and properties of actively running services. For instance, these modifications will help determine any memory leaks caused by running programs and monitor telnet transactions by tracing the traffic between systems.

In this practice, you will perform the following tasks:

- Create the `crmsrv` service.
- Modify the service configuration.
 - Change an environment variable of a service.
 - Change a property for an `inetd`-controlled service.

Task 1: Create the `crmsrv` Service

1. Verify that the **s11-testbed** VM is running. If it is not, start it now.
2. Log in to the **s11-testbed** VM as user `oracle`. Use the password `oracle1`. Assume administrative privileges.

```
oracle@s11-testbed:~$ su -
Password:oracle1
Oracle Corporation          SunOS 5.11 11.1          September 2012
root@s11-testbed:~#
```

3. Verify that the user `sadmin` exists. If not, create the user `sadmin` and then confirm that the user has been created.

```
root@s11-testbed:~# listusers
noaccess      No Access User
nobody        NFS Anonymous Access User
nobody4       SunOS 4.x NFS Anonymous Access User
oracle        oracle
sadmin        super admin
root@s11-testbed:~#
```

Alternatively, user's information can be obtained using below command.

```
root@s11-testbed:~# tail /etc/passwd
svctag:x:95:12:Service Tag UID:/:
unknown:x:96:96:Unknown Remote UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
aiuser:x:61:61:AI User:/:
pkg5srv:x:97:97:pkg(5) server UID:/:
oracle:x:100:10:oracle:/home/oracle:/usr/bin/bash
sadmin:x:1000:10:super admin:/export/home/sadmin:/bin/bash
```

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

```
root@s11-testbed:~#
```

Note: A non-administrative user `sadmin` was already created and exists on the system.

If `sadmin` does not exist, run the following command:

```
root@s11-testbed:~# useradd -u 1000 -g 10 -d /export/home/sadmin -m -s /bin/bash -c "super admin" sadmin
```

4. As the `sadmin` user, create the `smf` directory in the home directory. Create a file called `monitor.crm` with the content shown below. Finally, grant `execute` permission on the script.

```
root@s11-testbed:~# su - sadmin
Oracle Corporation          SunOS 5.11 11.1          September 2012
sadmin@s11-testbed:~$ pwd
/export/home/sadmin
sadmin@s11-testbed:~$ mkdir smf
sadmin@s11-testbed:~$ ls
local.cshrc  local.login  local.profile  smf
sadmin@s11-testbed:~$ cd smf
sadmin@s11-testbed:~/smf$ vi monitor.crm
sadmin@s11-testbed:~/smf$ cat monitor.crm
#!/bin/sh
echo "crm monitoring service" > /export/home/sadmin/smf/crmrep

sadmin@s11-testbed:~/smf$ chmod 774 monitor.crm
```

5. Exit the `sadmin` user account to return to the administrative user to configure the service. Copy an existing service to serve as a template by using the `svccfg` command.

```
sadmin@s11-testbed:~/smf$ exit
root@s11-testbed:~# svccfg export system/utmp > /var/svc/manifest/site/crmsvc.xml
```

Instead of creating the manifest file from scratch, you can leverage from a template.

6. Edit the `crmsvc.xml` file to match the content displayed below. Your file should match this content *exactly*, so ensure that you delete all unnecessary tags from the template.

```
root@s11-testbed:~# vi /var/svc/manifest/site/crmsvc.xml
root@s11-testbed:~# more /var/svc/manifest/site/crmsvc.xml
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM
'/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='crmsvc'>
  <service name='site/crmsvc' type='service' version='1'>
    <create_default_instance enabled='false' />
    <single_instance/>
  </service>
</service_bundle>
```

[Ensure that you delete the dependency and dependent tags.]

```
<exec_method name='start' type='method'
exec='/export/home/sadmin/smf/monitor.crm'
timeout_seconds='60'/>
<exec_method name='stop' type='method' exec=':true'
timeout_seconds='60'/>
```

[Ensure that you delete the stability value and template tags and their associated information]

```
<property_group name='startd' type='framework'>
  <propval name='duration' type='astring' value='transient'/>
</property_group>
</service>
</service_bundle>
```

After editing, the manifest for `crmsvc` service should look like this. Review the content for any missing XML tags and typing errors.

7. Validate the manifest file by using the `svccfg validate` command.

```
root@s11-testbed:~# svccfg validate /var/svc/manifest/site/crmsvc.xml
```

Unless there are any spelling mistakes, the `validate` command should run smoothly.

8. Make the manifest available to SMF by using the `svcadm restart` command.

```
root@s11-testbed:~# svcadm restart system/manifest-import
```

The service you created is in an SMF standard manifest directory. You can therefore just restart the manifest service. This imports the newly created service. You do not have to import the service individually. This is the recommended practice.

9. Display the service by using the `svcs` command. If it is disabled, enable it by using the `svcadm` command.

```
root@s11-testbed:~# svcs crmsvc
disabled      13:14:07 svc:/site/crmsvc:default
root@s11-testbed:~# svcadm enable /site/crmsvc
root@s11-testbed:~# svcs crmsvc
STATE          STIME      FMRI
online         13:43:36  svc:/site/crmsvc:default
```

Is your service enabled and online? Yes.

- Verify that the command `echo` was executed by using the new service.

```
root@s11-testbed:~# cat /export/home/sadmin/smf/crmrep
crm monitoring service
```

Summary: The action you specified in the `monitor.crm` file was executed by bringing up the service resulting in echoing the above string to the `crmrep` file. This is how you execute a program as a service.

Task 2: Modify Service Configuration

You have the flexibility to modify service environment variables, network service properties, and processes. You just need to know how to modify the various properties of a service to address your need. In this case, you modify the `cron` service to monitor memory leaks.

Task: Change an Environment Variable of a Service

- Verify that the **s11-testbed** VM is running. If it is not, start it now.
- Log in to the **s11-testbed** VM as the user `oracle`. Use the password `oracle1`. Assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation      SunOS 5.11    11.1          September 2012
root@s11-testbed:~#
```

- Check whether the `cron` service is running by using the `svcs` command.

```
root@s11-testbed:~# svcs system/cron
STATE      STIME      FMRI
online     Oct_22     svc:/system/cron:default

The cron service is up and running.
```

- Use the `svccfg` command to modify the `UMEM_DEBUG` and `LD_PRELOAD` memory environment variables for the `cron` service to monitor memory leaks.

```
root@s11-testbed:~# svccfg -s system/cron:default setenv UMEM_DEBUG default
root@s11-testbed:~# svccfg -s system/cron:default setenv LD_PRELOAD libumem.so

The two environment variables are configured for debugging memory leaks while the
cron service is executing a program.
```

- Refresh and restart the `cron` service by using the `svcadm` command to make the changes effective.

```
root@s11-testbed:~# svcadm refresh system/cron
root@s11-testbed:~# svcadm restart system/cron
```


6. Verify that the environment variables have been modified.

Note: Use the *backtick* key on the keyboard to enclose the `pgrep` command. Look for the backtick below the tilde (~) key on the keyboard.

```
root@s11-testbed:~# pargs -e `pgrep -f /usr/sbin/cron`
1593: /usr/sbin/cron
...
...
envp[10]: LD_PRELOAD=libumem.so
...
...
envp[19]: UMEM_DEBUG=default
envp[20]: A__z="*SHLVL
```

Your display may be slightly different.

Summary: The environment variables, `envp[10]` and `envp[19]` reflect the changes that you made.

Practice 3-2: Modifying a Service Profile

Overview

At system startup, many services are started. Of these, some are required, others not. To allow unwanted services to start would entail system resources, such as processing bandwidth and additional time. In big data environments, this would be unsound.

You need to ensure that only required services are enabled at system startup. For the testbed environment, for now you do not require the printer service (`cups/scheduler`) to be enabled by default. You will therefore modify the service profile to ensure that the printer service is disabled at startup.

In this practice, you will create an SMF profile and modify the manifest to address your need.

Tasks

1. Verify that the **s11-testbed** VM is running. If not, start it at this time.
2. Log in to the **s11-testbed** VM as the user `oracle`. Use the password `oracle1`.
3. In the terminal window, run the `su -` command to assume administrator privileges.

```
oracle@s11-testbed:~$ su -  
Password: oracle1  
Oracle Corporation          SunOS 5.11 11.1          September 2012  
root@s11-testbed:~#
```

4. Check the current status of the `cups/scheduler` service by using the `svcs` command.

```
root@s11-testbed:~# svcs cups/scheduler  
STATE      STIME      FMRI  
online     Oct_22     svc:/application/cups/scheduler:default  
  
Currently, the service is enabled.
```

5. Copy the currently active SMF profile into a file called `profile.xml` by using the `svccfg extract` command.

```
root@s11-testbed:~# svccfg extract > profile.xml
```

6. Modify the extracted file `profile.xml` by using the `vi` editor. Change the enabled property of `application/cups/scheduler` service from `true` to `false`.

```
root@s11-testbed:~# vi profile.xml
root@s11-testbed:~# more profile.xml
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM
'/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='profile' name='profile'>
...
...
...
<service name='application/cups/scheduler' type='service' version='0'>
  <create_default_instance enabled='false'>
    </service>
...
...
...
```

7. Apply the modified profile by using the `svccfg` command.

```
root@s11-testbed:~# svccfg apply profile.xml
```

Note: Allow the OS to apply the changes. It takes a few minutes.

```
root@s11-testbed:~# svcs cups/scheduler
STATE          STIME          FMRI
disabled       16:48:33      svc:/application/cups/scheduler:default
```

Summary: Notice that the `cups/scheduler` service is disabled.

Additional Note: If you need to bring the `cups/scheduler` service online again, refresh and then enable the service.

```
root@s11-testbed:~# svcadm refresh cups/scheduler
root@s11-testbed:~# svcadm enable cups/scheduler
root@s11-testbed:~# svcs cups/scheduler
STATE          STIME          FMRI
online         16:50:15      svc:/application/cups/scheduler:default
```

The service is once again enabled.

Practice 3-3: Restoring a Service

Overview

You will now be exposed to some restore and recovery procedures. You need to know a little troubleshooting to start with because you might encounter these situations as you continue to consolidate the testbed environment.

In this practice, you will perform the following tasks:

- Restore a service in the maintenance state.
- Revert to a previous SMF snapshot.
- Repair a corrupt repository.

Task 1: Restore a Service in the maintenance State

Just as with the `crmsvc` service, there is yet another user-defined service called `logapp` configured in your system. The `logapp` service is designed to capture application-related logs. While running it, you observe that the service has gone into the maintenance state. You need to bring the service back up again.

1. Verify that the **s11-testbed** VM is running. If not, start it at this time.
2. Log in to the **s11-testbed** VM as the user `oracle`. Use the password `oracle1`.
3. Right-click the desktop and open a terminal window.
4. In the terminal window, run the `su -` command to assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation          SunOS 5.11 11.1          September 2012
root@s11-testbed:~#
```

5. Check whether the `logapp` service is running by using the `svcs` command.

```
root@s11-testbed:~# svcs logapp
STATE          STIME      FMRI
maintenance    Oct_22     svc:/site/logapp:default
```

Notice that the service is in the maintenance state.

6. Clear the maintenance state by using the `svcadm clear` command.

```
root@s11-testbed:~# svcadm clear logapp
root@s11-testbed:~# svcs logapp
STATE          STIME      FMRI
maintenance    23:53:25   svc:/site/logapp:default
```

The service continues to be in the maintenance state. When a service is placed in the maintenance mode, SMF is unable to bring it up. A system administrator has to debug the problem.

7. Obtain some debugging details by using the `svcs` command with the `-xv` option.

```
root@s11-testbed:~# svcs -xv logapp
svc:/ site/crmsvc:default (?)
State: maintenance since December 15, 2012 08:22:41 PM UTC
```

```
Reason: Start method failed repeatedly, last exited with status 127
See: http://support.oracle.com/msg/SMF-8000-KS
See: /var/svc/log/site-logapp:default.log
Impact: This service is not running
```

The display indicates that there is a problem with the `start` method, which exited with status 127. You can get more details in the service log.

```
root@s11-testbed:~# tail /var/svc/log/site-logapp:default.log
/usr/sbin/sh[1:exec: /usr/smf/monitor.cr: not found
Dec 15 08:22:41 Method "start" exited with status 127.
...
root@s11-testbed:~# cd
```

Observe that the log spells out that it cannot execute your `monitor.cr` script.

8. Edit the `logapp.xml` file to correct the typing error. Refer to previous steps for editing content.

```
root@s11-testbed:~# cd /var/svc/manifest/site
root@s11-testbed:/var/svc/manifest/site# vi logapp.xml
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM
'/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='logapp'>
  <service name='site/logapp' type='service' version='1'>
    <create_default_instance enabled='false'>
    <single_instance/>
    <exec_method name='start' type='method' exec='/usr/smf/monitor.crm'
...
...
...

root@s11-testbed:/var/svc/manifest/site# cd
root@s11-testbed:~#
```

You have corrected the spelling error from “`monitor.cr`” to “`monitor.crm`” in the method block.

9. Try bringing the `logapp` service up again.

```
root@s11-testbed:~# svcadm restart manifest-import
root@s11-testbed:~# svcs logapp
STATE          STIME      FMRI
maintenance    23:53:25  svc:/site/logapp:default
root@s11-testbed:~# svcadm clear logapp
root@s11-testbed:~# svcs logapp
STATE          STIME      FMRI
online         0:00:03   svc:/site/logapp:default
root@s11-testbed:~# exit
```

Summary: The `logapp` service is now up and running.

Task 2: Revert to a Previous SMF Snapshot

Often when a service is corrupted, it is really the current instance of that service that is non-operational. In that case, one of the options is to revert to a previous functional snapshot and correct the problem in that instance of the service.

1. Verify that the **s11-testbed** VM is running. If not, start it at this time.
2. Log in to the **s11-testbed** VM as the user **oracle**. Use the password **oracle1**.
3. Right-click the desktop and open a terminal window.
4. In the terminal window, run the **su -** command to assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation          SunOS 5.11 11.1          September 2012
root@s11-testbed:~#
```

5. Display the status of the **console-login** service.

```
root@s11-testbed:~# svcs console-login:default
STATE          STIME      FMRI
online         23:47:15   svc:/system/console-login:default

Currently, the service is online and running. Assume that it is in the maintenance state and you would like to revert to an earlier snapshot.
```

6. Use the **svccfg** utility to list the **console-login** service snapshots. Select the previous snapshot.

```
root@s11-testbed:~# svccfg
svc:> select system/console-login:default
svc:/system/console-login:default> listsnap
previous
running
start
svc:/system/console-login:default> revert previous
svc:/system/console-login:default> quit

In this step, you revert to the previous snapshot.
```

7. Use the **svcadm** commands to refresh and restart the service. Confirm that it is up and running.

```
root@s11-testbed:~# svcadm refresh system/console-login:default
root@s11-testbed:~# svcadm restart system/console-login:default
root@s11-testbed:~# svcs console-login:default
STATE          STIME      FMRI
online         00:08:34   svc:/system/console-login:default
```

Summary: The **refresh** option updates the SMF repository with the configuration information from the previous snapshot. After you refresh, you can start the service.

Task 3: Repair a Corrupt Repository

If you have corrupted service(s), SMF would not be able to bring them up and offer you the relevant functionality. In such situations where multiple services are corrupted, it is more reasonable to revert to an earlier functional repository.

1. Verify that the **s11-testbed** VM is running. If not, start it at this time.
2. Log in to the **s11-testbed** VM as the user **oracle**. Use the password **oracle1**.
3. Right-click the desktop and open a terminal window.
4. In the terminal window, run the **su -** command to assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation          SunOS 5.11 11.1          September 2012
root@s11-testbed:~#
```

5. Assuming that you are having trouble with many services, you would like to revert to the service repository version created on August 21, 2013.

Note: You may observe the service repository version with a different date.

```
root@s11-testbed:~# cd /lib/svc/bin
root@s11-testbed:/lib/svc/bin# ./restore_repository

See http://support.oracle.com/msg/SMF-8000-MY for more information on the use
of
this script to restore backup copies of the smf(5) repository.

If there are any problems which need human intervention, this script will
give instructions and then exit back to your shell.
./restore_repository[71]: [: /: arithmetic syntax error
The following backups of /etc/svc/repository.db exist, from
oldest to newest:

boot-20121210_032833
boot-20121210_040058
boot-20130814_021202
boot-20130821_001704
manifest_import-20130821_031039
manifest_import-20130821_231848
manifest_import-20130825_224547
manifest_import-20130825_225525

The backups are named based on their type and the time what they were taken.
Backups beginning with "boot" are made before the first change is made to
the repository after system boot. Backups beginning with "manifest_import"
are made after svc:/system/manifest-import:default finishes its processing.
The time of backup is given in YYYYMMDD_HHMMSS format.

Please enter either a specific backup repository from the above list to
restore it, or one of the following choices:
```

CHOICE	ACTION
-----	-----

```

boot          restore the most recent post-boot backup
manifest_import restore the most recent manifest_import backup
-seed-        restore the initial starting repository (All
               customizations will be lost, including those
               made by the install/upgrade process.)
-quit-        cancel script and quit

```

Enter response [boot]: **boot-20130821_001704**

Note: Your display may be different.

In this step, you are reverting to the service repository version created on August 21, 2013. Know that a new version is created by SMF after any service configuration.

6. The system responds as follows. If you would like to revert to the specified version, enter **yes**, otherwise **no**. Because this practice is meant to only provide a glimpse of some troubleshooting activities, you enter **no**.

```

...
...
After confirmation, the following steps will be taken:

svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
    -- renamed --> /etc/svc/repository.db_old_20130825_231805
/etc/svc/repository-boot-20130821_001704
    -- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M) .

```

Proceed [yes/no]? **no**

```

Exiting...
root@s11-testbed:/lib/svc/bin# cd
root@s11-testbed:~#

```

Summary: If you had entered **yes**, you would have successfully reverted to the **boot-20130821_001704** snapshot dated August 21, 2013.

Practices for Lesson 4: Administering ZFS

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

ZFS is the default file system in Oracle Solaris 11.1. ZFS presents a pooled storage model that completely eliminates the problems associated with partitioning, provisioning, bandwidth wastage, and stranded storage. Thousands of file systems can draw from a common storage pool, each consuming only as much space as is required.

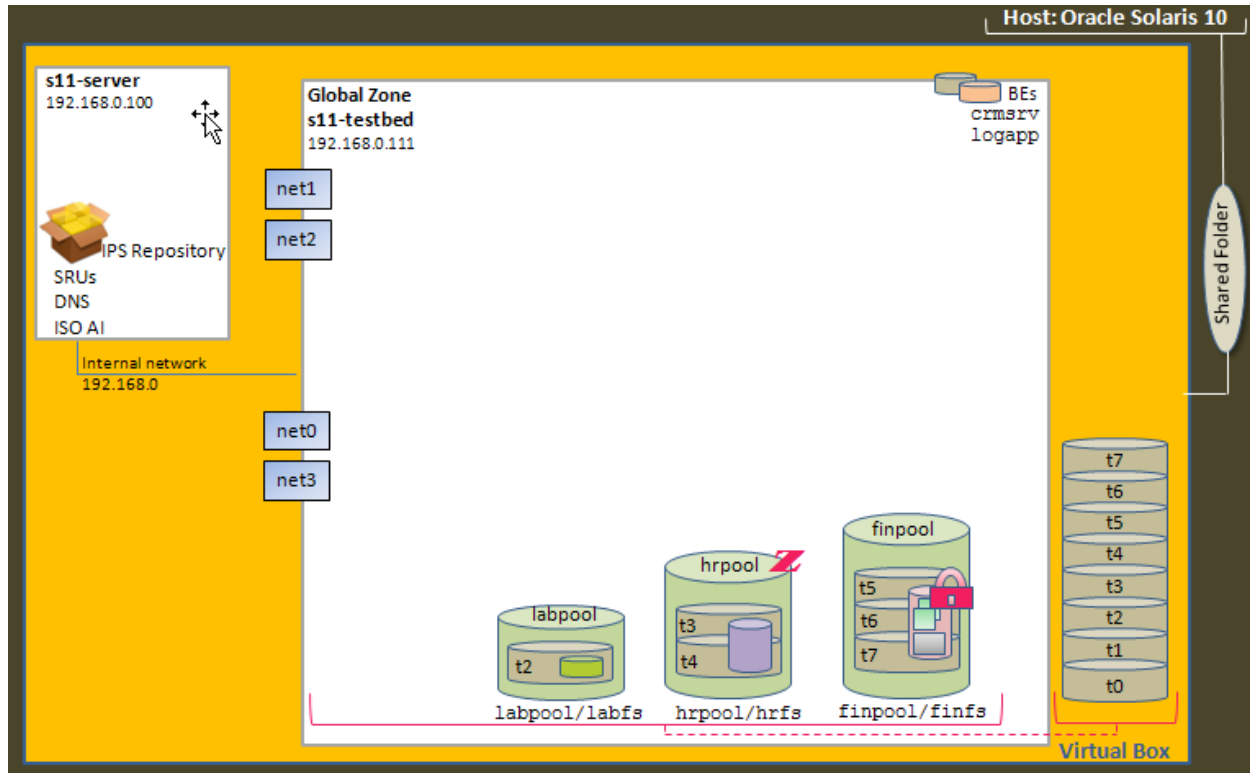
The s11-testbed environment is meant to validate and tune a host of Oracle ERP applications. To that effect, you need to address the following requirements:

- Store business application data.
- Ensure data redundancy to hedge against any sort of data loss.
- Set file system properties to ensure storage optimization and data security.
- Share data across systems.
- Provision backup and recovery options so that data can be recalled in the event of a disaster.

Notes

- Some command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.

Below is the schematic representation of the tasks you will accomplish in this practice.



In this practice, you will perform the following:

- Configure ZFS storage pools.
- Administer ZFS file systems.
- Configure ZFS properties.
- Configure ZFS shadow migration.
- Administer ZFS snapshots and clones.
- Perform ZFS data backup and recovery.

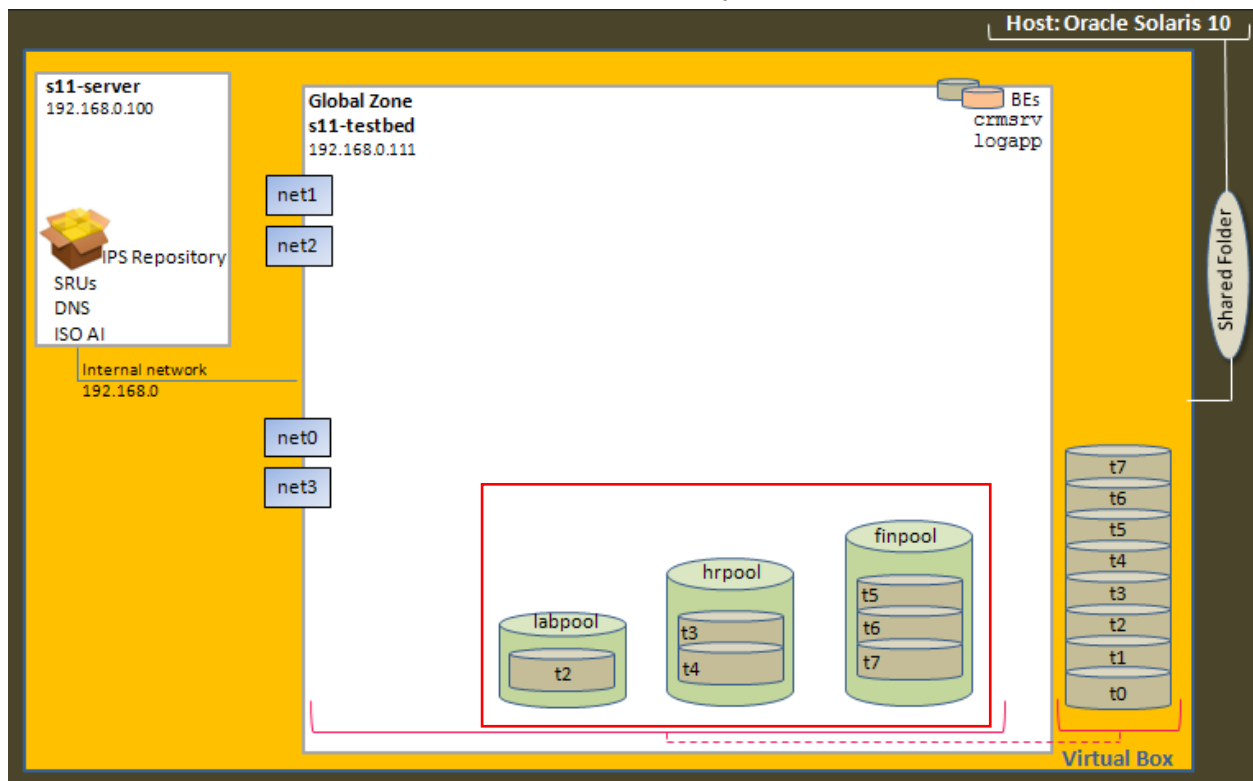
Practice 4-1: Configuring ZFS Storage Pools

Overview

To store business application data, you need to create the following storage pools.

- **labpool:** This is a non-redundant (RAID 0) storage pool meant for you and your team to experiment with various Oracle Solaris features and properties. Data loss is not a concern here.
- **hrpool:** Considering the importance of HR data, you will create a mirrored (RAID 1) storage pool.
- **finpool:** Given the criticality of financial transactions, a mirrored configuration would be the best. However, the volume of data involved would make it cost-prohibitive. A RAID-Z configuration would be the next best configuration.

Below is the schematic representation of the tasks that you will accomplish in this practice.



In this practice, you will perform the following tasks:

- Create ZFS pools.
- Display the status of the ZFS pools.

Tasks

1. Verify that the **s11-testbed** VM is running. If not, start the VM.
2. Log in to the **s11-testbed** VM as user `oracle`, and then run the `su -` command to assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password:
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```

3. Determine whether there are any existing ZFS pools.

```
root@s11-testbed:~# zpool list
NAME      SIZE  ALLOC   FREE      CAP  DEDUP  HEALTH  ALTROOT
rpool    39.5G  6.69G   32.8G    16%   1.00x  ONLINE   -
root@s11-testbed:~#
```

Note that `rpool` is the default ZFS pool created during OS installation. The purpose of this pool is to provide ZFS as the root file system.

4. Check the disks being used by `rpool`.

```
root@s11-testbed:~# zpool status
pool: rpool
state: ONLINE
scan: none requested
config:

    NAME            STATE        READ WRITE CKSUM
    rpool            ONLINE         0     0     0
        c7t0d0       ONLINE         0     0     0

errors: No known data errors
root@s11-testbed:~#
```

Notice that slice 0 of the disk `c7t0d0` is being used by `rpool`.

5. Identify the storage disks that are available on your system.

```
root@s11-testbed:~# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
  0. c7t0d0 <ATA-VBOX HARDDISK-1.0-40.00GB>
    /pci@0,0/pci8086,2829@d/disk@0,0
  1. c7t2d0 <ATA-VBOX HARDDISK-1.0-1.00GB>
    /pci@0,0/pci8086,2829@d/disk@2,0
  2. c7t3d0 <ATA-VBOX HARDDISK-1.0-1.00GB>
    /pci@0,0/pci8086,2829@d/disk@3,0
  3. c7t4d0 <ATA-VBOX HARDDISK-1.0-1.00GB>
    /pci@0,0/pci8086,2829@d/disk@4,0
```

4. c7t5d0 <ATA-VBOX HARDDISK-1.0-5.00GB>
/pci@0,0/pci8086,2829@d/disk@5,0
5. c7t6d0 <ATA-VBOX HARDDISK-1.0-5.00GB>
/pci@0,0/pci8086,2829@d/disk@6,0
6. c7t7d0 <ATA-VBOX HARDDISK-1.0-5.00GB>
/pci@0,0/pci8086,2829@d/disk@7,0
7. c7t8d0 <ATA-VBOX HARDDISK-1.0-20.00GB>
/pci@0,0/pci8086,2829@d/disk@8,0
8. c7t9d0 <ATA-VBOX HARDDISK-1.0-40.00GB>
/pci@0,0/pci8086,2829@d/disk@9,0

Specify disk (enter its number): ^C

root@s11-testbed:~#

Except c7t0d0 (which is being used by rpool), you can choose any other disk for creating your ZFS storage pools.

Note that disk addresses on your virtual machine may differ from the ones shown in this example.

6. Create a simple pool called labpool with one disk, c7t2d0.

```
root@s11-testbed:~# zpool create labpool c7t2d0
```

```
root@s11-testbed:~# zpool status labpool
```

```
pool: labpool
state: ONLINE
scan: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
labpool	ONLINE	0	0	0
c7t2d0	ONLINE	0	0	0

```
errors: No known data errors
```

```
root@s11-testbed:~#
```

7. Check the input/output activity of labpool.

```
root@s11-testbed:~# zpool iostat labpool
```

pool	capacity		operations		bandwidth	
	alloc	free	read	write	read	write
labpool	85K	1008M	1	12	7.75K	79.3K

```
root@s11-testbed:~#
```

Here, you see the total storage for the pool as well as read/write operation information.

8. Now, create a ZFS mirror pool called `hrpool` by using disks `c7t3d0` and `c7t4d0`. Note that you determined in step 7 that these disks are available.

```
root@s11-testbed:~# zpool create hrpool mirror c7t3d0 c7t4d0
root@s11-testbed:~# zpool status hrpool

  pool: hrpool
  state: ONLINE
  scan: none requested
config:

    NAME            STATE        READ  WRITE CKSUM
    hrpool           ONLINE         0     0     0
      mirror-0       ONLINE         0     0     0
        c7t3d0       ONLINE         0     0     0
        c7t4d0       ONLINE         0     0     0

errors: No known data errors
root@s11-testbed:~#
```

Note: The purpose of creating a mirror pool is to provide data redundancy.

9. Create a RAID-Z pool called `finpool` by using the next three available disks. Then view the status of the new pool.

```
root@s11-testbed:~# zpool create finpool raidz c7t5d0 c7t6d0 c7t7d0
root@s11-testbed:~# zpool status finpool

  pool: finpool
  state: ONLINE
  scan: none requested
config:

    NAME            STATE        READ  WRITE CKSUM
    finpool          ONLINE         0     0     0
      raidz1-0       ONLINE         0     0     0
        c7t5d0       ONLINE         0     0     0
        c7t6d0       ONLINE         0     0     0
        c7t7d0       ONLINE         0     0     0

errors: No known data errors
root@s11-testbed:~#
```

Here, you created the `raidz` pool, as indicated by the `status` command. The display shows `raidz1-0` as the name of the virtual device in the pool. The redundancy is `raidz1`, meaning that `raidz` level 1 maintains single parity. The digit 0 represents the first virtual device.

Note that `raidz` can provide redundancy at a lower cost compared to a mirrored pool. However, writing to a `raidz` pool is slower because of calculating and writing parity data compared to mirroring.

Summary: You have so far created `labpool`, `hrpool`, and `finpool` based on the requirement stated earlier.

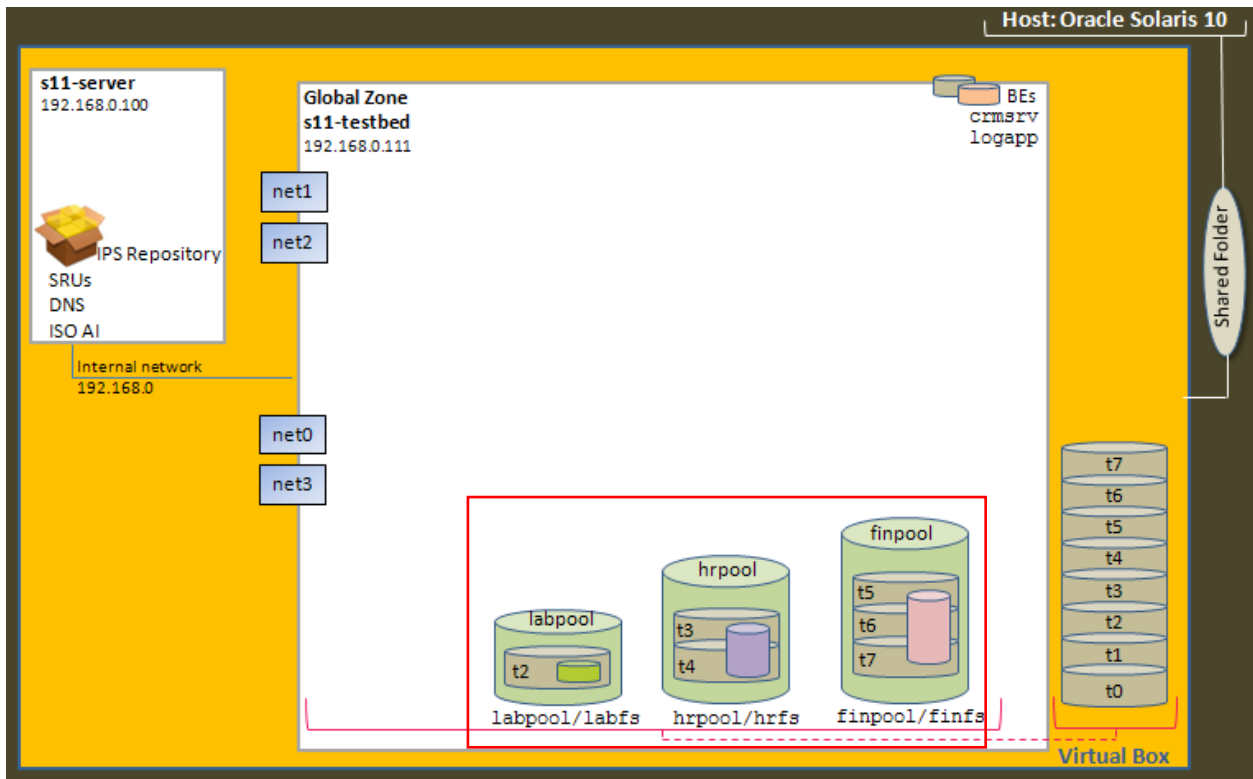
Practice 4-2: Administering ZFS File Systems

Overview

Having created the storage pools, you need to create the following file systems:

- labfs in labpool
- hrfs in hrpool
- finfs in finpool

Below is the schematic representation of the tasks that you will accomplish in this practice.



In this practice, you will perform the following tasks:

- Create ZFS file systems.
- Display the status of the file systems.

Tasks

1. Verify that the **s11-testbed** VM is running. If not, start the VM.
2. Log in to the **s11-testbed** VM as user `oracle`, and then run the `su -` command to assume primary administrator privileges. Use the password `oracle1`.

```
oracle@s11-testbed:~$ su -
Password:
Oracle Corporation          SunOS 5.11          11.1          September 2012
root@s11-testbed:~#
```

3. Create a ZFS file system called `labfs` within `labpool` and view the file systems in the pool.

```
root@s11-testbed:~# zfs create labpool/labfs
root@s11-testbed:~# zfs list -r labpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
labpool	124K	976M	32K	/labpool
labpool/labfs	31K	976M	31K	/labpool/labfs

```
root@s11-testbed:~#
```

4. Similarly, create a ZFS file system called `hrfs` within `hrpool` and then view the file systems in the pool.

```
root@s11-testbed:~# zfs create hrpool/hrfs
root@s11-testbed:~# zfs list -r hrpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
hrpool	124K	976M	32K	/hrpool
hrpool/hrfs	31K	976M	31K	/hrpool/hrfs

```
root@s11-testbed:~#
```

5. Create a ZFS file system called `finfs` within `finpool` and then view the file systems in the pool.

```
root@s11-testbed:~# zfs create finpool/finfs
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	165K	9.75G	36.0K	/finpool
finpool/finfs	34.6K	9.75G	34.6K	/finpool/finfs

```
root@s11-testbed:~#
```

Summary: You have successfully created the required file systems. They can be called datasets too. You will see how these file systems or datasets are deployed in Oracle Solaris Zones later in the course.

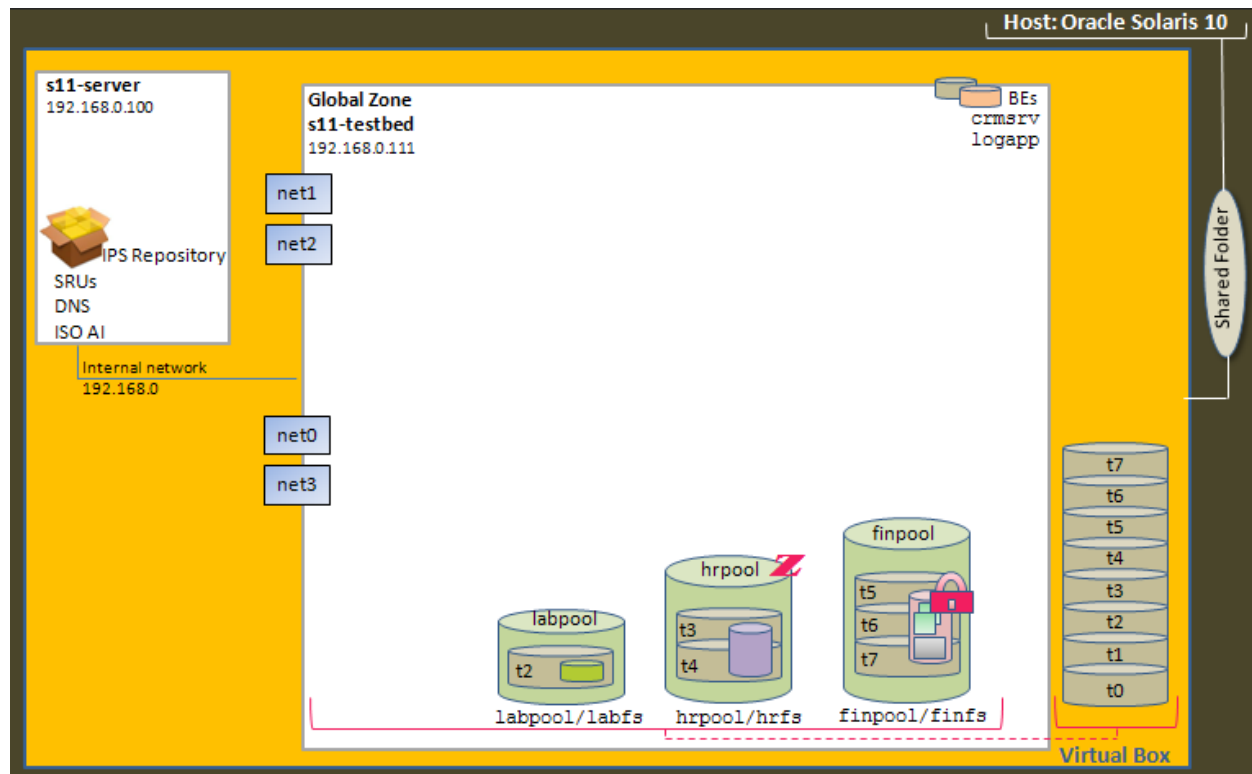
Practice 4-3: Configuring ZFS Properties

Overview

The storage pool and the required file systems have been created. You now need to set the following file system properties to ensure that data is stored optimally, data is secure, and that data can be shared across systems:

- Quota
- Reservation
- Data compression
- Data encryption
- Data sharing

Below is the schematic representation of the tasks that you will accomplish in this practice.



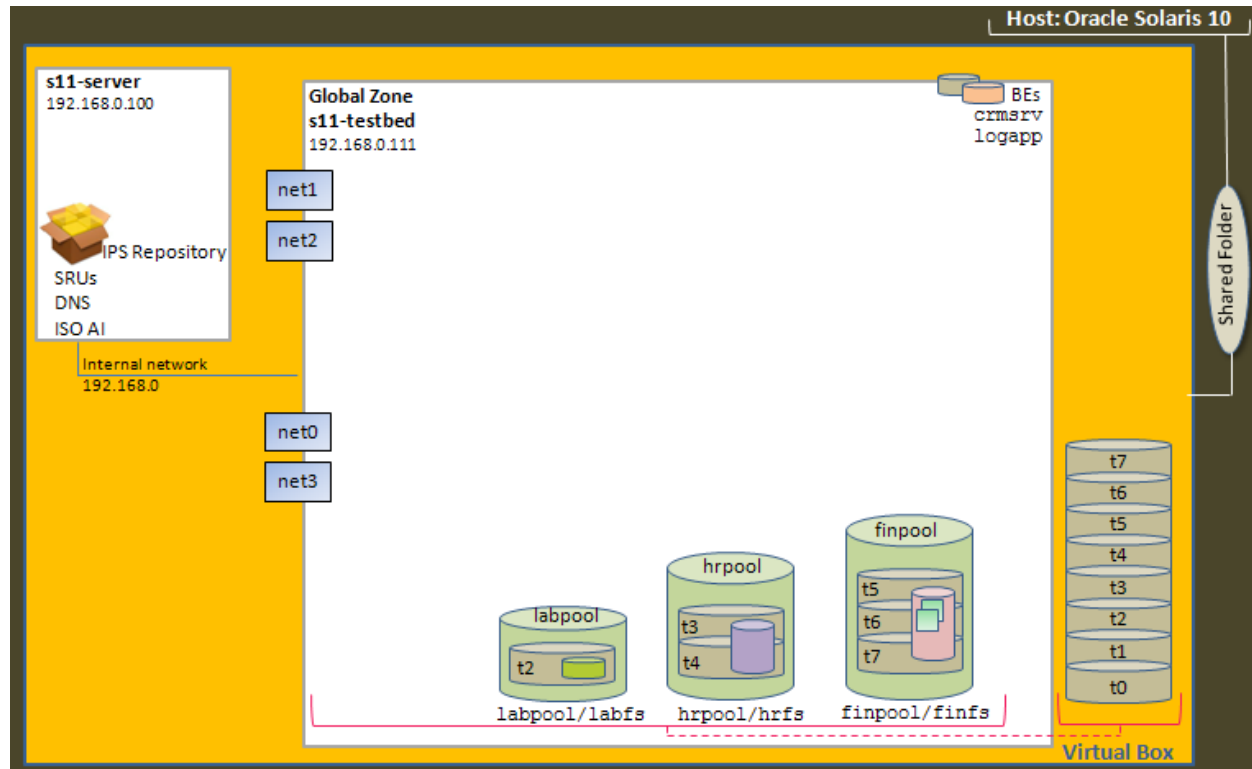
In this practice, you will perform the following activities:

- Configure the `quota` property.
- Configure the `reservation` property.
- Configure ZFS compression.
- Configure a ZFS encrypted file system.
- Configure the `share` property.

Task 1: Configure the quota Property

Considering that the Oracle Financials application will operate under various categories, you need to create separate datasets for the Accounts Receivable (`ar`), Invoicing (`inv`), and Delivery (`del`) segments. Apart from that, set a 5 GB quota of disk storage space for the `ar` and `inv` datasets.

As a rule, all datasets in a pool share the same disk space. The `quota` property is used to limit the amount of space consumed by a dataset and all its children.



1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume primary administrator privileges.
3. Based on requirement, create the `ar`, `inv`, and `del` datasets within the `finfs` file system.

```
root@s11-testbed:~# zfs create finpool/finfs/ar
root@s11-testbed:~# zfs create finpool/finfs/del
root@s11-testbed:~# zfs create finpool/finfs/inv
root@s11-testbed:~# zfs list -r finpool
NAME                                USED  AVAIL  REFER  MOUNTPOINT
finpool                             309K  9.75G  36.0K  /finpool
finpool/finfs                       143K  9.75G  38.6K  /finpool/finfs
finpool/finfs/ar                    34.6K  9.75G  34.6K  /finpool/finfs/ar
finpool/finfs/del                   34.6K  9.75G  34.6K  /finpool/finfs/del
finpool/finfs/inv                   34.6K  9.75G  34.6K  /finpool/finfs/inv
root@s11-testbed:~#
```

4. Get the quota property for `finpool` by using the `zfs get` command.

```
root@s11-testbed:~# zfs get quota finpool
```

NAME	PROPERTY	VALUE	SOURCE
finpool	quota	none	default

Note that no quota is assigned for `finpool` at this point. This implies that all file systems or datasets created under `finpool` will have equal access to the total disk space available. In practice, this can become a contentious issue.

5. View the file systems of `finpool`.

```
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	309K	9.75G	36.0K	/finpool
finpool/finfs	143K	9.75G	38.6K	/finpool/finfs
finpool/finfs/ar	34.6K	9.75G	34.6K	/finpool/finfs/ar
finpool/finfs/del	34.6K	9.75G	34.6K	/finpool/finfs/del
finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv

```
root@s11-testbed:~#
```

The total disk space for `finpool` is 9.75 GB. Because a quota is not assigned to `finpool` (as you saw earlier), all the datasets under `finpool` have equal access to the 9.75 GB.

6. Set a quota of 5 GB on the `ar` and `del` datasets and display the results.

```
root@s11-testbed:~# zfs set quota=5G finpool/finfs/ar
root@s11-testbed:~# zfs set quota=5G finpool/finfs/del
root@s11-testbed:~# zfs get -r quota finpool
```

NAME	PROPERTY	VALUE	SOURCE
finpool	quota	none	default
finpool/finfs	quota	none	default
finpool/finfs/ar	quota	5G	local
finpool/finfs/del	quota	5G	local
finpool/finfs/inv	quota	none	default

```
root@s11-testbed:~#
```

A quota of 5 GB is set to the `ar` and `del` datasets.

7. View the file systems of `finpool` again.

```
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	458K	9.75G	36.0K	/finpool
finpool/finfs	143K	9.75G	38.6K	/finpool/finfs
finpool/finfs/ar	34.6K	5.00G	34.6K	/finpool/finfs/ar
finpool/finfs/del	34.6K	5.00G	34.6K	/finpool/finfs/del
finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv

```
root@s11-testbed:~#
```

Summary: Note that the 5 GB set as a quota is not guaranteed to `ar` and `del`. If other datasets in the pool were to use up all the disk space, `ar` and `del` would be left with nothing. That is where the `reservation` property is used to ensure that the space assigned is only used by that specific dataset.

Additional Note: The `ar` dataset at this point has so far used up 34.6 KB. Trying to set a quota on `finpool/finfs/ar` to 10 KB throws an error message. This is because you cannot set a quota to an amount less than what is currently being used.

```
root@s11-testbed:~# zfs list finpool/finfs/ar
```

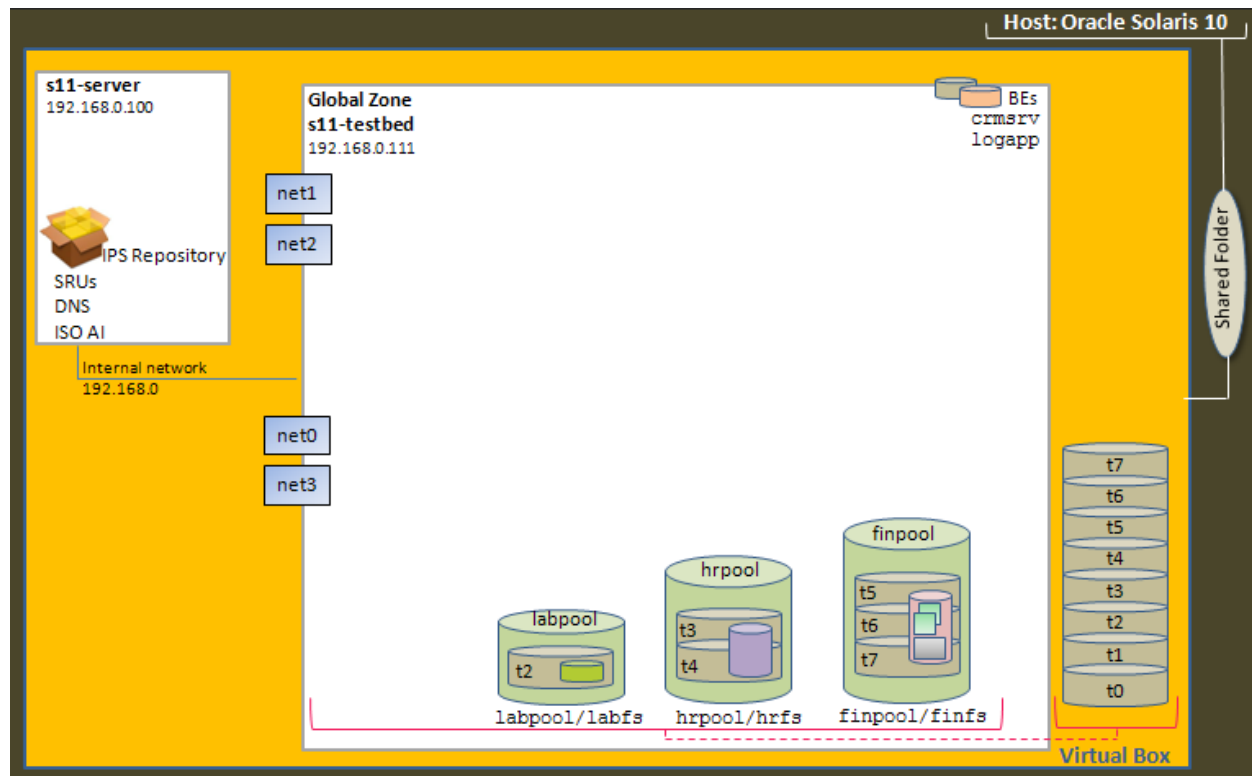
NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool/finfs/ar	34.6K	5.00G	34.6K	/finpool/finfs/ar

```
root@s11-testbed:~# zfs set quota=10K finpool/finfs/ar
```

```
cannot set quota for 'finpool/finfs/ar': size is less than current
used or reserved space
```

Task 2: Configure the `reservation` Property

You now need to reserve 1 GB for the `inv` dataset. Unlike `quota`, the `reservation` property guarantees the allocated disk space for a dataset by removing that amount from the free space that the other datasets have equal access to.



1. Display the reservation property of finpool by using the `zfs get` command.

```
root@s11-testbed:~# zfs get -r reservation finpool
```

NAME	PROPERTY	VALUE	SOURCE
finpool	reservation	none	default
finpool/finfs	reservation	none	default
finpool/finfs/ar	reservation	none	default
finpool/finfs/del	reservation	none	default
finpool/finfs/inv	reservation	none	default

```
root@s11-testbed:~#
```

The reservation property is not set on any datasets under finpool.

2. View the file systems of finpool again.

```
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	458K	9.75G	36.0K	/finpool
finpool/finfs	143K	9.75G	38.6K	/finpool/finfs
finpool/finfs/ar	34.6K	5.00G	34.6K	/finpool/finfs/ar
finpool/finfs/del	34.6K	5.00G	34.6K	/finpool/finfs/del
finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv

```
root@s11-testbed:~#
```

Observe that `inv` continues to have equal access to the total disk space of 9.75 GB.

3. Reserve 1 GB for the `inv` dataset by using the `zfs set` command.

```
root@s11-testbed:~# zfs set reservation=1G finpool/finfs/inv
```

```
root@s11-testbed:~# zfs get reservation finpool/finfs/inv
```

NAME	PROPERTY	VALUE	SOURCE
finpool/finfs/inv	reservation	1G	local

4. Display the reservation property of finpool by using the `zfs get` command.

```
root@s11-testbed:~# zfs get -r reservation finpool
```

NAME	PROPERTY	VALUE	SOURCE
finpool	reservation	none	default
finpool/finfs	reservation	none	default
finpool/finfs/ar	reservation	none	default
finpool/finfs/del	reservation	none	default
finpool/finfs/inv	reservation	1G	local

```
root@s11-testbed:~#
```

Observe that the `inv` dataset has been guaranteed 1 GB disk space.

5. View the file systems of `finpool` again.

```

root@s11-testbed:~# zfs list -r finpool
NAME                                USED  AVAIL  REFER  MOUNTPOINT
finpool                             1.00G  8.75G  36.0K  /finpool
finpool/finfs                       1.00G  8.75G  38.6K  /finpool/finfs
finpool/finfs/ar                    34.6K  5.00G  34.6K  /finpool/finfs/ar
finpool/finfs/del                   34.6K  5.00G  34.6K  /finpool/finfs/del
finpool/finfs/inv                   34.6K  9.75G  34.6K  /finpool/finfs/inv
root@s11-testbed:~#

```

Summary: As expected, the space available to `ar` and `del` is limited to 5 GB. However, because `inv` was guaranteed 1 GB disk space from the total of 9.75 GB, the `AVAIL` field for `inv` now shows 9.75 GB, while the total disk space of the pool has reduced to 8.75 GB. This is because the reserved disk space of 1 GB is not available to `finpool` anymore.

Additional Note: Reservations could mislead you to over-estimating the space used in your pool. The `df` command always displays the actual usage.

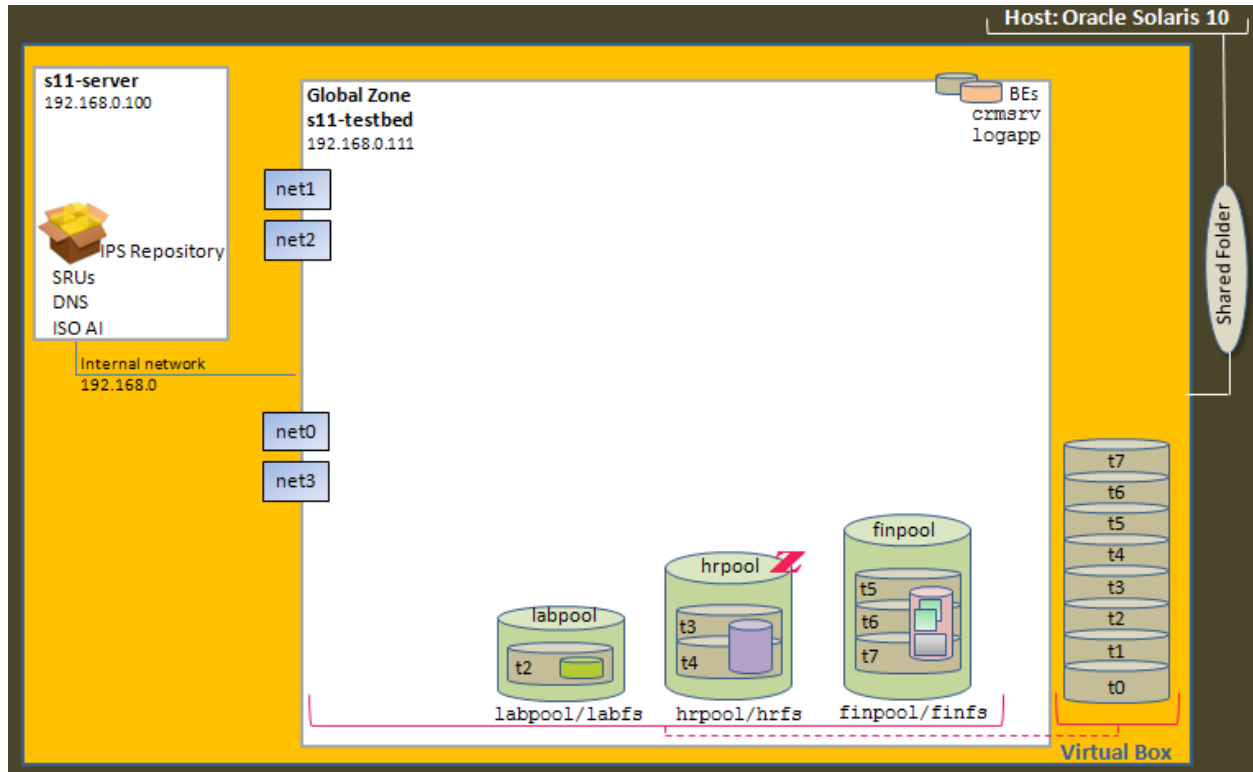
```

root@s11-testbed:~# df -h /finpool/finfs/ar
Filesystem      Size  Used  Available Capacity  Mounted on
finpool/finfs/ar  5.0G   35K    5.0G      1%    finpool/finfs/ar
root@s11-testbed:~# df -h /finpool/finfs/del
Filesystem      Size  Used  Available Capacity  Mounted on
finpool/finfs/del  5.0G   35K    5.0G      1%    inpool/finfs/del
root@s11-testbed:~# df -h /finpool/finfs/inv
Filesystem      Size  Used  Available Capacity  Mounted on
finpool/finfs/inv  9.8G   34K    9.8G      1%    finpool/finfs/inv
root@s11-testbed:~#

```


Task 3: Configure ZFS Compression

Considering the volume of data transaction that is likely to happen in `hrpool`, you would like to apply some compression property to the pool so that data storage can be optimized. For now, you need to try a few options in compression to see how compression can be optimized for your future needs.



1. Verify that the **s11-testbed** VM is running.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume primary administrator privileges.
3. Run the `zfs list` command to list the space currently used by `hrpool`. Make a note of the value indicated.

```
root@s11-testbed:~# zfs list hrpool
NAME          USED  AVAIL  REFER  MOUNTPOINT
hrpool        124K   976M   32K    /hrpool
```

`hrpool` currently has 976 MB available space.

4. List the size of the archive file in the `/opt/ora/lab/custarchive.tar` directory by using the `ls` command with the `-lh` options. Make a note of it.

```
root@s11-testbed:~# ls -lh /opt/ora/lab/custarchive.tar
-rw-r--r--  1 root    root      787K Nov  4 09:09
/opt/ora/data/custarchive.tar
```

The archive file takes approximately 787 KB.

5. You need to copy the archive file to a directory under `hrpool`. Create a directory named `cmp`.

```
root@s11-testbed:~# mkdir /hrpool/cmp
```

This directory will be used to store the compressed archive data.

6. Display the current settings of the `compression` and `compressratio` properties for `hrpool` by using the `zfs get` command.

```
root@s11-testbed:~# zfs get compression,compressratio hrpool
```

NAME	PROPERTY	VALUE	SOURCE
hrpool	compression	off	default
hrpool	compressratio	1.00x	-

The `compression` property is set to `off` by default and, because compression is off, the `compressratio` property is set to `1.00x`. A ratio of 1:1 for data means no compression.

7. Copy `/opt/ora/lab/custarchive.tar` to `/hrpool/cmp/custarchive.tar`. List the file to display its size.

```
root@s11-testbed:~# cp /opt/ora/lab/custarchive.tar \
/hrpool/cmp/custarchive.tar
```

```
root@s11-testbed:~# ls -lh /hrpool/cmp
```

```
total 1
-rw-r--r--  1 root    root    786K Nov 4 09:47 custarchive.tar
```

After copying the file into the pool, it consumes approximately the same space.

8. List the space used by `hrpool` by using the `zfs list` command.

```
root@s11-testbed:~# zfs list hrpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
hrpool	1024K	975M	931K	/hrpool

```
root@s11-testbed:~#
```

Observe that the used space in `hrpool` has now grown to 1024 KB.

9. Set the `compression` property for `hrpool` to `gzip` and verify that the new value is set.

```
root@s11-testbed:~# zfs set compression=gzip hrpool
```

```
root@s11-testbed:~# zfs get compression hrpool
```

NAME	PROPERTY	VALUE	SOURCE
hrpool	compression	gzip	local

You set the `compression` property on the `hrpool` file system to `gzip`. Notice the space usage of the files, which is stored in the `hrpool` file system.

Know that instead of `gzip`, you can set any of the following values for compression:

```
on | off | lzjb | gzip-[1-9] | zle
```

10. Copy `/opt/ora/lab/custarchive.tar` to `/hrpool/cmp/archive2.tar`. List all the files in `/hrpool/cmp` to display their sizes. Are the files in `/hrpool/cmp` the same size?

```
root@s11-testbed:~# cp /opt/ora/lab/custarchive.tar \
/hrpool/cmp/archive2.tar
root@s11-testbed:~# ls -lh /hrpool/cmp
total 3529
-rw-r--r--  1 root      root           786K Nov  4 09:11 archive2.tar
-rw-r--r--  1 root      root           786K Nov  4 09:09 custarchive.tar
```

Yes, they are equal as displayed by the `ls` command.

11. List the space used by `hrpool` by using the `zfs list` command.

```
root@s11-testbed:~# zfs list hrpool
NAME          USED  AVAIL  REFER  MOUNTPOINT
hrpool        1.15M  975M   1.06M   /hrpool
```

Note that the sum of the two tar files should be 1572 KB. However, the `zfs list` command displays a lesser amount at 1177 KB (1.15 MB) because compression is in effect.

12. Display the current setting of the `compressratio` property for `hrpool` by using the `zfs get` command.

```
root@s11-testbed:~# zfs get compressratio hrpool
NAME          PROPERTY          VALUE  SOURCE
hrpool        compressratio     1.67x  -
```

The ratio is 1.67x, which means that data is being compressed at a ratio of 1.67-1 (approximately 59%).

13. Copy `/opt/ora/lab/custarchive.tar` to `/hrpool/cmp/archive3.tar` and list all the files in `/hrpool/cmp`.

```
root@s11-testbed:~# cp /opt/ora/lab/custarchive.tar \
/hrpool/cmp/archive3.tar
root@s11-testbed:~# ls -lh /hrpool/cmp
total 2405
-rw-r--r--  1 root      root           786K Nov  4 09:11 archive2.tar
-rw-r--r--  1 root      root           786K Nov  4 09:12 archive3.tar
-rw-r--r--  1 root      root           786K Nov  4 09:09 custarchive.tar
```

Note that all files report the same size.

14. Display the space used by the files in `/hrpool/cmp` by using the `du -h` command.

```
root@s11-testbed:~# du -h /hrpool/cmp/*
898K  /hrpool/cmp/custarchive.tar
153K  /hrpool/cmp/archive2.tar
153K  /hrpool/cmp/archive3.tar
```

The `custarchive.tar` file uses about the same space as the `ls -lh` command indicates. The `custarchive.tar` file was created in the `cmp` file system before enabling compression. However, the other two files show a percentage of the original size of the files precisely because they were created after implementing compression.

15. Display the current value of the `compressratio` property for `hrpool` by using the `zfs get` command.

```
root@s11-testbed:~# zfs get compressratio hrpool
```

NAME	PROPERTY	VALUE	SOURCE
hrpool	compressratio	2.18x	-

The compression ratio is now 2.18x. It has increased with the addition of the second compressed file. A larger portion of the data in the pool is now being compressed.

16. Remove the `/hrpool/cmp/custarchive.tar` file.

```
root@s11-testbed:~# rm /hrpool/cmp/custarchive.tar
```

17. Display the current value of the `compressratio` property for `hrpool` by using the `zfs get` command.

```
root@s11-testbed:~# zfs get compressratio hrpool
```

NAME	PROPERTY	VALUE	SOURCE
hrpool	compressratio	5.18x	-

The compression ratio has increased again with the removal of the uncompressed file.

18. List the space used by `hrpool` by using the `zfs list` command and list the space used by the remaining two files in `/hrpool/cmp` by using the `du -h` command.

```
root@s11-testbed:~# zfs list hrpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
hrpool	432K	976M	339K	/ hrpool

```
root@s11-testbed:~# du -h /hrpool/cmp/*
```

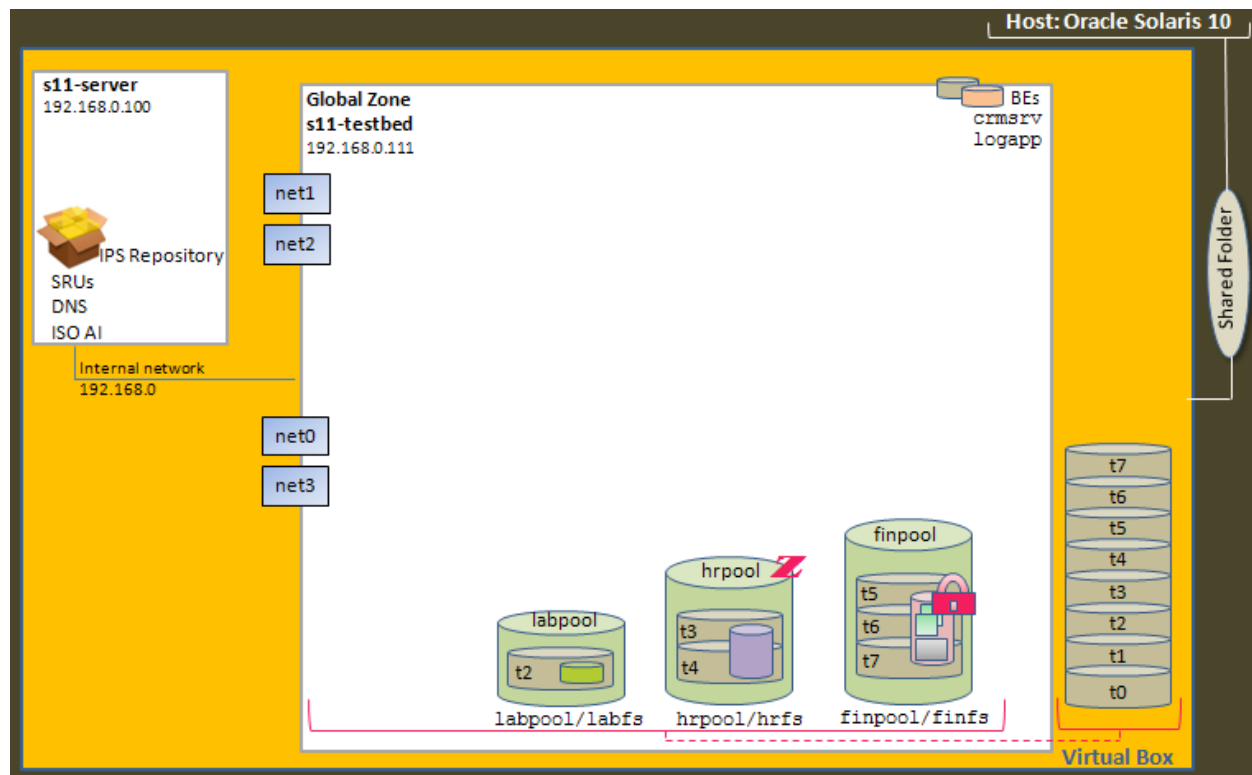
153K	/hrpool/cmp/archive2.tar
153K	/hrpool/cmp/archive3.tar

The `REFER` value and the sum of the space used by the two files in `/hrpool/cmp` are correlated.

Summary: With more data files in a ZFS file system with compression enabled, there is more compression.

Task 4: Configure ZFS Encryption

Considering the nature of transactions in the Finance department, it would be useful to encrypt the `finfs` dataset created under the non-encrypted `finpool` pool with a raw key. Encryption is the process where data is encoded for privacy and a key is needed by the data owner to access the encoded data.



Perform these steps to configure a ZFS encrypted file system:

1. Generate a 256-bit AES raw key in a keystore file named `/myzfskey`.

```
root@s11-testbed:~# pktool genkey keystore=file
outkey=/myzfskey keytype=aes keylen=256
```

2. Encrypt the ZFS file system named `finpool/finfs1` by using the `aes-256-ccm` algorithm and the key that you generated in the previous step.

```
root@s11-testbed:~# zfs create -o encryption=aes-256-ccm
-o keysource=raw,file:///myzfskey finpool/finfs1
```

3. Display the encryption property of the `finpool/finfs1` file system.

```
root@s11-testbed:~# zfs get encryption finpool/finfs1
```

NAME	PROPERTY	VALUE	SOURCE
finpool/finfs1	encryption	aes-256-ccm	local

4. Display the `keysource` property of the `finpool/finfs1` file system.

```
root@s11-testbed:~# zfs get keysource finpool/finfs1
```

NAME	PROPERTY	VALUE	SOURCE
finpool/finfs1	keysource	raw,file:///myzfskey	local

After performing the above step, remove the `finpool/finfs1` file system.

```
root@s11-testbed:~# zfs destroy finpool/finfs1
```

Task 5: Configure the share Property

As you continue to build the testbed environment, you and the many users who will use the platform might have to access files residing on the **s11-server1** system. In this case, some archived data resides on gail's home directory in the **s11-server1** system that needs to be accessed from the **s11-testbed** system.

1. Verify that the **s11-server1** VM is running. If it is not, start the VM. Also start the **s11-testbed** VM.
2. Log in to the **s11-server1** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.
3. Display the file systems that are available by using `zfs list`.

```
root@s11-server1:~# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               10.4G  28.5G   4.58M   /rpool
rpool/ROOT                          2.25G  28.5G    31K   legacy
rpool/ROOT/solaris                  2.25G  28.5G   2.06G   /
rpool/ROOT/solaris/var              112M   28.5G   92.3M   /var
rpool/VARSHARE                      48K    28.5G   48K    /var/share
rpool/dump                          1.03G  28.5G   1.00G   -
rpool/export                       6.07G  28.5G    35K   /export
rpool/export/IPS                   6.07G  28.5G   6.07G   /export/IPS
rpool/export/ZFS_data               31K    28.5G    31K   /export/ZFS_data
rpool/export/home                   104K    28.5G    33K   /export/home
rpool/export/home/gail              35K    28.5G    35K   /export/home/gail
rpool/export/home/oracle            36K    28.5G    36K   /export/home/oracle
rpool/swap                          1.03G  28.5G   1.00G   -
root@s11-server1:~#
```

4. Create a file in gail's directory.

```
root@s11-server1:~# cd /export/home/gail
root@s11-server1:/export/home/gail# touch crmreq
```

In gail's home directory, you created the `crmreq` file.

5. Change the permissions on gail's home directory by using the `chmod` command.

```
root@s11-server1:/export/home/gail# chmod 777 /export/home/gail
root@s11-server1:/export/home/gail# ls -ld /export/home/gail
drwxrwxrwx  2 gail  staff          4 Nov 4 08:27 /export/home/gail
```

6. Share gail's home directory with other users on the network.

```
root@s11-server1:/export/home/gail# zfs set share=name=gail,
path=/export/home/gail,prot=nfs rpool/export/home/gail
name=gail,path=/export/home/gail,prot=nfs
```

Enable the share property on `/export/home/gail`.

```
root@s11-server1:/export/home/gail# zfs set sharenfs=on \
rpool/export/home/gail
```

```
root@s11-server1:/export/home/gail# share
gail      /export/home/gail      nfs      sec=sys,rw
```

This confirms that the file system is being shared.

Check whether the NFS server is online.

```
root@s11-server1:/export/home/gail# svcs -a | grep nfs
disabled      Oct_28      svc:/network/nfs/cbd:default
disabled      Oct_28      svc:/network/nfs/client:default
online        Oct_28      svc:/network/nfs/fedfs-client:default
online        21:24:21   svc:/network/nfs/status:default
online        21:24:22   svc:/network/nfs/mapid:default
online        21:24:22   svc:/network/nfs/nlockmgr:default
online        21:24:24   svc:/network/nfs/rquota:default
online        21:24:24   svc:/network/nfs/server:default
root@s11-server1:/export/home/gail#
```

Note: You may need to manually share the NFS file system if it fails to do so automatically. If the NFS server is not enabled, issue this command:

```
# share -F nfs -o rw /export/home/gail
```

- Log in to the s11-testbed VM as the oracle user. Use oracle1 as the password. Open a terminal window and assume administrator privileges. Check whether you can see the share.

```
root@s11-testbed:~# dfshares s11-server1
RESOURCE                                SERVER  ACCESS  TRANSPORT
s11-server1:/export/home/gail          s11-server1  -      -
. . .
```

Yes, you can see the resource shared by the s11-server1 server.

- Create the mount point and mount the shared resource.

```
root@s11-testbed:~# mkdir /gaildir
root@s11-testbed:~# mount -F nfs s11-server1:/export/home/gail /gaildir
root@s11-testbed:~# cd /gaildir
root@s11-testbed:/gaildir# ls
crmreq      local.cshrc local.login local.profile
You can see the shared file crmreq in gail's home directory.

root@s11-testbed:/gaildir# touch crmdata
root@s11-testbed:/gaildir# ls
crmdata      crmreq      local.cshrc local.login local.profile
```

Summary: You can now create another file in the shared directory because you have read/write access to it.

Note: Do not perform the steps given in the following Additional Note section. The note is only for your reference.

Additional Note: When you are finished working with gail's directory, you can unmount it if you wish to.

```
root@s11-testbed:/gaildir# cd
root@s11-testbed:~# umount /gaildir
```

If you are unable to unmount the directory, use the `-f` option.

```
root@s11-testbed:~# umount -f /gaildir
```

Return to the **s11-server1** VM and stop sharing the directory.

```
root@s11-server1:~# zfs set sharenfs=off rpool/export/home/gail
```


Practice 4-4: Configuring ZFS Shadow Migration

Overview

Oracle Solaris 11.1 features ZFS shadow migration. Using the shadow migration feature, you can migrate data from one file system to another file system while simultaneously allowing access and modification of the shared file system. In this case, you would like to migrate the `rpool/export/ZFS_data` file system residing on the **s11-server1** system to the **s11-testbed** system.

In this practice, you will perform the following tasks:

- Identify the source file system.
- Migrate the file system.

Task 1: Identify the Source File System

Perform these steps to prepare the source file systems:

1. Verify that the **s11-server1** and **s11-testbed** VMs are running.
2. Log in to the **s11-server1** VM as the user `oracle` and `su` to the `root` role.
3. Verify that the `rpool/export/ZFS_data` file system exists.

```
root@s11-server1:~# zfs list rpool/export/ZFS_data
NAME                                USED    AVAIL    REFER  MOUNTPOINT
rpool/export/ZFS_data              31K    28.5G    31K    /export/ZFS_data
root@s11-server1:~#
```

4. Share the ZFS file systems as read-only and display the results.

```
root@s11-server1:~# share -F nfs -o ro /export/ZFS_data
root@s11-server1:~# showmount -e
export list for s11-server1:
/export/ZFS_data (everyone)
```

5. Store data in the ZFS file system.

```
root@s11-server1:~# cp /opt/ora/iso/sol-11_1-ai-x86.iso \
/export/ZFS_data
```

Task 2: Migrate the File System

Perform the following steps on the **s11-testbed** VM to migrate the required file system as the `root` role.

1. Verify that this server can access DNS services.

```
root@s11-testbed:~# nslookup s11-server1
Server:      192.168.0.100
Address:     192.168.0.100#53

Name:       s11-server1.mydomain.com
Address:    192.168.0.100
```

2. Search for the shadow-migration package in the IPS repository.

```
root@s11-testbed:~# pkg search shadow-migration
INDEX      ACTION VALUE                                PACKAGE
pkg.fmri   set      solaris/system/file-system/shadow-migration
pkg:/system/file-system/shadow-migration@0.5.11-0.175.1.0.0.24.2
```

3. Display detailed information about the shadow-migration package.

```
root@s11-testbed:~# pkg info -r shadow-migration
      Name: system/file-system/shadow-migration
      Summary: Shadow migration libraries and commands
      Description: Shadow migration can be used to migrate data
                   from an existing file system to a new file
                   system. It can be enabled by setting the shadow
                   property on the destination ZFS dataset using
                   the zfs(1M) command.
      Category: System/File System
      State: Not installed
      Publisher: solaris
      Version: 0.5.11
      Build Release: 5.11
      Branch: 0.175.1.0.0.24.2
      Packaging Date: Wed Sep 19 18:48:33 2012
      Size: 497.99 kB
      FMRI: pkg://solaris/system/file-system/shadow-
migration@0.5.11,5.11-0.175.1.0.0.24.2:20120919T184833Z
```

4. Install the shadow-migration package and display the results.

```
root@s11-testbed:~# pkg install shadow-migration
Creating Plan ...
      Packages to install:      1
      Create boot environment:  No
      Create backup boot environment:  No
      Services to change:      1
DOWNLOAD      PKGS      FILES      XFER (MB)      SPEED
Completed      1/1      14/14      0.2/0.2      1.3M/s

PHASE      ITEMS
Installing new action      39/39
Updating package state database      Done
Updating image state      Done
Creating fast lookup database      Done
root@s11-testbed:~# pkg list shadow-migration
NAME (PUBLISHER)      VERSION      IFO
system/file-system/shadow-migration 0.5.11-0.175.1.0.0.24.2 i--
```

5. Enable the shadow migration service and display the results.

```
root@s11-testbed:~# svcadm enable shadowd
root@s11-testbed:~# svcs shadowd
STATE          STIME          FMRI
online         17:22:40      svc:/system/filesystem/shadowd:default
```

6. Create a ZFS file system for the shared ZFS file system.

```
root@s11-testbed:~# zfs create -o shadow=nfs://s11-server1/export/ZFS_data
rpool/export/shadow_ZFS_data
```

Note: Perform step 7 immediately after step 6 to observe the output mentioned in the step 7.

7. Display statistics on in-progress shadow migration until the process is complete.

```
root@s11-testbed:~# shadowstat
```

	EST			
	BYTES	BYTES		ELAPSED
DATASET	XFRD	LEFT	ERRORS	TIME
rpool/export/shadow_ZFS_data	-	-	-	00:00:18
rpool/export/shadow_ZFS_data	-	-	-	00:00:28
rpool/export/shadow_ZFS_data	-	-	-	00:00:38
rpool/export/shadow_ZFS_data	-	-	-	00:00:48
rpool/export/shadow_ZFS_data	-	-	-	00:00:58
rpool/export/shadow_ZFS_data	-	-	-	00:01:09
rpool/export/shadow_ZFS_data	-	-	-	00:01:19
rpool/export/shadow_ZFS_data	-	-	-	00:01:29

No migrations in progress

```
root@s11-testbed:~#
```

8. List the contents of the shadow migration directories after the migration is complete.

```
root@s11-testbed:~# ls -l /export/shadow_ZFS_data
total 625617
-rwxr--r--  1 root root 325054464   Nov 4 17:15 sol-11_1-ai-x86.iso
```

Summary: You can see that the `iso` file is now available in the **s11-testbed** system.

Practice 4-5: Administering ZFS Snapshots and Clones

Overview

ZFS provides snapshot and cloning capabilities. For backing up data, you can create snapshots and save it on the local or remote system. In the event of a disaster where you need to recall data, you can `rollback` and recover lost data.

Considering the criticality of financial transactions, it is a good practice to take snapshots at the end of each day. For now, you will create a snapshot every Friday on the `ar` file system that you created earlier.

In this practice, you will perform the following tasks:

- Create a ZFS snapshot.
- Roll back to a previous snapshot.
- Create a ZFS clone from a ZFS snapshot.
- Promote the ZFS clone into production.

Task 1: Create a ZFS Snapshot

1. Verify that the **s11-server1** VM is running. If it is not, start it at this time.
2. Verify that the **s11-testbed** VM is running. If it is not, start the VM.
3. Log in to the **s11-testbed** VM as user `oracle`, and then run the `su -` command to assume primary administrator privileges. Use the password `oracle1`.

```
oracle@s11-testbed:~$ su -
Password:
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```

4. Create a ZFS snapshot of the `ar` file system that you created earlier.

```
root@s11-testbed:~# zfs list -r finpool
NAME                USED  AVAIL  REFER  MOUNTPOINT
finpool              1.00G  8.75G  36.0K  /finpool
finpool/finfs        1.00G  8.75G  38.6K  /finpool/finfs
finpool/finfs/ar      34.6K  5.00G  34.6K  /finpool/finfs/ar
finpool/finfs/del     34.6K  5.00G  34.6K  /finpool/finfs/del
finpool/finfs/inv     34.6K  9.75G  34.6K  /finpool/finfs/inv

root@s11-testbed:~# zfs snapshot finpool/finfs/ar@friday
root@s11-testbed:~# zfs list -rt all finpool
NAME                USED  AVAIL  REFER  MOUNTPOINT
finpool              1.00G  8.75G  36.0K  /finpool
finpool/finfs        1.00G  8.75G  38.6K  /finpool/finfs
finpool/finfs/ar      34.6K  5.00G  34.6K  /finpool/finfs/ar
finpool/finfs/ar@friday  0      -    34.6K  -
finpool/finfs/del     34.6K  5.00G  34.6K  /finpool/finfs/del
finpool/finfs/inv     34.6K  9.75G  34.6K  /finpool/finfs/inv
root@s11-testbed:~#
```

Note that a snapshot is created by appending the @ symbol to the file system name, together with the name of the snapshot. It is a good practice to keep a meaningful name. (Friday, Nov11, TSmith, and so on).

The option `-t all` is used to include the snapshots in the displayed output. Without this option, the snapshots are not displayed as illustrated below.

```
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	1.00G	8.75G	36.0K	/finpool
finpool/finfs	1.00G	8.75G	38.6K	/finpool/finfs
finpool/finfs/ar	34.6K	5.00G	34.6K	/finpool/finfs/ar
finpool/finfs/del	34.6K	5.00G	34.6K	/finpool/finfs/del
finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv

```
root@s11-testbed:~#
```

Now, verify the value of the `listsnapshots` pool property.

```
oracle@s11-testbed:~# zpool get listsnapshots finpool
```

NAME	PROPERTY	VALUE	SOURCE
finpool	listsnapshots	off	default

As displayed here, the `listsnapshots` property is `off` by default. Set it `on` and verify its status.

```
oracle@s11-testbed:~# zpool set listsnapshots=on finpool
```

```
oracle@s11-testbed:~# zpool get listsnapshots finpool
```

NAME	PROPERTY	VALUE	SOURCE
finpool	listsnapshots	on	local

Now, when you display the descendent file systems of the `finpool` pool, the snapshots are included.

```
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	1.00G	8.75G	36.0K	/finpool
finpool/finfs	1.00G	8.75G	38.6K	/finpool/finfs
finpool/finfs/ar	34.6K	5.00G	34.6K	/finpool/finfs/ar
finpool/finfs/ar@friday	0	-	34.6K	-
finpool/finfs/del	34.6K	5.00G	34.6K	/finpool/finfs/del
finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv

```
root@s11-testbed:~#
```

5. Create a file named `/finpool/finfs/ar/fin2013`.

```
root@s11-testbed:~# touch /finpool/finfs/ar/fin2013
root@s11-testbed:~# ls /finpool/finfs/ar/fin2013
/finpool/finfs/ar/fin2013
```

You confirmed that it exists. Note that this file was created after taking a backup on Friday.

6. Create another recursive snapshot named `finpool/finfs/ar@saturday`.

```
root@s11-testbed:~# zfs snapshot -r finpool/finfs/ar@saturday
```

Note that the `fin2013` file will be included in the Saturday snapshot but not in the Friday snapshot.

7. Attempt to roll back the `finpool/finfs/ar` snapshot by using the `finpool/finfs/ar@friday` snapshot.

```
root@s11-testbed:~# zfs rollback finpool/finfs/ar@friday
cannot rollback to 'finpool/finfs/ar@friday': more recent snapshots
exist
use '-r' to force deletion of the following snapshots:
finpool/finfs/ar@saturday
root@s11-testbed:~#
```

Notice that more recent snapshots (`finpool/finfs/ar@saturday`) exist; therefore, you cannot roll back to an earlier snapshot unless you use the `-r` option that deletes the more recent snapshots until the `finpool/finfs/ar@friday` snapshot becomes the most recent. Do not roll back yet.

Question: If you roll back to the Friday snapshot, what data will be lost?

Answer: The file named `/finpool/finfs/ar/fin2013` will be lost.

8. Delete the file named `/finpool/finfs/ar/fin2013`.

```
root@s11-testbed:~# rm /finpool/finfs/ar/fin2013
```

Remove the `fin2013` data file to see whether you can recover it.

9. List the descendant `finpool` file systems. Roll back the `finpool/finfs/ar@saturday` snapshot.

```
root@s11-testbed:~# zfs list -r finpool
NAME                                USED  AVAIL  REFER  MOUNTPOINT
finpool                             1.00G  8.75G  36.0K  /finpool
finpool/finfs                       1.00G  8.75G  38.6K  /finpool/finfs
finpool/finfs/ar                     75.9K  5.00G  34.6K  /finpool/finfs/ar
finpool/finfs/ar@friday              20.6K   -    34.6K  -
finpool/finfs/ar@saturday            20.6K   -    34.6K  -
finpool/finfs/del                    34.6K  5.00G  34.6K  /finpool/finfs/del
finpool/finfs/inv                    34.6K  9.75G  34.6K  /finpool/finfs/inv
root@s11-testbed:~#
```

```
root@s11-testbed:~# zfs rollback finpool/finfs/ar@saturday
```

You rolled back to the `ar@saturday` backup. Does it include the `fin2013` customer file? You will find out in the next step.

10. Confirm that `/finpool/finfs/ar/fin2013` is restored.

```
root@s11-server1:~# ls /finpool/finfs/ar/fin2013
/finpool/finfs/ar/fin2013
```

Summary: The Saturday backup was taken after you created the `fin2013` data file. Therefore, you were able to recover `fin2013` from the `ar@saturday` backup.

Note: Do not perform the steps given in the following Additional Note section. The note is only for your reference.

Additional Note: If you wish to delete the snapshot, use the ZFS destroy command.

```
root@s11-testbed:~# zfs destroy finpool/finfs/ar@friday
```

```
root@s11-testbed:~# zfs list -rt all finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	1.00G	8.75G	36.0K	/finpool
finpool/finfs	1.00G	8.75G	38.6K	/finpool/finfs
finpool/finfs/ar	74.6K	5.00G	34.6K	/finpool/finfs/ar
finpool/finfs/ar@saturday	19.3K	-	34.6K	-
finpool/finfs/del	34.6K	5.00G	34.6K	/finpool/finfs/del
finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv

```
root@s11-testbed:~#
```

The snapshot has been successfully deleted.

Task 2: Create a ZFS Clone

A ZFS clone provides a writable copy of a snapshot that can be safely used to work on projections before finalizing the details on to a production copy.

1. Create a clone from the `ar@friday` snapshot and verify that it has been created.

```
root@s11-testbed:~# zfs list -r finpool
NAME                                USED  AVAIL  REFER  MOUNTPOINT
finpool                            1.00G  8.75G  36.0K  /finpool
finpool/finfs                      1.00G  8.75G  38.6K  /finpool/finfs
finpool/finfs/ar                    74.6K  5.00G  34.6K  /finpool/finfs/ar
finpool/finfs/ar@friday             20.6K      -   34.6K  -
finpool/finfs/ar@satursday          19.3K      -   34.6K  -
finpool/finfs/del                   34.6K  5.00G  34.6K  /finpool/finfs/del
finpool/finfs/inv                   34.6K  9.75G  34.6K  /finpool/finfs/inv
root@s11-testbed:~#
root@s11-testbed:~# zfs clone finpool/finfs/ar@friday \
finpool/finfs/ar/fridayclone

root@s11-testbed:~# zfs list -r finpool
NAME                                USED  AVAIL  REFER  MOUNTPOINT
finpool                            1.00G  8.75G  36.0K  /finpool
finpool/finfs                      1.00G  8.75G  38.6K  /finpool/finfs
finpool/finfs/ar                    96.6K  5.00G  36.0K  /finpool/finfs/ar
finpool/finfs/ar@friday             20.6K      -   34.6K  -
finpool/finfs/ar@satursday          20.6K      -   34.6K  -
finpool/finfs/ar/fridayclone        19.3K  5.00G  34.6K  /finpool/finfs/ar/fridayclone
finpool/finfs/del                   34.6K  5.00G  34.6K  /finpool/finfs/del
finpool/finfs/inv                   34.6K  9.75G  34.6K  /finpool/finfs/inv
root@s11-testbed:~#
```

Note that the snapshot `finpool/finfs/ar@friday` is not mounted, as displayed in the `MOUNTPOINT` column. It is therefore not accessible. On the other hand, its clone `finpool/finfs/ar/fridayclone` is mounted, which makes it accessible. The clone `fridayclone` is a read/write copy, meaning that it is modifiable as opposed to the snapshot `finpool/finfs/ar@friday`, which is read-only.

2. Compare the attributes of the snapshot and the clone.

```
root@s11-testbed:~# ls -ld /finpool/finfs/ar/fridayclone
drwxr-xr-x  2 root root  2 Sep 13 08:14 /finpool/finfs/ar/fridayclone
root@s11-testbed:~# ls -ld /finpool/finfs/ar@friday
/finpool/finfs/ar@friday: No such file or directory

root@s11-testbed:~# cd /finpool/finfs/ar/fridayclone
root@s11-testbed:/finpool/finfs/ar/fridayclone# touch newcust
root@s11-testbed:/finpool/finfs/ar/fridayclone# ls
newcust  ...
```


These commands illustrate the major difference between a snapshot and a clone. The snapshot is not available, while the clone is available and modifiable.

3. View the space usage of the clone, assuming that you made some changes in the clone.

```
root@s11-testbed:/finpool/finfs/ar/fridayclone# cd
```

```
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	1.00G	8.75G	36.0K	/finpool
finpool/finfs	1.00G	8.75G	38.6K	/finpool/finfs
finpool/finfs/ar	97.9K	5.00G	36.0K	/finpool/finfs/ar
finpool/finfs/ar@friday	20.6K	-	34.6K	-
finpool/finfs/ar@satursday	20.6K	-	34.6K	-
finpool/finfs/ar/fridayclone	20.6K	5.00G	34.6K	
/finpool/finfs/ar/fridayclone				

finpool/finfs/del	34.6K	5.00G	34.6K	/finpool/finfs/del
finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv

```
root@s11-testbed:~#
```

Note the used column for the clone. The space utilization has gone up when compared to the same column prior to editing the clone. The clone is using some storage to keep track of the new file that you created.

4. Replace the live file system with the modified clone, assuming that you want the changes to go into production.

```
root@s11-testbed:~# zfs get origin finpool/finfs/ar/fridayclone
```

NAME	PROPERTY	VALUE	SOURCE
finpool/finfs/ar/fridayclone	origin	finpool/finfs/ar@friday	-

```
root@s11-testbed:~#
```

```
root@s11-testbed:~# zfs promote finpool/finfs/ar/fridayclone
```

```
root@s11-testbed:~# zfs get origin finpool/finfs/ar/fridayclone
```

NAME	PROPERTY	VALUE	SOURCE
finpool/finfs/ar/fridayclone	origin	-	

```
root@s11-testbed:~# zfs list -r finpool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
finpool	1.00G	8.75G	36.0K	/finpool
finpool/finfs	1.00G	8.75G	38.6K	/finpool/finfs
finpool/finfs/ar	97.9K	5.00G	36.0K	/finpool/finfs/ar
finpool/finfs/ar@satursday	20.6K	-	34.6K	-
finpool/finfs/ar/fridayclone	55.3K	5.00G	34.6K	
/finpool/finfs/ar/fridayclone				

finpool/finfs/ar/fridayclone@friday	20.6K	-	34.6K	-
-------------------------------------	-------	---	-------	---

finpool/finfs/del	34.6K	5.00G	34.6K	/finpool/finfs/del
-------------------	-------	-------	-------	--------------------

finpool/finfs/inv	34.6K	9.75G	34.6K	/finpool/finfs/inv
-------------------	-------	-------	-------	--------------------

```
root@s11-testbed:~#
```

Summary: If you do the math, the used space of the clone `fridayclone` now reflects the total of the main file system and the clone, that is, 34.6 KB + 20.6 KB = 55.2 KB. This means that the new file `newcust` in the clone has been added to `fridayclone`.

Note: Do not perform the steps given in the following Additional Note section. The note is only for your reference.

Additional Note: If you wish to delete a clone, use the following command.

```
root@s11-testbed:~# zfs destroy finpool/finfs/ar/fridayclone  
root@s11-testbed:~# zfs list -rt all finpool
```

The clone has been successfully deleted.

Practice 4-6: Performing ZFS Data Backup and Recovery

Overview

Data backup and recovery are crucial to maintaining data high availability (HA). As a system administrator, you will be required to perform backup and recovery procedures on an ongoing basis. As of now, it is a good time to explore the options available with ZFS to back up and recover data.

For instance, create an archive of `labpool` on **s11-testbed** and store it on a remote server, **s11-server1**. Assuming that the data on `labpool` residing on **s11-testbed** has been corrupted, recover the data in entirety from the archive in **s11-server1**.

In this practice, you will perform the following tasks:

- Create a data archive.
- Store the archive on the remote server.
- Recover ZFS data from the remote server.

Task: Create a Data Archive

1. Verify that the **s11-server1** and **s11-testbed** VMs are running. If not, start the VMs.
2. Log in to the **s11-server1** and **s11-testbed** VMs as user `oracle`, and then run the `su -` command to assume primary administrator privileges. Use the password `oracle1`.
3. Log in to **s11-server1** and create the `backuppool` by using disk `c7t2d0`. Verify the action.

```
root@s11-server1:~# zpool create backuppool c7t2d0
```

Note: If you find any issues in `backuppool` creation, use `-f` option in the command as follows.

```
root@s11-server1:~# zpool create -f backuppool c7t2d0
```

```
root@s11-server1:~# zpool status backuppool
```

```
pool: backuppool
state: ONLINE
scan: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
backuppool	ONLINE	0	0	0
c7t2d0	ONLINE	0	0	0

```
errors: No known data errors
```

Now, verify the value of the `listsnapshots` pool property.

```
root@s11-server1:~# zpool get listsnapshots backuppool
```

NAME	PROPERTY	VALUE	SOURCE
backuppool	listsnapshots	off	default

As displayed here, the `listsnapshots` property is off by default. Set it to on and verify its status.

```
root@s11-server1:~# zpool set listsnapshots=on backuppool
```

```
root@s11-server1:~# zpool get listsnapshots backuppool
```

NAME	PROPERTY	VALUE	SOURCE
backuppool	listsnapshots	on	local

4. On **s11-testbed**, create the **serverapps** and **appanalysts** files under the **labfs** file system.

```
root@s11-testbed:~# cd /labpool/labfs
root@s11-testbed:/labpool/labfs# echo serverapps > serverapps
root@s11-testbed:/labpool/labfs# echo appanalysts > appanalysts
root@s11-testbed:/labpool/labfs# cd
root@s11-testbed:~#
```

5. Create the **labpool**'s recursive snapshots. Verify the action.

```
root@s11-testbed:~# zpool set listsnapshots=on labpool
root@s11-testbed:~# zfs snapshot -r labpool@snap1
root@s11-testbed:~# zfs list -r labpool
```

NAME	USED	AVAIL	REFER	MOUNTPPOINT
labpool	144K	976M	32K	/labpool
labpool@snap1	0	-	32K	-
labpool/labfs	32K	976M	32K	/labpool/labfs
labpool/labfs@snap1	0	-	32K	-

```
root@s11-testbed:~#
```

Store the Archive on the Remote Server

6. On **s11-testbed**, generate the public/private RSA key pair and save it in the **id_migrate** file.

```
root@s11-testbed:~# ssh-keygen -t rsa -P "" -f ~/id_migrate
Generating public/private rsa key pair.
Your identification has been saved in /root/id_migrate.
Your public key has been saved in /root/id_migrate.pub.
The key fingerprint is:
06:40:32:f5:ce:22:12:5c:50:8d:7f:5f:d3:f9:1a:21 root@s11-testbed
root@s11-testbed:~#
```

7. From **s11-testbed**, copy your public key to the oracle account on **s11-server1**.

```
root@s11-testbed:~# scp ~/id_migrate.pub oracle@s11-server1:id_migrate.pub
The authenticity of host 's11-server1 (192.168.0.100)' can't be
established.
RSA key fingerprint is
02:07:95:c7:a1:75:42:94:71:09:8f:11:6c:f9:38:1d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 's11-server1,192.168.0.100' (RSA) to the
list of known hosts.
Password:
id_migrate.pub          100% |*****| 398
00:00
root@s11-testbed:~#
```

8. Log in to the destination system **s11-server1**, create a directory called **.ssh**, and copy the public key into the **authorized_keys** file. Exit to the administrator account.

```
root@s11-server1:~# su - oracle
Oracle Corporation      SunOS 5.11      11.1      September 2012
oracle@s11-server1:~$ mkdir -m 700 .ssh
oracle@s11-server1:~$ ls
id_migrate.pub
oracle@s11-server1:~$ cat id_migrate.pub >> .ssh/authorized_keys
oracle@s11-server1:~$ exit
logout
root@s11-server1:~#
```

9. From **s11-testbed**, send the **labpool@snap1** snapshot and its children snapshots to **s11-server1** **backuppools** as a bundle.

```
root@s11-testbed:~# zfs send -Rv labpool@snap1 | ssh -l oracle -i
~/id_migrate s11-server1 pfexec /usr/sbin/zfs recv -F backuppool
sending from @ to labpool@snap1
sending from @ to labpool/labfs@snap1
root@s11-testbed:~#
```

10. Assume that the data on **labpool** has been corrupted and you are forced to delete it.

```
root@s11-testbed:~# zpool destroy labpool
root@s11-testbed:~# zpool list labpool
cannot open 'labpool': no such pool
root@s11-testbed:~#
```

11. Recreate **labpool** so that when data is recovered it can reside in the recreated pool. Verify the action.

```
root@s11-testbed:~# zpool create labpool c7t2d0
root@s11-testbed:~# zpool status labpool
  pool: labpool
state: ONLINE
  scan: none requested
config:

    NAME      STATE    READ WRITE CKSUM
    labpool   ONLINE      0     0     0
      c7t2d0  ONLINE      0     0     0

errors: No known data errors
root@s11-testbed:~# zfs list -r labpool
NAME      USED  AVAIL  REFER  MOUNTPOINT
labpool   85K   976M   31K    /labpool
root@s11-testbed:~# zpool list labpool
NAME      SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
labpool   1008M   85K   1008M  0%   1.00x  ONLINE  -
```

The `listsnapshots` property is off by default for `labpool`. Set it to on and verify its status.

```
root@s11-testbed:~# zpool set listsnapshots=on labpool
root@s11-testbed:~# zpool get listsnapshots labpool
```

NAME	PROPERTY	VALUE	SOURCE
labpool	listsnapshots	on	local

12. On `s11-server1`, verify that the snapshot archive exists and the files are available.

```
root@s11-server1:~# zfs list -r backuppool
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
backuppool	138K	39.1G	32K	/backuppool
backuppool@snap1	0	-	32K	-
backuppool/labfs	32K	39.1G	32K	/backuppool/labfs
backuppool/labfs@snap1	0	-	32K	-

```
root@s11-server1:~# cd /backuppool/labfs/
root@s11-server1:/backuppool/labfs# ls
appanalysts  serverapps
root@s11-server1:/backuppool/labfs# cd
root@s11-server1:~#
```

Recover ZFS Data from the Remote Server

13. Set up the user `oracle` authentication from `s11-server1` to `s11-testbed` as you did earlier (from `s11-testbed` to `s11-server1`).

```
root@s11-server1:~# ssh-keygen -t rsa -P "" -f ~/id_migrate
Generating public/private rsa key pair.
Your identification has been saved in /root/id_migrate.
Your public key has been saved in /root/id_migrate.pub.
The key fingerprint is:
fa:6f:d6:8b:8a:26:a0:b9:ef:6a:68:ea:6b:f9:d6:89 root@s11-server1
root@s11-server1:~#
root@s11-server1:~# scp ~/id_migrate.pub oracle@s11-
testbed:id_migrate.pub
```

Note: While establishing the authentication, if the system displays a message and asks you a question about storing the key, respond with a `yes`.

```
The authenticity of host 's11-testbed (192.168.0.111)' can't be
established.
RSA key fingerprint is
40:fc:c3:78:a5:dc:28:50:fa:13:ad:fd:b4:3a:12:5e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 's11-testbed,192.168.0.111' (RSA) to the
list of known hosts.
Password:
```

```
id_migrate.pub      100% |*****|      398
00:00
root@s11-server1:~#
```

14. Complete the authentication steps on s11-testbed.

```
root@s11-testbed:~# su - oracle
Oracle Corporation      SunOS 5.11  11.1  September 2012
oracle@s11-testbed:~$ mkdir -m 700 .ssh
oracle@s11-testbed:~$ ls
Desktop      Documents    Downloads    id_migrate.pub  Public
oracle@s11-testbed:~$ cat id_migrate.pub >> .ssh/authorized_keys
oracle@s11-testbed:~$ exit
logout
root@s11-testbed:~#
```

15. From s11-server1, send the labpool archive to s11-testbed.

```
root@s11-server1:~# zfs send -Rv backuppool@snap1 | ssh -l oracle -i
~/id_migrate s11-testbed pfexec /usr/sbin/zfs recv -F labpool
sending from @ to backuppool@snap1
sending from @ to backuppool/labfs@snap1
root@s11-server1:~#
```

16. On s11-testbed, verify the recovery of the pool and the serverapps files.

```
root@s11-testbed:~# zfs list -r labpool
NAME                USED  AVAIL  REFER  MOUNTPOINT
labpool              138K  976M   32K    /labpool
labpool@snap1        0      -    32K    -
labpool/labfs        32K  976M   32K    /labpool/labfs
labpool/labfs@snap1  0      -    32K    -
root@s11-testbed:~# zfs list -r /labpool
NAME                USED  AVAIL  REFER  MOUNTPOINT
labpool              138K  976M   32K    /labpool
labpool@snap1        0      -    32K    -
labpool/labfs        32K  976M   32K    /labpool/labfs
labpool/labfs@snap1  0      -    32K    -
root@s11-testbed:~# cd /labpool/labfs/
root@s11-testbed:/labpool/labfs# ls
appanalysts  serverapps
root@s11-testbed:/labpool/labfs# cat appanalysts
appanalysts
root@s11-testbed:/labpool/labfs# cat serverapps
serverapps
root@s11-testbed:/labpool/labfs# cd
```

Summary: Observe that the labpool archive has been recovered successfully from the s11-server1 system.

Practices for Lesson 5: Configuring the Network

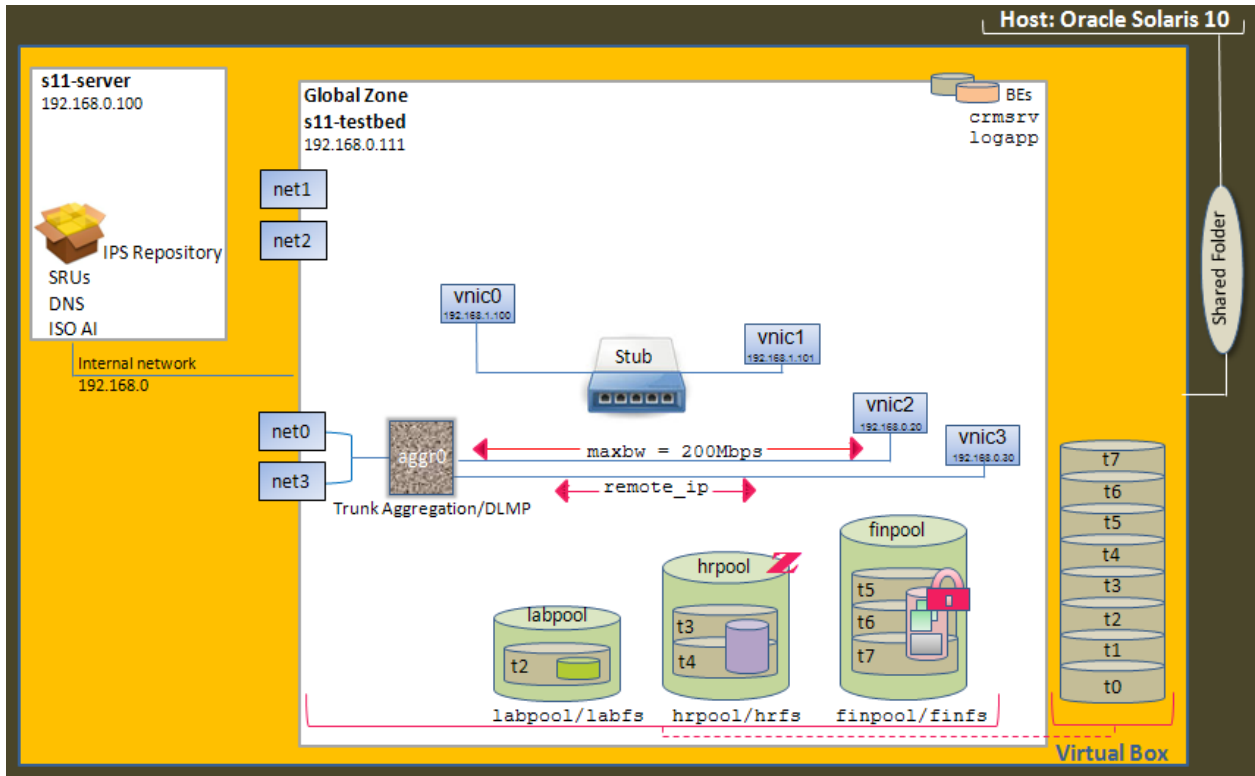
Chapter 5

Practices for Lesson 5: Overview

Practices Overview

Network virtualization is a powerful technology that provides immense possibilities in consolidating distributed computing environment on a single system. For the testbed environment, you will use the basic building blocks of network virtualization, such as VNICs and etherstubs to create a network-in-a-box setup. In addition, you need to configure link aggregation to address intense bandwidth requirements of the evolving infrastructure. Finally, you need to implement resource control so that network resources are optimally distributed and used.

Below is the diagrammatic representation of the tasks you will accomplish in this practice.



In this practice, you will perform the following:

- Initialize the network setup.
- Create a private virtual network.
- Configure link aggregation in the global zone.
- Regulate bandwidth by using datalink properties.
- Manage the virtual network data flow.

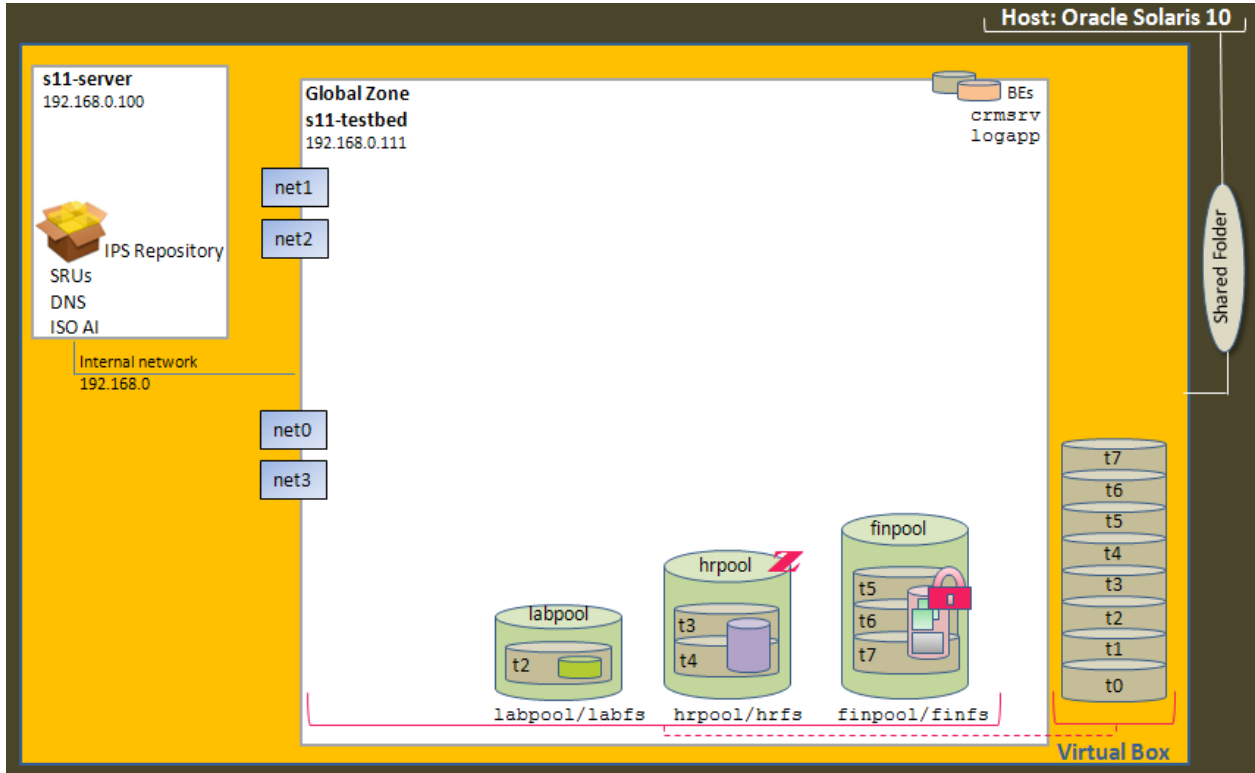
Notes

- Some command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.

Practice 5-1: Initializing the Network Setup

Overview

Before setting up the network environment, it is useful to identify the substrate network fabric and gather some fundamental details about the platform that is available for establishing the network infrastructure.



In this practice, you will perform the following tasks:

- Identify the network configuration of the **s11-server1** VM.
- Identify the network configuration of the **s11-testbed** VM.
- Identify the active network configuration profiles.
- Identify the available network interface.

Task 1: Identify the network configuration of the s11-server1 VM

1. Verify that the **s11-server1** and **s11-testbed** VMs are running.
2. Log in to the **s11-server1** VM as the `oracle` user with the password, `oracle1` and assume administrator privileges.

```
oracle@ s11-server1:~$ su -  
Password: oracle1  
Oracle Corporation          SunOS 5.11 11.1          September 2012
```

3. Display information about the physical attributes of the datalinks currently on the **s11-server1** VM.

```
root@s11-server1:~# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	1000	full	e1000g0
net1	Ethernet	unknown	0	unknown	e1000g1
net2	Ethernet	unknown	0	unknown	e1000g2
net3	Ethernet	unknown	0	unknown	e1000g3

Note: The **s11-server1** VM has the following:

- Four physical network interface cards: net0, net1, net2, and net3
- Only net0 is configured.
- The hardware-based link name is net0.
- Media is Ethernet.
- Device state is up.
- Data transfer speed is 1000 MB.
- Duplex state is full, which means that there can be two-way data transmission.
- Device type is e1000g0, which refers to the Intel gigabit controller type device.

4. Find the active network configuration profile by using the `netadm` command.

```
root@s11-server1:~# netadm list
```

TYPE	PROFILE	STATE
ncp	Automatic	disabled
ncp	DefaultFixed	online
loc	Automatic	offline
loc	NoNet	offline
loc	DefaultFixed	online

Alternatively, active configuration profile information can be obtained by using the `svcprop` command.

```
root@s11-server1:~# svcprop network/physical:default | grep -i netcfg/active_ncp
netcfg/active_ncp astring DefaultFixed
root@s11-server1:~#
```

5. Display the configured interfaces by using the `ipadm show-if` command.

```
root@s11-server1:~# ipadm show-if
```

IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--
net0	ip	ok	yes	--

```
root@s11-server1:~#
```

6. Display the address information of the interface by using the `ipadm` command.

```
root@s11-server1:~# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
net0/v4       static    ok         192.168.0.100/24
lo0/v6       static    ok         ::1/128
net0/v6       addrconf  ok         fe80::a00:27ff:fe46:7201/10
```

7. New with Oracle Solaris 11.1, the `ipadm` command can be used without a subcommand to display both the configured interfaces and their addresses.

```
root@s11-server1:~# ipadm
NAME          CLASS/TYPE  STATE  UNDER  ADDR
lo0           loopback   ok     --      --
  lo0/v4      static     ok     --      127.0.0.1/8
  lo0/v6      static     ok     --      ::1/128
net0          ip         ok     --      --
  net0/v4     static     ok     --      192.168.0.100/24
  net0/v6     addrconf   ok     --      fe80::a00:27ff:feee:3bfd/10
root@s11-server1:~#
```

Summary: The IP version used in this network is IPv4. The IP address assigned to the system is 192.168.0.100 and is static.

Task 2: Identify the network configuration of the `s11-testbed` VM

1. Open a new terminal in the **s11-testbed** VM. The default user is `oracle` and the password is `oracle1`.
2. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1` and assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation      SunOS 5.11 11.1          September 2012
root@s11-testbed:~#
```

3. Display information about the datalinks on the **s11-testbed** VM.

```
root@s11-testbed:~# dladm show-link
LINK          CLASS      MTU      STATE      OVER
net1          phys      1500     unknown    --
net2          phys      1500     unknown    --
net3          phys      1500     unknown    --
net0          phys      1500     up         --
root@s11-testbed:~#
```

4. Display information about the physical attributes of the datalinks on the **s11-testbed** VM.

```
root@s11-testbed:~# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	1000	full	e1000g0
net1	Ethernet	unknown	0	unknown	e1000g1
net2	Ethernet	unknown	0	unknown	e1000g2
net3	Ethernet	unknown	0	unknown	e1000g3

5. Display the configured interfaces by using the `ipadm show-if` command.

```
root@s11-testbed:~# ipadm show-if
```

IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--
net0	ip	ok	yes	--

```
root@s11-testbed:~#
```

6. Display the address information of the network interfaces.

```
root@s11-testbed:~# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
net0/v4	static	ok	192.168.0.111/24
lo0/v6	static	ok	:::1/128
net0/v6	addrconf	disabled	::

7. New with Oracle Solaris 11.1, the `ipadm` command can be used without a subcommand to display both the configured interfaces and their addresses.

```
root@s11-testbed:~# ipadm
```

NAME	CLASS/TYPE	STATE	UNDER	ADDR
lo0	loopback	ok	--	--
lo0/v4	static	ok	--	127.0.0.1/8
lo0/v6	static	ok	--	:::1/128
net0	ip	ok	--	--
net0/v4	static	ok	--	192.168.0.111/24
net0/v6	addrconf	disabled	--	::

```
root@s11-testbed:~#
```

Summary: The IP version used in this network is IPv4. The IP address assigned to the system is 192.168.0.111 and is static.

Task 3: Assess the Current Reactive Network Configuration

Reactive network is a technology that simplifies and automates network configuration on Oracle Solaris 11.1. The key reactive network components are the network profiles, which allow you to specify various network configurations to be created depending on the current network conditions. In this task, you assess the current reactive network configuration.

1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1` and assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation          SunOS 5.11 11.1          September 2012
root@s11-testbed:~#
```

2. List all available reactive network profiles and their current state.

```
root@s11-testbed:~# netadm list
TYPE          PROFILE      STATE
ncp           Automatic    disabled
ncp           start_state  online
ncu:phys      net0         online
ncu:ip        net0         online
ncp           DefaultFixed disabled
loc           Automatic    offline
loc           NoNet        offline
loc           aces         online
```

3. List the reactive network `Automatic` profile.

```
root@s11-testbed:~# netadm list Automatic
TYPE          PROFILE      STATE
ncp           Automatic    disabled
loc           Automatic    offline
```

4. List the reactive network `start_state` profile.

```
root@s11-testbed:~# netadm list start_state
TYPE          PROFILE      STATE
ncp           start_state  online
ncu:phys      net0         online
ncu:ip        net0         online
```

5. List the reactive network location profiles.

```
root@s11-testbed:~# netadm list -p loc
TYPE          PROFILE      STATE
loc           Automatic    offline
loc           NoNet        offline
loc           aces         online
```


6. List all the `phys` and `ip` network configuration units (NCUs) in the active network configuration profiles (NCPs).

```
root@s11-testbed:~# netadm list -c phys
TYPE          PROFILE      STATE
ncu:phys      net0          online
root@s11-testbed:~# netadm list -c ip
TYPE          PROFILE      STATE
ncu:ip        net0          online
```

7. List all the reactive network profiles and their auxiliary state.

```
root@s11-testbed:~# netadm list -x
TYPE          PROFILE      STATE      AUXILIARY STATE
ncp           Automatic    disabled    disabled by administrator
ncp           start_state  online      active
ncu:phys      net0         online      interface/link is up
ncu:ip        net0         online      interface/link is up
ncp           DefaultFixed disabled     disabled by administrator
loc           Automatic    offline     conditions for activation are unmet
loc           NoNet        offline     conditions for activation are unmet
loc           aces         online      active
```

8. Use the `netcfg` utility to select the `start_state` profile and list its NCUs.

```
root@s11-testbed:~# netcfg
netcfg> select ncp start_state
netcfg:ncp:start_state> list
ncp:start_state
    management-type reactive
NCUs:
    phys    net0
    ip      net0
```

9. Select the `phys` NCU and display its properties.

```
netcfg:ncp:start_state> select ncu phys net0
netcfg:ncp:start_state:ncu:net0> list
ncu:net0
    type          link
    class         phys
    parent        "start_state"
    activation-mode manual
    enabled        true
netcfg:ncp:start_state:ncu:net0> end
```

10. Select the `ip` NCU and display its properties.

```
netcfg:ncp:start_state> select ncu ip net0
netcfg:ncp:start_state:ncu:net0> list
ncu:net0
    type          interface
    class         ip
    parent        "start_state"
```

```

        enabled                true
        ip-version              ipv4
        ipv4-addrsrc            static
        ipv4-addr               "192.168.0.111/24"
        ipv6-addrsrc            dhcp,autoconf
netcfg:ncp:start_state:ncu:net0> end
netcfg:ncp:start_state> end
netcfg>

```

11. Select the `aces` location profile and list its properties.

```

netcfg> select loc aces
netcfg:loc:aces> list
loc:aces
        activation-mode        conditional-all
        conditions              "system domain is mydomain.com"
        enabled                 true
        nameservices            dns
        nameservices-config-file "/etc/nsswitch.dns"
        dns-nameservice-configsrc manual
        dns-nameservice-domain  "mydomain.com"
        dns-nameservice-servers "192.168.0.100"
netcfg:loc:aces> end
netcfg> exit
root@s11-testbed:~# exit

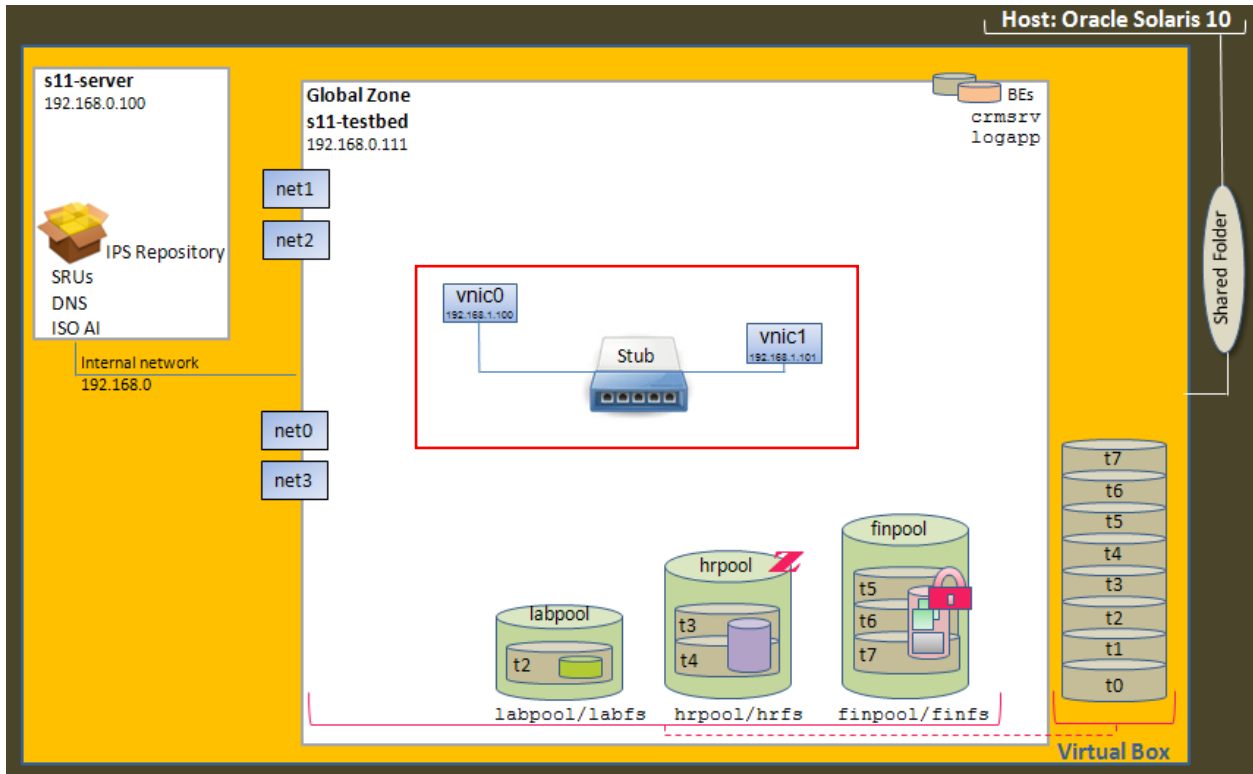
```

12. Close the terminal window.

Practice 5-2: Creating a Private Virtual Network

Overview

In a later practice, you will create some zones in the testbed environment for workload separation. The kind of applications that these zones will host must be protected from the larger network and the outside world. Recall, that etherstubs help create private virtual networks. For now, you will create the network infrastructure and keep it ready. It will be a little while before you can deploy the framework.



In this practice, you will perform the following tasks:

- Create an etherstub `stub0`.
- Create `vnic0` and `vnic1` from `stub0`.

Task:

1. Verify that the **s11-testbed** VM is running. If the VM is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.
3. Display the link status by using the `show-link` subcommand.

```
root@s11-testbed:~# dladm show-link
LINK          CLASS      MTU      STATE    OVER
net1          phys      1500     unknown --
net2          phys      1500     unknown --
net3          phys      1500     unknown --
net0          phys      1500     up       --
```

4. Before you create the VNICs, you need to create an etherstub (private virtual switch). Create an etherstub named `stub0` by using the `dladm` utility.

```
root@s11-testbed:~# dladm create-etherstub stub0
```

Confirm the creation of the etherstub by using the `show-link` subcommand.

```
root@s11-testbed:~$ dladm show-link
```

LINK	CLASS	MTU	STATE	OVER
net1	phys	1500	unknown	--
net2	phys	1500	unknown	--
net3	phys	1500	unknown	--
net0	phys	1500	up	--
stub0	etherstub	9000	unknown	--

```
root@s11-testbed:~#
```

5. Create `vnic0` and `vnic1` VNICs by using the `dladm` command and attach them to the etherstub.

```
root@s11-testbed:~# dladm create-vnic -l stub0 vnic0
root@s11-testbed:~# dladm create-vnic -l stub0 vnic1
```

6. Display the VNIC details.

```
root@s11-testbed:~# dladm show-vnic
```

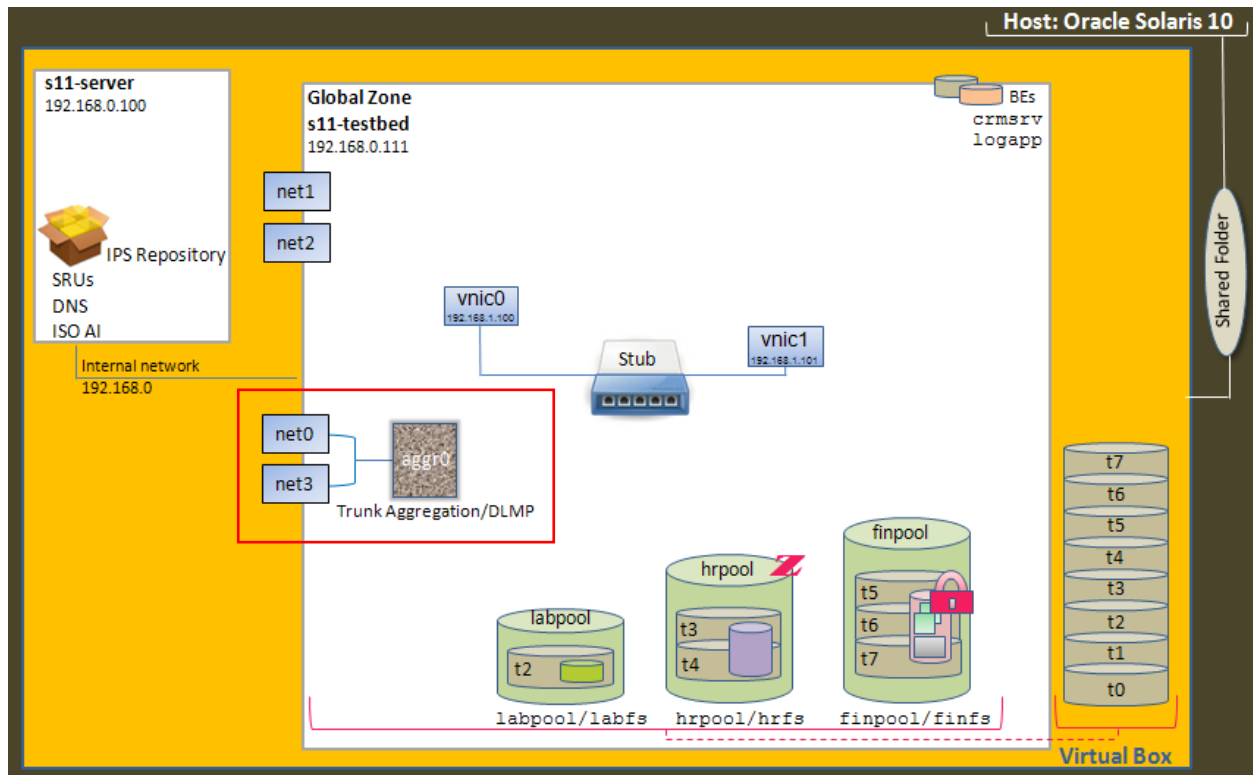
LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VID
vnic0	stub0	40000	2:8:20:84:d:cb	random	0
vnic1	stub0	40000	2:8:20:a:97:10	random	0

Summary: The two VNICs have been created as displayed. Notice that each VNIC has a randomly assigned MAC address.

Practice 5-3: Configuring Link Aggregation in the Global Zone

Overview

Link aggregation allows multiple NICs to be grouped into a single logical interface. Link aggregation requires at least two network interfaces. Link aggregations are useful for increasing bandwidth as well as for providing high availability (HA). Based on the requirement for intense bandwidth allocation for the global zone, you now configure trunk aggregation on the `net0` and `net3` interfaces. Note that the network interfaces must be unplumbed before they can be aggregated.



In this practice, you will perform the following tasks:

- Dismantle the network interface configured on the global zone.
- Configure trunk aggregation.
- Configure Datalink Multipathing (DLMP).

Task 1: Dismantle the network interface configured on the global zone

To configure trunk aggregation, you require two interfaces: `net0` and `net3`. Because the `net0` interface has already been configured on the global zone, you need to dismantle and repurpose it for creating a larger network pipe.

1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1`.

- Switch to the root role by using the `su -` command and the password `oracle1`.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```

- Display information about the physical attributes of the datalinks currently on the system.

```
root@s11-testbed:~# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net1	Ethernet	unknown	0	unknown	e1000g1
net2	Ethernet	unknown	0	unknown	e1000g2
net3	Ethernet	unknown	0	unknown	e1000g3
net0	Ethernet	up	1000	full	e1000g0

```
root@s11-testbed:~#
```

- Display the link status by using the `show-link` subcommand.

```
root@s11-testbed:~# dladm show-link
```

LINK	CLASS	MTU	STATE	OVER
net1	phys	1500	unknown	--
net2	phys	1500	unknown	--
net3	phys	1500	unknown	--
net0	phys	1500	up	--
stub0	etherstub	9000	unknown	--
vn1c0	vn1c	9000	up	stub0
vn1c1	vn1c	9000	up	stub0

- Display the IP address for the active interface.

```
root@s11-testbed:~# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
net0/v4	static	ok	192.168.0.111/24
lo0/v6	static	ok	:::1/128
net6/v4	addrconf	disabled	::

- Remove the `net0` interface.

```
root@s11-testbed:~# ipadm delete-addr net0/v4
root@s11-testbed:~# ipadm delete-ip net0
root@s11-testbed:~# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
lo0/v6	static	ok	:::1/128
net0/v4	addrconf	disabled	::

Summary: `net0` has been unplumbed. `net3` is also available. You can now use both `net0` and `net3` for trunk aggregation.

Task 2: Configure trunk aggregation

1. Create an aggregation, `aggr0`, over the `net0` and `net3` interfaces.

```
root@s11-testbed:~# dladm create-aggr -l net0 -l net3 aggr0
```

2. Verify that the aggregation is configured.

```
root@s11-testbed:~# dladm show-link
```

LINK	CLASS	MTU	STATE	OVER
net1	phys	1500	unknown	--
net2	phys	1500	unknown	--
net3	phys	1500	up	--
net0	phys	1500	up	--
stub0	etherstub	9000	unknown	--
vn00	vn00	9000	up	stub0
vn01	vn01	9000	up	stub0
aggr0	aggr	1500	up	net0 net3

Link states may vary between systems depending on their usage in the setup.

3. Display detailed information about the `aggr0` aggregation.

```
root@s11-testbed:~# dladm show-aggr
```

LINK	MODE	POLICY	ADDRPOLICY	LACPACTIVITY	LACPTIMER
aggr0	trunk	L4	auto	off	short

4. Assign static IP to the `aggr0` trunk aggregation.

```
root@s11-testbed:~# ipadm create-ip aggr0
root@s11-testbed:~# ipadm create-addr -T static -a 192.168.0.10
aggr0/ipv4
root@s11-server:~# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
aggr0/ipv4	static	ok	192.168.0.10/24
lo0/v6	static	ok	::1/128
net0/v4	addrconf	disabled	::

5. In the **s11-server1** VM, confirm that the `aggr0` aggregation is configured and active by using the `ping` command.

```
oracle@s11-server1:~$ ping 192.168.0.10
192.168.0.10 is alive
```

Summary: The `aggr0` aggregation is now available and can be utilized by the global zone for additional bandwidth requirements.

Task 3: Configure DLMP

HA at the datalink level can be achieved through DLMP. However, because of the limited interfaces in a virtual box setup and the requirement for physical switches, you will be unable to configure DLMP. Configuring DLMP involves a simple step of mentioning the mode type in the `dladm create-aggr` command. You can switch back to **s11-testbed** to perform the subsequent steps.

1. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```

2. Modify link aggregation to DLMP.

```
root@s11-testbed:~# dladm modify-aggr -m dlmp aggr0
```

For DLMP, no switch-level configuration is required.

3. Display detailed information about DLMP.

```
root@s11-testbed:~# dladm show-aggr
LINK          MODE  POLICY  ADDRPOLICY  LACPACTIVITY  LACPTIMER
aggr0         dlmp  --      --          --            --
```

Summary: Observe that `aggr0` shows DLMP under the `MODE` column.

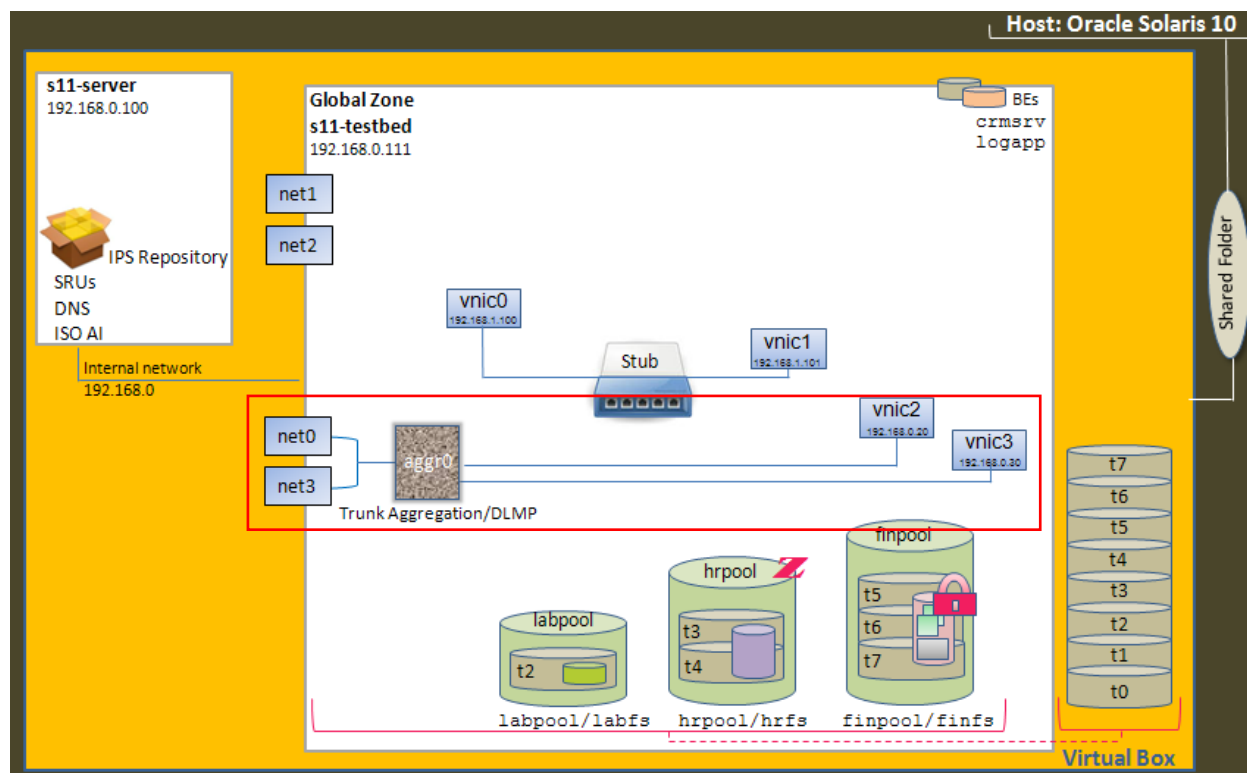
Practice 5-4: Regulating Bandwidth by Using the Datalink Properties

Overview

Considering that you are creating the initial framework of the testbed environment, you need to provision certain resources in anticipation of future requirements.

The `aggr0` aggregation that you just created is currently assigned to the global zone. However, you can go a step further and configure additional VNICs, `vnic2` and `vnic3`, on the aggregation and assign them to various resources in the global zone. In this case, you derive the cumulative benefit of the virtualization technology and HA. You can further reinforce this setup with resource control to ensure that network resources are judiciously used.

While the two VNICs you create are entitled to the total bandwidth of the aggregation, you need to allocate designated bandwidths to the two VNICs—600 Mbps to `vnic2` and 400 Mbps to `vnic3`.



In this practice, you will perform the following tasks:

- Create `vnic2` and `vnic3` from the `aggr0` aggregation.
- Assign IP address to VNICs.
- Allocate designated bandwidth to the VNICs.

Tasks

1. Log in to the **s11-testbed** VM as the **oracle** user. Use **oracle1** as the password. Assume administrator privileges.
2. Display detailed information about the **aggr0** aggregation,

```
root@s11-testbed:~# dladm show-aggr
LINK          MODE  POLICY  ADDRPOLICY  LACPACTIVITY  LACPTIMER
aggr0         dlmp  --      --           --            --
root@s11-testbed:~#
```

3. Create the **vnic2** and **vnic3** VNICs on the **aggr0** aggregation by using the **dladm** utility.

```
root@s11-testbed:~# dladm create-vnic -l aggr0 vnic2
root@s11-testbed:~# dladm create-vnic -l aggr0 vnic3
```

4. Display the result of the preceding step.

```
root@s11-testbed:~# dladm show-vnic
LINK          OVER      SPEED  MACADDRESS      MACADDRTYPE  VID
vnic0         stub0     40000  2:8:20:51:82:91  random       0
vnic1         stub0     40000  2:8:20:cb:ba:e2  random       0
vnic2         aggr0     1000   2:8:20:b0:8e:7e  random       0
vnic3         aggr0     1000   2:8:20:51:f4:93  random       0
root@s11-testbed:~#
```

5. Determine the state of all the links that are currently configured in the system by using the **dladm show-link** command.

```
root@s11-testbed:~# dladm show-link
LINK          CLASS     MTU    STATE    OVER
net1          phys     1500   unknown  --
net2          phys     1500   unknown  --
net3          phys     1500   up       --
net0          phys     1500   up       --
stub0         etherstub 9000   unknown  --
vnic0         vnic     9000   up       stub0
vnic1         vnic     9000   up       stub0
aggr0         aggr     1500   up       net0 net3
vnic2         vnic     1500   up       aggr0
vnic3         vnic     1500   up       aggr0
root@s11-testbed:~#
```

6. Assign an IP address to the vnic2 and vnic3 VNics by using the `ipadm create-addr` command and display the interfaces.

```

root@s11-testbed:~# ipadm create-ip vnic2
root@s11-testbed:~# ipadm create-ip vnic3
root@s11-testbed:~# ipadm create-addr -T static -a 192.168.0.20/24 vnic2
vnic2/v4
root@s11-testbed:~# ipadm create-addr -T static -a 192.168.0.30/24 vnic3
vnic3/v4
root@s11-testbed:~# ipadm show-addr
ADDROBJ          TYPE      STATE      ADDR
lo0/v4           static    ok         127.0.0.1/8
aggr0/ipv4       static    ok         192.168.0.10/24
vnic2/v4         static    ok         192.168.0.20/24
vnic3/v4         static    ok         192.168.0.30/24
lo0/v6           static    ok         ::1/128
net0/v6          addrconf disabled  ::

```

7. Set the bandwidth of vnic2 to 600 Mbps and vnic3 to 400 Mbps by using the `dladm set-linkprop` command and display the results.

```

root@s11-testbed:~# dladm set-linkprop -p maxbw=600m vnic2
root@s11-testbed:~# dladm set-linkprop -p maxbw=400m vnic3
root@s11-testbed:~# dladm show-vnic
LINK              OVER      SPEED  MACADDRESS      MACADDRTYPE  VID
vnic0             stub0     40000  2:8:20:51:82:91  random       0
vnic1             stub0     40000  2:8:20:cb:ba:e2  random       0
vnic2             aggr0     600    2:8:20:b0:8e:7e  random       0
vnic3             aggr0     400    2:8:20:51:f4:93  random       0
root@s11-testbed:~#

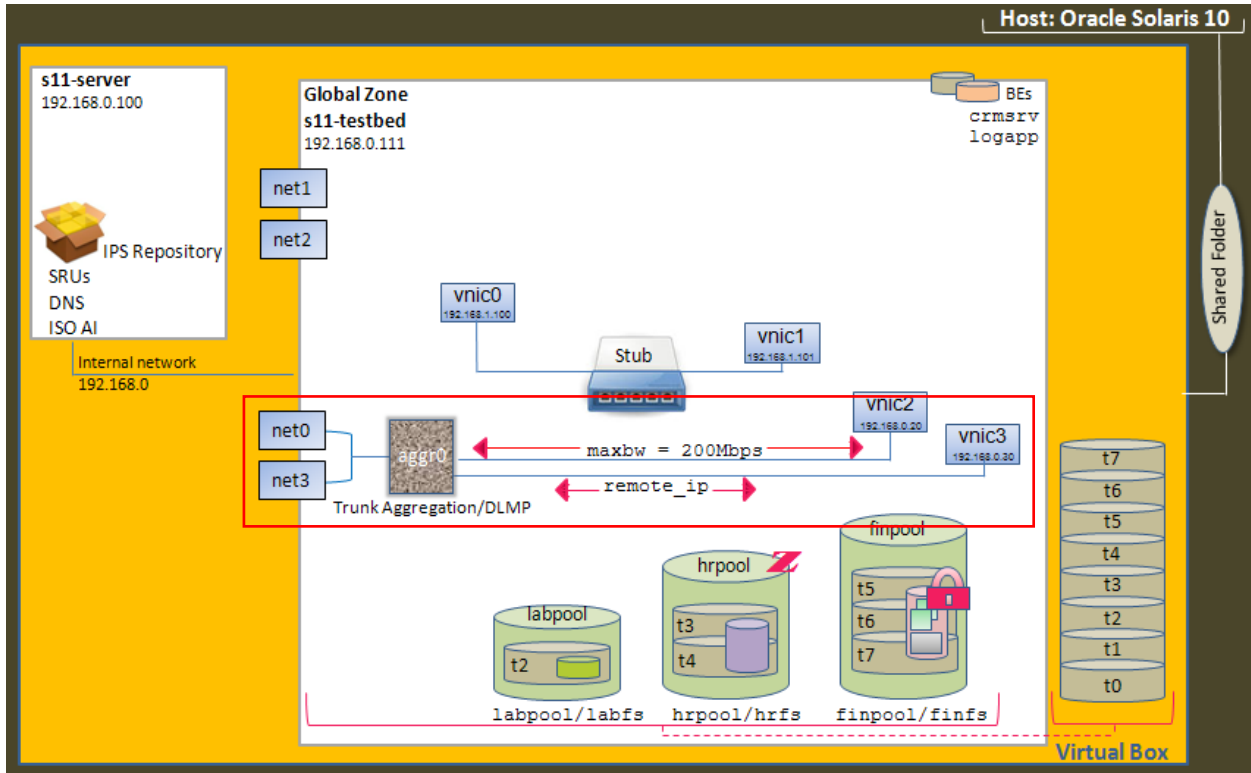
```

Summary: The two VNics, vnic2 and vnic3, have the designated bandwidth and can now be assigned to any resource, such as a web server, application server, or zone, based on further requirements.

Practice 5-5: Regulating Bandwidth by Using Flows

Overview

Assuming `vnic2` is eventually assigned to the Oracle iPlanet web server, you would like to ensure that network traffic on `vnic2` does not overwhelm the web server. To that effect, you need to create a flow on the IP address (`remote_ip`) attribute and configure it with a maximum bandwidth of 200 Mbps.



In this practice, you will configure a flow to regulate bandwidth based on a specific attribute.

Tasks

1. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.

- Determine the state of all the links that are currently configured in the system by using `dladm show-link`.

```
root@s11-testbed:~# dladm show-link
LINK          CLASS      MTU      STATE    OVER
net1          phys       1500     unknown --
net2          phys       1500     unknown --
net3          phys       1500     up       --
net0          phys       1500     up       --
stub0         etherstub  9000     unknown --
vnic0         vnic       9000     up       stub0
vnic1         vnic       9000     up       stub0
aggr0         aggr       1500     up       net0 net3
vnic2         vnic       1500     up       aggr0
vnic3         vnic       1500     up       aggr0
root@s11-testbed:~#
```

- Create a flow called `http1` on `vnic2` by using the `flowadm` command. Define this traffic to port 80 and display the results.

```
root@s11-testbed:~# flowadm add-flow -l vnic2 -a
transport=tcp,local_port=80 http1

root@s11-server1:~# flowadm show-flow
FLOW          LINK          IPADDR          PROTO  LPORT  RPORT  DSFLD
http1         vnic2          --              tcp    80     --     --
```

- Set the maximum bandwidth of 200 Mbps on the `http1` flow and display the results.

```
root@s11-testbed:~# flowadm set-flowprop -p maxbw=200m http1
root@s11-testbed:~# flowadm show-flowprop http1
FLOW          PROPERTY      VALUE          DEFAULT      POSSIBLE
http1         maxbw         200           --           --
```

Summary: The maximum bandwidth for port 80 that responds to HTTP traffic has been capped to 200 Mbps. Based on business application transactions at the workplace, you may have to manipulate bandwidth allocation accordingly.

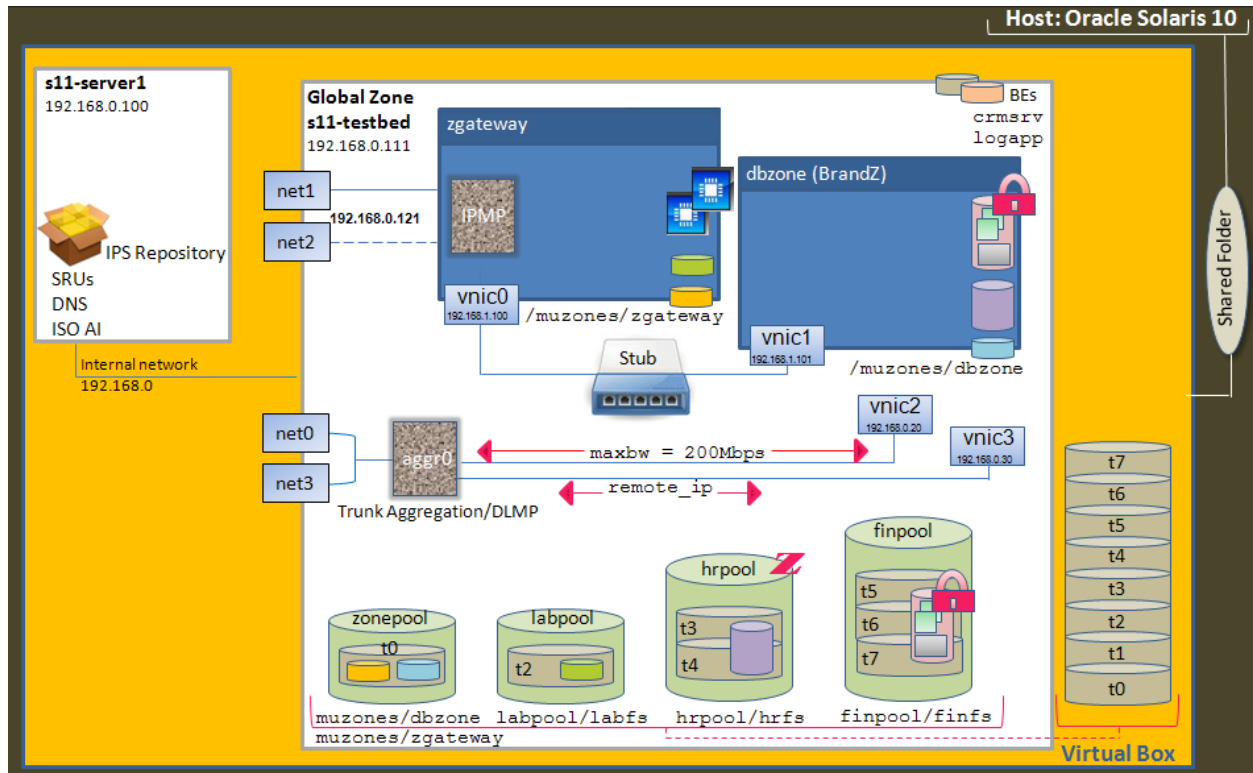
Practices for Lesson 6: Administering: Oracle Solaris Zones

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

Workload separation is one of the key benefits of Oracle Solaris Zones. For the testbed environment, you need to configure two non-global zones: `zgateway` as a whole root zone and `dbzone` as a branded zone. You then need to provision these zones with required file systems, datasets, and network interfaces, both physical and virtual. In addition, you need to provide for network high availability so that these zones meet Murraya's business continuity plan. Finally, you need to ensure that resources on these zones are optimally distributed and used.



In this practice, you perform the following tasks:

- Configure `zgateway` as a whole root zone.
- Configure `dbzone` as a branded zone.
- Configure IPMP on the `zgateway` zone.
- Delegate ZFS datasets to zones.
- Allocate resources to the zones.

Notes

- Command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.


```

        /pci@0,0/pci8086,2829@d/disk@2,0
2. c7t3d0 <ATA-VBOX HARDDISK-1.0-1.00GB>
        /pci@0,0/pci8086,2829@d/disk@3,0
3. c7t4d0 <ATA-VBOX HARDDISK-1.0-1.00GB>
        /pci@0,0/pci8086,2829@d/disk@4,0
4. c7t5d0 <ATA-VBOX HARDDISK-1.0-5.00GB>
        /pci@0,0/pci8086,2829@d/disk@5,0
5. c7t6d0 <ATA-VBOX HARDDISK-1.0-5.00GB>
        /pci@0,0/pci8086,2829@d/disk@6,0
6. c7t7d0 <ATA-VBOX HARDDISK-1.0-5.00GB>
        /pci@0,0/pci8086,2829@d/disk@7,0
7. c7t8d0 <ATA-VBOX HARDDISK-1.0-20.00GB>
        /pci@0,0/pci8086,2829@d/disk@8,0
8. c7t9d0 <ATA-VBOX HARDDISK-1.0-40.00GB>
        /pci@0,0/pci8086,2829@d/disk@9,0
Specify disk (enter its number): ^C
root@s11-testbed:~#

```

4. Run the `zpool status` command to determine which disks are currently configured in the ZFS pools.

```

root@s11-testbed:~# zpool status

pool: finpool
state: ONLINE
scan: none requested
config:

    NAME            STATE        READ  WRITE CKSUM
    finpool          ONLINE       0     0     0
    raidz1-0         ONLINE       0     0     0
    c7t5d0            ONLINE       0     0     0
    c7t6d0            ONLINE       0     0     0
    c7t7d0            ONLINE       0     0     0

errors: No known data errors


pool: hrpool
state: ONLINE
scan: none requested
config:

    NAME            STATE        READ  WRITE CKSUM
    hrpool           ONLINE       0     0     0
    mirror-0         ONLINE       0     0     0
    c7t3d0            ONLINE       0     0     0
    c7t4d0            ONLINE       0     0     0

errors: No known data errors


pool: labpool

```

```

state: ONLINE
  scan: none requested
config:

    NAME      STATE      READ  WRITE CKSUM
    labpool   ONLINE        0     0     0
    c7t2d0    ONLINE        0     0     0

errors: No known data errors

pool: rpool
state: ONLINE
  scan: none requested
config:

    NAME      STATE      READ  WRITE CKSUM
    rpool     ONLINE        0     0     0
    c7t0d0    ONLINE        0     0     0

errors: No known data errors
root@s11-testbed:~#

```

5. Use the c7t8d0 disk to create a ZFS pool named zonepool.

```
root@s11-testbed:~# zpool create zonepool c7t8d0
```

Note: Recommended pool configuration for a production system is a redundant pool like mirror or raidz.

6. Use the `zpool list` command to verify zonepool creation.

```

root@s11-testbed:~# zpool list zonepool
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
zonepool  19.9G   85K  19.9G   0%  1.00x  ONLINE  -
root@s11-testbed:~#

```

7. Create a ZFS file system named zonepool/muzones and a mount point as /muzones.

```
root@s11-testbed:~# zfs create -o mountpoint=/muzones zonepool/muzones
```

Verify that the zonepool/muzones ZFS file system is mounted as /muzones.

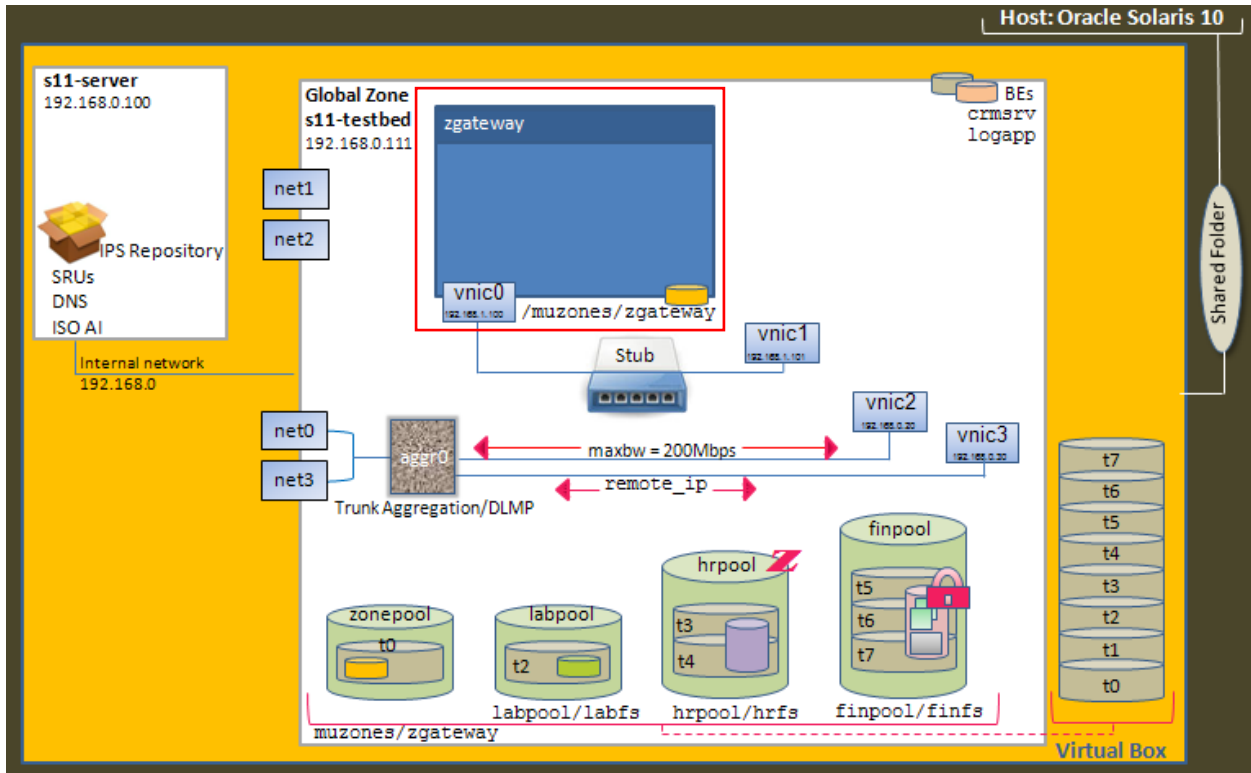
```

root@s11-testbed:~# zfs list zonepool/muzones
NAME                USED   AVAIL  REFER  MOUNTPOINT
zonepool/muzones    31K    19.6G   31K    /muzones

```

The `-o` option sets the mount point and automatically creates the necessary directory.

Task 2: Configure the zgateway Zone



Perform the following steps to configure the zgateway zone:

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.
3. Verify that the Image Packaging System (IPS) publisher is configured correctly and is operational.

```
root@s11-testbed:~# pkg publisher
PUBLISHER          TYPE      STATUS P  LOCATION
solaris             origin   online F  http://s11-server1.mydomain.com/
root@s11-testbed:~# ping s11-server1
s11-server1 is alive
root@s11-testbed:~#
```

4. In an earlier practice, you created two vnics (`vnic0` and `vnic1`) on an etherstub. Verify their status.

```
root@s11-testbed:~# dladm show-vnic
LINK          OVER      SPEED  MACADDRESS      MACADDRTYPE  VID
vnic0         stub0     40000  2:8:20:7c:64:b5  random       0
vnic1         stub0     40000  2:8:20:42:fd:37  random       0
vnic2         aggr0     600    2:8:20:b0:20:17  random       0
vnic3         aggr0     400    2:8:20:65:e7:f   random       0
root@s11-testbed:~#
```

5. Configure the zgateway zone and display the results.

```

root@s11-testbed:~# zonecfg -z zgateway
Use 'create' to begin configuring a new zone
zonecfg:zgateway> create
create: Using system default template 'SYSdefault'
zonecfg:zgateway> set zonepath=/muzones/zgateway
zonecfg:zgateway> set autoboot=true
zonecfg:zgateway> add net
zonecfg:zgateway:net> set physical=vnic0
zonecfg:zgateway:net> end
zonecfg:zgateway> add net
zonecfg:zgateway:net> set physical=net1
zonecfg:zgateway:net> set configure-allowed-address=false
zonecfg:zgateway:net> end
zonecfg:zgateway> add net
zonecfg:zgateway:net> set physical=net2
zonecfg:zgateway:net> set configure-allowed-address=false
zonecfg:zgateway:net> end
zonecfg:zgateway> remove anet
zonecfg:zgateway> verify
zonecfg:zgateway> commit
zonecfg:zgateway> exit

root@s11-testbed:~# zonecfg -z zgateway info
zonename: zgateway
zonepath: /muzones/zgateway
brand: solaris
autoboot: true
bootargs:
file-mac-profile:
pool:
limitpriv:
scheduling-class:
ip-type: exclusive
hostid:
fs-allowed:
net:
    address not specified
    allowed-address not specified
    configure-allowed-address: true
    physical: vnic0
    defrouter not specified
net:
    address not specified
    allowed-address not specified
    configure-allowed-address: false

```

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

```

    physical: net1
    defrouter not specified
net:
    address not specified
    allowed-address not specified
    configure-allowed-address: false
    physical: net2
    defrouter not specified
root@s11-testbed:~#

```

The two network interfaces, net1 and net2, are assigned to zgateway. These interfaces will be required for configuring IP Multipathing (IPMP).

6. Display the configured zone by using the zoneadm command.

```

root@s11-testbed:~# zoneadm list -cv

```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	zgateway	configured	/muzones/zgateway	solaris	excl

The zgateway zone is in the configured state. It now needs to be installed.

7. Install the zgateway zone.

```

root@s11-testbed:/# zoneadm -z zgateway install
The following ZFS file system(s) have been created:
    zonepool/muzones/zgateway
Progress being logged to
/var/log/zones/zoneadm.20130916T013335Z.zgateway.install
Image: Preparing at /muzones/zgateway/root.

AI Manifest: /tmp/manifest.xml.qLayEg
SC Profile: /usr/share/auto_install/sc_profiles/enable_sci.xml
Zonename: zgateway
Installation: Starting ...

    Creating IPS image
Startup linked: 1/1 done
Installing packages from:
    solaris
        origin: http://s11-server1.mydomain.com/
DOWNLOAD                                PKGS          FILES      XFER (MB)   SPEED
Completed                             183/183      33556/33556  222.2/222.2  2.6M/s

PHASE                                ITEMS
Installing new actions                46825/46825
Updating package state database                    Done
Updating image state                          Done
Creating fast lookup database                    Done
Installation: Succeeded

```

Note: Man pages can be obtained by installing `pkg:/system/manual`

done.

Done: Installation completed in 234.781 seconds.

Next Steps: Boot the zone, then log into the zone console (`zlogin -C`)

to complete the configuration process.

Log saved in non-global zone as
`/muzones/zgateway/root/var/log/zones/zoneadm.20130916T013335Z.zgateway.install`
`root@s11-testbed:/#`

Note: This step normally takes several minutes to complete.

8. Check the status of the zgateway zone.

```
root@s11-testbed:/# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
-	zgateway	installed	/muzones/zgateway	solaris	excl

The zgateway zone is in installed state.

9. Boot the zgateway zone and display the result.

```
root@s11-testbed:/# zoneadm -z zgateway boot
```

```
root@s11-testbed:/# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
2	zgateway	running	/muzones/zgateway	solaris	excl

The zgateway zone is in running state.

10. Log in to zgateway and complete the system configuration by using the following parameters.

- Computer name: `zgateway`
- Ethernet network configuration: Manually
- Network Interface: `vnic0` (Press Tab to see this option.)
- IP Address: `192.168.1.100`
- DNS Name Service: Do not configure DNS. (Press Tab to see this option.)
- Alternate Name Service: None
- Time zone: Use your local region.
- Date and time: Set to the current date and time.
- Root password: `oracle1`
- User account:
- Your real name: `oracle`

- Username: oracle
- Password: oracle1

```
root@s11-testbed:~# zlogin -C zgateway
[Connected to zone 'zgateway' console]
```

Note: Wait for a few minutes before the `sysconfig` utility tool is displayed. If the tool is not displayed automatically, press Enter. In the tool, if the F2 and F3 keys do not work, press Esc + 2 to navigate through the screens and press Esc + 3 to go back. If the up and down arrow keys on the keyboard do not work, use the Tab key. However, you need to be careful while using Tab. The options on the screen might look confusing (appearing twice). You should therefore read the options carefully as you press Tab. When the system configuration is completed, log in to `zgateway` with the username `oracle` and password `oracle1`. Use the `~.` escape sequence to exit to the global zone.

11. Check the virtual network configuration in the `zgateway` zone.

```
root@s11-testbed:~# zlogin zgateway
[Connected to zone 'zgateway' pts/3]
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@zgateway:~# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4        static    ok         127.0.0.1/8
vnic0/v4      static    ok         192.168.1.100/24
lo0/v6        static    ok         ::1/128
vnic0/v6      addrconf  ok         fe80::8:20ff:fe17:c42a/10
root@zgateway:~# exit
```

Summary: You have successfully configured the `zgateway` zone.


```

Category: System/Virtualization
State: Installed
Publisher: solaris
Version: 0.5.11
Build Release: 5.11
Branch: 0.175.1.0.0.24.2
Packaging Date: September 19, 2012 06:53:02 PM
Size: 804.26 kB
FMRI: pkg://solaris/system/zones/brand/brand-solaris10@0.5.11,5.11-
0.175.1.0.0.24.2:20120919T185302Z
root@s11-testbed:~#

```

If the brand-solaris10 package is not installed, use the `pkg install` command to install it.

```

root@s11-testbed:~# pkg install brand-solaris10
root@s11-testbed:~# pkg info brand-solaris10

```

Task 2: Create dbzone by Using the solaris10 Brand

1. Configure dbzone and display the results.

```

root@s11-testbed:~# zonecfg -z dbzone
Use 'create' to begin configuring a new zone.
zonecfg:dbzone> create -t SYSSolaris10
zonecfg:dbzone> set zonepath=/muzones/dbzone
zonecfg:dbzone> set autoboot=true
zonecfg:dbzone> add net
zonecfg:dbzone:net> set physical=vnic1
zonecfg:dbzone:net> end
zonecfg:dbzone> remove anet
zonecfg:dbzone> verify
zonecfg:dbzone> commit
zonecfg:dbzone> exit
root@s11-testbed:~#

```

2. Verify the dbzone configuration by using the zonecfg command.

```

root@s11-testbed:~# zonecfg -z dbzone info
zonename: dbzone
zonepath: /muzones/dbzone
brand: solaris10
autoboot: true
bootargs:
pool:
limitpriv:
scheduling-class:
ip-type: exclusive
hostid:
fs-allowed:
net:
    address not specified

```

```

    allowed-address not specified
    configure-allowed-address: true
    physical: vnic1
    defrouter not specified
root@s11-testbed:~#

```

3. Verify the dbzone configuration by using the zoneadm command.

```

root@s11-testbed:~# zoneadm list -cv

```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
2	zgateway	running	/muzones/zgateway	solaris	excl
-	dbzone	configured	/muzones/dbzone	solaris10	excl

```

root@s11-testbed:~#

```

4. Verify that the s10coregroup.flar image file is located in the /opt/ora/lab directory.

```

root@s11-testbed:~# ls /opt/ora/lab/s10coregroup.flar
/opt/ora/s10coregroup.flar

```

5. Install dbzone by using a minimal Oracle Solaris 10 flash archive created on a physical machine and provided for the course.

```

root@s11-testbed:~# zoneadm -z dbzone install -a
/opt/ora/lab/s10coregroup.flar -u
The following ZFS file system(s) have been created:
    zonepool/muzones/dbzone
Progress being logged to
/var/log/zones/zoneadm.20131106T051131Z.dbzone.install
    Installing: This may take several minutes...
        WARNING: Skipping image sanity checks.
Postprocessing: This may take a while...
    Postprocess: ERROR: Invalid '/etc/zones/index' file within the zone
    Postprocess: Updating the image to run within a zone

    Result: Installation completed successfully.
Log saved in non-global zone as
/muzones/dbzone/root/var/log/zones/zoneadm.20131106T051131Z.dbzone.install
root@s11-testbed:~#

```

Note: Ignore the following message and proceed to the next step.

```

WARNING: Skipping image sanity checks.
Postprocess: ERROR: Invalid '/etc/zones/index' file within the zone

```

6. Verify the zone installation.

```
root@s11-testbed:~# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
2	zgateway	running	/muzones/zgateway	solaris	excl
-	dbzone	installed	/muzones/dbzone	solaris10	excl

```
root@s11-testbed:~#
```

7. Boot the zone and display the result.

```
root@s11-testbed:~# zoneadm -z dbzone boot
```

Note: Ignore the warning message and proceed.

zone 'dbzone': Warning: "/usr/lib/netshvc/rstat/rpc.rstatd" is not installed in the global zone

```
root@s11-testbed:~# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
2	zgateway	running	/muzones/zgateway	solaris	excl
5	dbzone	running	/muzones/dbzone	solaris10	excl

```
root@s11-testbed:~#
```

8. Log in to dbzone and complete the system configuration by using the following parameters.

```
root@s11-testbed:~# zlogin -C dbzone
```

[Connected to zone 'dbzone' console]

You did not enter a selection.

What type of terminal are you using?

- 1) ANSI Standard CRT
- 2) DEC VT100
- 3) PC Console
- 4) Sun Command Tool
- 5) Sun Workstation
- 6) X Terminal Emulator (xterms)
- 7) Other

Type the number of your choice and press Return: 6

Note: Ensure that you get your terminal type correct. X Terminal Emulator (xterms) seems to work better in a VirtualBox environment.

Wait for a few minutes before the `sysconfig` utility tool is displayed. If the tool is not displayed automatically, press Enter. In the tool, if the F2 and F3 keys do not work, press Esc + 2 to navigate through the screens and press Esc + 3 to go back. If the up and down arrow keys on the keyboard do not work, use the Tab key. However, you need to be careful while using Tab. The options on the screen might look confusing (appearing twice). You should therefore read the options carefully as you press Tab.

- Host name: dbzone

- IP address for `vnic1`: 192.168.1.101
- System part of a subnet: Yes
- Netmask for `vnic1`: 255.255.255.0
- Enable IPV6 for `vnic1`: No
- Default Route for `vnic1`: None
- Configure Kerberos Security: No
- Name service: None
- NFSv4 Domain Configuration: Use the NFSv4 domain derived by the system.
- Time zone: Use your local region.
- Root password: `oracle1`

When the system configuration of `dbzone` is complete, the console login prompt appears.

9. Log in to the console login with the password `oracle1`.

```
dbzone console login: root
Password: oracle1
Last login: Wed Jan  4 12:26:07 on console
Nov  6 10:56:36 dbzone login: ROOT LOGIN /dev/console
Oracle Corporation      SunOS 5.10      Generic Patch    January 2005
#
```

10. Display the hostname `dbzone`.

```
#hostname
dbzone
```

11. Verify that the Oracle Solaris operating system is installed on `dbzone`.

```
# cat /etc/release

                Oracle Solaris 10 8/11 s10x_u10wos_17b X86

    Copyright (c) 1983, 2011, Oracle and/or its affiliates. All rights
    reserved.

                Assembled 23 August 2011

#
```

12. Determine the `dbzone`'s network interface and IP configuration.

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu
8232 index 1
        inet 127.0.0.1 netmask ff000000
vnic1: flags=1000863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,IPv4>
mtu 9000 index 2
        inet 192.168.1.101 netmask ffffffff00 broadcast 192.168.1.255
        ether 2:8:20:42:fd:37

#
```

13. Execute the `zpool list` command to display the ZFS pools that are currently configured in `dbzone`.

```
# zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
rpool    19.9G  1.00G  18.9G   5%  1.00x  ONLINE  -

# zpool status
pool: rpool
state: ONLINE
scan: none requested
config:

          NAME      STATE      READ WRITE CKSUM
          rpool     ONLINE         0     0     0
              c7t8d0 ONLINE         0     0     0

errors: No known data errors
#
```

14. Verify that `dbzone` can connect with the `zgateway` zone by using the `ping` command.

```
# ping 192.168.1.100
192.168.1.100 is alive
#

Disconnect from the zone console by using the ~. or ~~. key combinations.

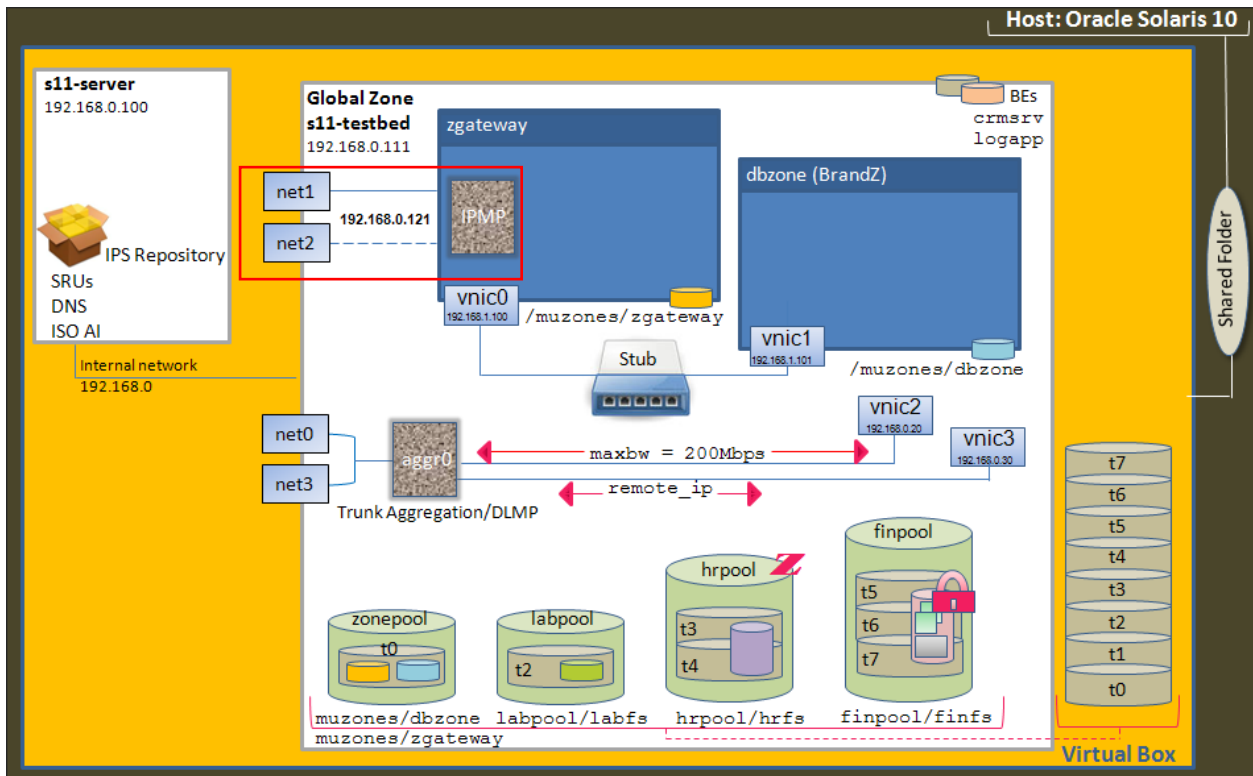
# ~.
[Connection to zone 'dbzone' console closed]
root@s11-testbed:~#
```

Summary: The `dbzone` branded zone is configured and can be used for archiving or migration of any Oracle Solaris 10 data in an Oracle Solaris 11 system.

Practice 6-3: Configuring IPMP for Network High Availability

Overview

The `zgateway` zone is the entry zone in the testbed environment. This means that if there is network failure on the primary interface that is connected with the `zgateway` zone, all zones in the internal network lose network connectivity with the external network. It is therefore critical to configure a redundant interface so that network continuity is ensured in the event of a failure of the primary interface.



In this practice, you reconfigure the `zgateway` zone to implement HA by using IPMP.

Tasks

To configure an IPMP group, you require two interfaces: `net1` and `net2`.

1. Verify that the `s11-testbed` VM is running.
2. Log in to the `s11-testbed` VM as the `oracle` user with the password, `oracle1`.
3. Switch to the `root` role by using the `su -` command.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation      SunOS 5.11      11.1      September 2012
root@s11-testbed:~#
```


4. Verify that the zgateway zone is in running state.

```
root@s11-testbed:~# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
5	dbzone	running	/muzones/dbzone	solaris10	excl
1	zgateway	running	/muzones/zgateway	solaris	excl

5. If required, run the following commands to start the zgateway zone and log in to it.

```
root@s11-testbed:~# zoneadm -z zgateway boot
root@s11-testbed:~# zlogin zgateway
[Connected to zone 'zgateway' pts/2]

Oracle Corporation      SunOS 5.11      11.1      September 2012
root@zgateway:~#
```

6. Identify the net devices to be used for configuring IPMP. Here you use net1 and net2, which map to e1000g1 and e1000g2, respectively.

```
root@zgateway:~# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net1	Ethernet	unknown	0	unknown	e1000g1
net2	Ethernet	unknown	1000	unknown	e1000g2

7. Display the address information of the interface by using the ipadm command.

```
root@zgateway:~# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
vnic0/v4	static	ok	192.168.1.100/24
lo0/v6	static	ok	:::1/128
vnic0/v6	addrconf	ok	fe80::8:20ff:fe17:c42a/10

8. Assign an IP to configure the two interfaces, net1 and net2.

```
root@zgateway:~# ipadm create-ip net1
root@zgateway:~# ipadm create-ip net2
```

9. Create an IPMP group named ipmp0 by using the net1 and net2 interfaces.

```
root@zgateway:~# ipadm create-ipmp -i net1 -i net2 ipmp0
root@zgateway:~# ipadm create-addr -T static -a 192.168.0.121 ipmp0/v4
```

10. Verify that ipmp0 has been configured.

```
root@zgateway:~# ipadm show-addr
```

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
vnic0/v4	static	ok	192.168.1.100/24
ipmp0/v4	static	ok	192.168.0.121/24
lo0/v6	static	ok	:::1/128
vnic0/v6	addrconf	ok	fe80::8:20ff:fe17:c42a/10

Data address has been plumbed on the IPMP group.

11. Display the group-wise IPMP subsystem status by using the `ipmpstat -g` command.

```
root@zgateway:~# ipmpstat -g
GROUP          GROUPNAME    STATE      FDT          INTERFACES
ipmp0          ipmp0        ok         --           net2 net1
root@zgateway:~#
```

12. Display the interface information about the IPMP group.

```
root@zgateway:~# ipmpstat -i
INTERFACE      ACTIVE  GROUP      FLAGS        LINK        PROBE        STATE
net2           yes    ipmp0      -----      up          disabled     ok
net1           yes    ipmp0      --mbM--      up          disabled     ok
```

where;

- m indicates that the interface is designated for sending and receiving IPv4 multicast traffic for the IPMP group.
- b indicates that the interface is designated for receiving broadcast traffic for the IPMP group.
- M indicates that the interface is designated for sending and receiving IPv6 multicast traffic for the IPMP group.

13. On the **s11-server1** VM, confirm that the IPMP group is active by pinging 192.168.0.121.

```
oracle@s11-server1:~$ ping 192.168.0.121
192.168.10.30 is alive
Are you able to ping the IP address assigned to the IPMP group? Yes
```

14. On the **s11-server1** VM, try to ping the IP addresses of `zgateway` and `dbzone` zones.

```
oracle@s11-server1:~$ ping 192.168.1.100
ping: sendto No route to host
oracle@s11-server1:~$ ping 192.168.1.101
ping: sendto No route to host

Are you able to ping? No
Why? IP forwarding is turned off in the zgateway zone.
```

15. Log in to `zgateway` and display the `ipv4` properties of the `zgateway` zone.

```
root@zgateway:~# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY      PERM CURRENT    PERSISTENT  DEFAULT    POSSIBLE
ipv4  forwarding      rw   off           --           off        on,off

ipv4 forwarding is turned off.
```

16. Enable IP forwarding and display the result.

```
root@zgateway:~# ipadm set-prop -p forwarding=on ipv4
root@zgateway:~# ipadm show-prop -p forwarding ipv4
PROTO PROPERTY      PERM CURRENT    PERSISTENT  DEFAULT    POSSIBLE
ipv4  forwarding      rw   on            on          off        on,off
root@zgateway:~# exit
```

17. From s11-server1, ping the IP addresses of the `zgateway` and `dbzone` zones.

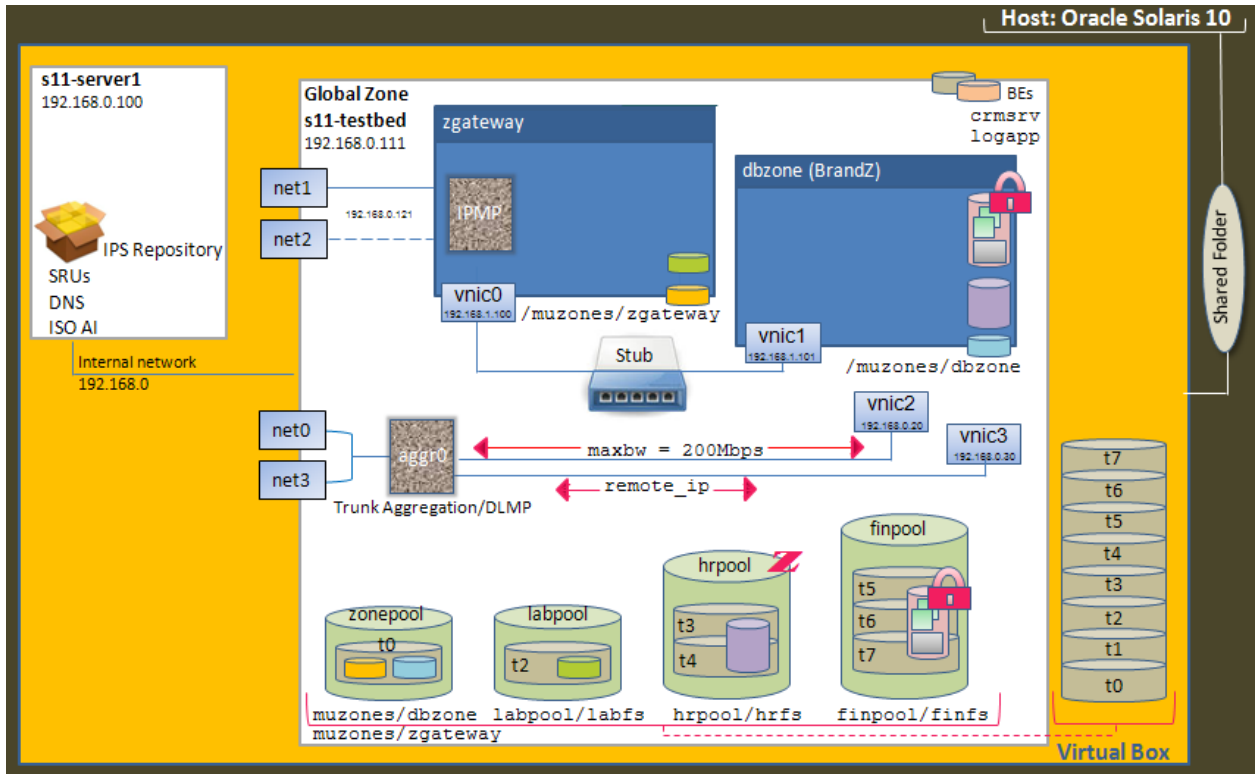
```
oracle@s11-server1:~$ ping 192.168.1.100
192.168.1.100 is alive
oracle@s11-server1:~$ ping 192.168.1.101
192.168.1.101 is alive
```

Summary: Both the `zgateway` zone and the `dbzone` zone are reachable from `s11-server1`. By configuring IPMP on the `zgateway` zone, you have ensured that the alternative interface (`net2`) becomes active if the primary network interface (`net1`) fails.

Practice 6-4: Delegating ZFS Datasets to Zones

Overview

In an earlier practice, you created datasets (labpool/labfs, hrpool/hrfs, and finpool/finfs) in the global zone. You now delegate those datasets to the non-global zones zgateway and dbzone.



In this practice, you perform the following tasks:

- Delegate the labpool/labfs dataset to zgateway.
- Delegate the hrpool/hrfs and finpool/finfs datasets to dbzone.

Task 1: Delegate the labpool/labfs Dataset to the zgateway Zone

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the **oracle** user. Use **oracle1** as the password. Assume administrator privileges.

3. Delegate the labpool/labfs dataset to the zgateway zone by using the zonecfg command.

```
root@s11-testbed:~# zonecfg -z zgateway
zonecfg:zgateway> add dataset
zonecfg:zgateway:dataset> set name=labpool/labfs
zonecfg:zgateway:dataset> end
zonecfg:zgateway> verify
zonecfg:zgateway> commit
zonecfg:zgateway> exit
root@s11-testbed:~#
```

4. Reboot the zgateway zone by using the zoneadm command.

```
root@s11-testbed:~# zoneadm -z zgateway reboot
```

Note: Reboot might take few seconds to few minutes.

5. Log in to the zgateway zone and display the results.

```
root@s11-testbed:~# zlogin zgateway
[Connected to zone 'zgateway' pts/4]
Oracle Corporation      SunOS 5.11  11.1   September 2012
root@zgateway:~# zpool list
```

NAME	SIZE	ALLOC	FREE	CAP	DEDUP	HEALTH	ALTROOT
labfs	1008M	212K	1008M	0%	1.00x	ONLINE	-
rpool	19.9G	1.01G	18.9G	5%	1.00x	ONLINE	-

```
root@zgateway:~#
```

6. Try to create the labfs/labfs1 file system and display the result.

```
root@zgateway:~# zfs create labfs/labfs1
root@zgateway:~# zfs list -r labfs
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
labfs	84K	976M	34K	/labfs
labfs@snap1	19K	-	32K	-
labfs/labfs1	31K	976M	31K	/labfs/labfs1

```
root@zgateway:~# exit
```

7. From s11-testbed, display the zoned property for labpool file systems.

```
root@s11-testbed:~# zfs get -r zoned labpool
```

NAME	PROPERTY	VALUE	SOURCE
labpool	zoned	off	default
labpool@snap1	zoned	-	-
labpool/labfs	zoned	on	local
labpool/labfs@snap1	zoned	-	-
labpool/labfs/labfs1	zoned	on	inherited from labpool/labfs

```
root@s11-testbed:~#
```

Note: The labpool/labfs file system was delegated to a zgateway zone and mounted in the zone, and it is under zone administrator control. ZFS uses the zoned property to indicate that a dataset has been delegated to a zgateway zone.

Summary: ZFS file systems are created in the global zone and assigned to non-global zones. Creating the labfs/labfs1 file system illustrates that you can create a file system after assigning the dataset to a non-global zone; in the non-global zone, the delegated dataset shows up as a pool.

Task 2: Delegate the hrpool/hrfs Dataset to the dbzone Zone

Just as you delegated the labpool/labfs dataset to the zgateway zone, you need to delegate the hrpool/hrfs and finpool/finfs datasets to dbzone.

1. Use the zonecfg command to delegate the hrpool/hrfs file system to the dbzone zone.

```
root@s11-testbed:~# zonecfg -z dbzone
zonecfg:dbzone> add dataset
zonecfg:dbzone:dataset> set name=hrpool/hrfs
zonecfg:dbzone:dataset> end
zonecfg:dbzone> verify
zonecfg:dbzone> commit
zonecfg:dbzone> exit
```

2. Log in to the dbzone zone and list the ZFS pools.

Note: If you observe the following message, you can ignore it.

Couldn't set locale correctly

```
root@s11-testbed:~# zlogin dbzone
[Connected to zone 'dbzone' pts/2]
Oracle Corporation      SunOS 5.10  Generic Patch      January 2005
# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
rpool    19.9G  1.01G  18.9G   5%  1.00x  ONLINE  -
# exit

Are you able to see hrpool/hrfs? No
Why? dbzone is not rebooted.
```

3. Reboot dbzone by using the zoneadm command.

```
root@s11-testbed:~# zoneadm -z dbzone reboot
```

Note: Ignore the warning message and proceed to the next step.

```
zone 'dbzone': Warning: "/usr/lib/netsvc/rstat/rpc.rstatd" is not
installed in the global zone
```

4. Log in to dbzone and list the ZFS pools.

```
root@s11-testbed:~# zlogin dbzone
[Connected to zone 'dbzone' pts/2]
Oracle Corporation      SunOS 5.10  Generic Patch      January 2005
#
# zpool list
NAME      SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
hrfs      1008M  435K  1008M   0%  1.00x  ONLINE  -
```

```
hrpool    15.9G    932M    15.0G    5%    1.00x    ONLINE    -
```

Are you now able to see hrpool? Yes

5. Try to create the hrfs/hrfs1 file system and display the result. When done, exit from dbzone.

```
# zfs create hrfs/hrfs1
# zfs list -r hrfs
NAME                                USED  AVAIL  REFER  MOUNTPOINT
hrfs                                63K   976M   32K    /hrfs
hrfs/hrfs1                          31K   976M   31K    /hrfs/hrfs1
# exit
```

6. From s11-testbed, display the zoned property for hrpool file systems.

```
root@s11-testbed:~# zfs get -r zoned hrpool
NAME                                PROPERTY  VALUE    SOURCE
hrpool                             zoned     off      default
hrpool/hrfs                        zoned     on       local
hrpool/hrfs/hrfs1                 zoned     on       inherited from hrpool/hrfs
root@s11-testbed:~#
```

The hrpool/hrfs file system was delegated from s11-testbed to a dbzone.

Summary: The hrpool/hrfs file system was created in the global zone and is now assigned to a non-global zone. Creating the hrfs/hrfs1 file system illustrates that you can create a file system even after assigning the dataset to a non-global zone.

Task 3: Delegate finpool/finfs Dataset to the dbzone Zone

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the oracle user. Use oracle1 as the password. Assume administrator privileges.
3. Delegate the finpool/finfs file system to the dbzone zone by using the zonecfg command.

```
root@s11-testbed:~# zonecfg -z dbzone
zonecfg:dbzone> add dataset
zonecfg:dbzone:dataset> set name=finpool/finfs
zonecfg:dbzone:dataset> end
zonecfg:dbzone> verify
zonecfg:dbzone> commit
zonecfg:dbzone> exit
root@s11-testbed:~#
```

4. Reboot the dbzone zone by using the zoneadm command.

```
# zoneadm -z dbzone reboot
```

Note: Ignore the warning message and proceed to next step.

5. Log in to the dbzone zone and display the result.

```

root@s11-testbed:~# zlogin dbzone
[Connected to zone 'dbzone' pts/2]
Oracle Corporation      SunOS 5.10  Generic Patch      January 2005
# zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
finfs    14.9G   715K  14.9G   0%  1.00x  ONLINE  -
hrfs     1008M   476K  1008M   0%  1.00x  ONLINE  -
rpool    19.9G  1.03G  18.8G   5%  1.00x  ONLINE  -
#

```

6. From s11-testbed, display the zoned property for finpool file systems.

```

root@s11-testbed:~# zfs get -r zoned finpool
NAME                                     PROPERTY  VALUE  SOURCE
finpool                                zoned    off    default
finpool/finfs                          zoned    on     local
finpool/finfs/ar                       zoned    on     inherited from
finpool/finfs/ar@saturday              zoned    -      -
finpool/finfs/ar/fridayclone           zoned    on     inherited from
finpool/finfs/ar/fridayclone@friday    zoned    -      -
finpool/finfs/del                      zoned    on     inherited from
finpool/finfs/inv                      zoned    on     inherited from
root@s11-testbed:~#

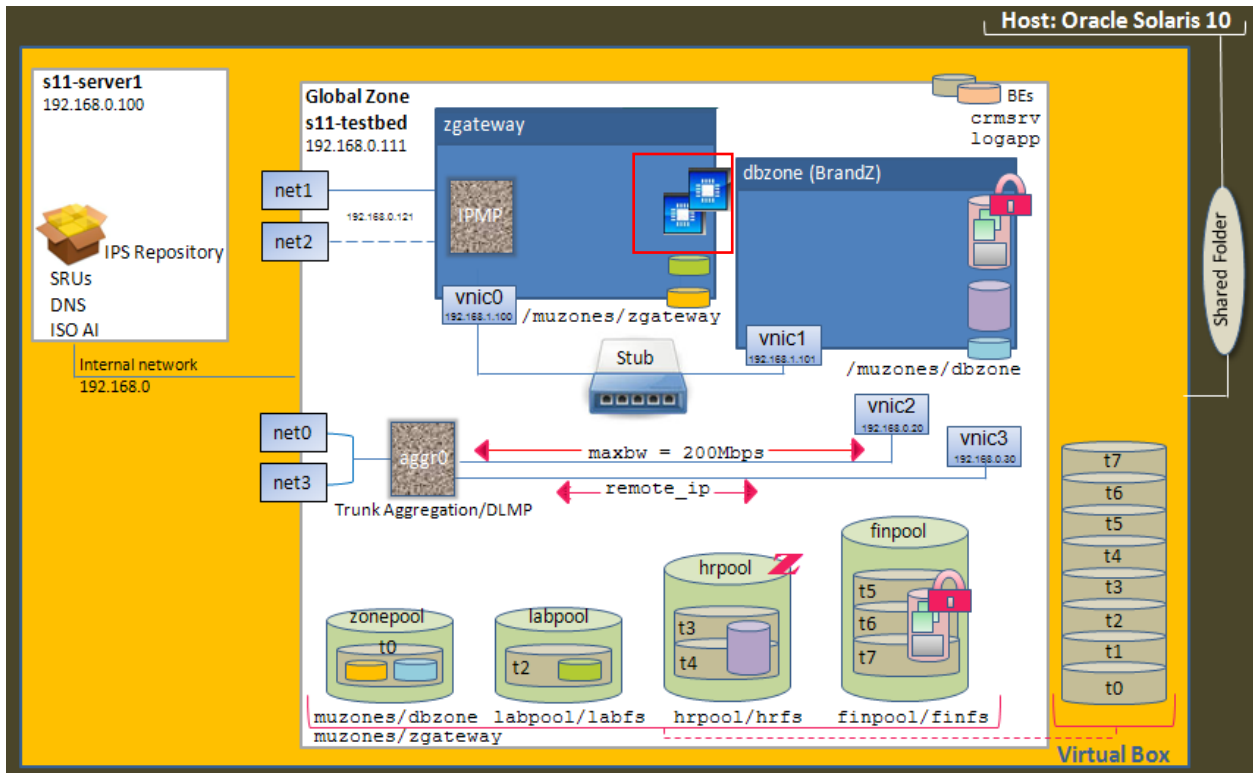
```

7. Exit the dbzone zone.

Practice 6-5: Allocating Resources to Zones

Overview

You have configured the `zgateway` and `dbzone` zones with basic resources. You can now further provision the zones by allocating additional resources. Because of the types of activities for which `zgateway` is being designated, it should have higher processing capabilities. In this practice, you configure a pool of two CPUs and allocate the pool to the `zgateway` zone.



In this practice, you perform the following tasks:

- Enable resource pool services.
- Configure a persistent resource pool.
- Bind the `zgateway` zone to the persistent resource pool.

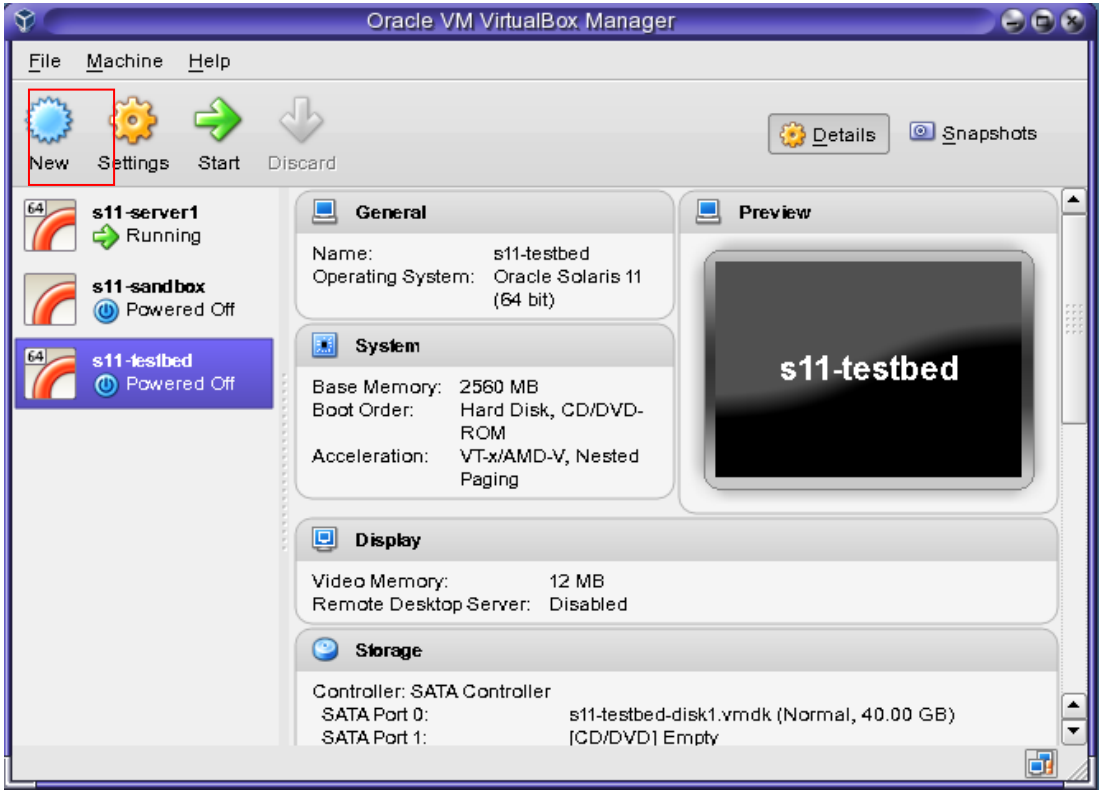
Prerequisite

This activity requires the `s11-testbed` VM to have two CPUs so that resource pools can be configured accordingly. To ensure that the `s11-testbed` VM has two CPUs, perform the following steps:

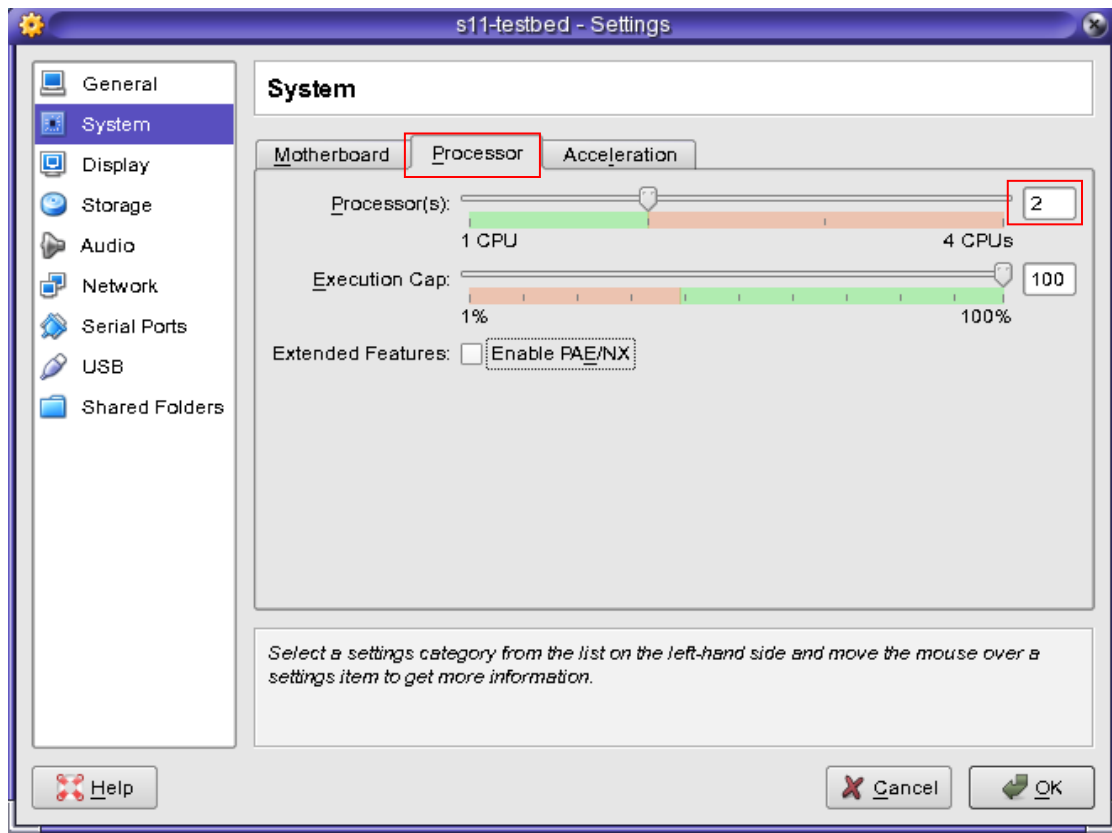
1. Shut down the `s11-testbed` VM.



- Open the VirtualBox Manager GUI and click the **Settings** utility for the **s11-testbed** VM.



- Under the System settings, click the **Processor** tab and verify that the number of processors is 2. If not, change the number of processors to 2. Click **OK** to continue and start the VM.



Task 1: Enable Resource Pool Services

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.
3. Verify that the `pool` daemon and the `pool` services are running.

```
root@s11-testbed:~# pgrep -lf poold
root@s11-testbed:~# svcs *pools*
STATE          STIME          FMRI
disabled        16:06:10      svc:/system/pools:default
disabled        16:05:55      svc:/system/pools/dynamic:default
```

Currently, all the pool services are disabled.

4. Verify that the `dynamic` service is dependent on the `default` pool service.

```
root@s11-testbed:~# svcs -d pools/dynamic
STATE          STIME          FMRI
disabled        16:06:10      svc:/system/pools:default
online          15:45:55      svc:/system/filesystem/local:default
```

5. Enable the pool services recursively by using the `svcadm` command.

```
root@s11-testbed:~# svcadm enable -r pools/dynamic
root@s11-testbed:~# svcs *pools*
STATE          STIME          FMRI
online          16:08:10      svc:/system/pools:default
online          16:08:11      svc:/system/pools/dynamic:default
root@s11-testbed:~# pgrep -lf poold
8493 /usr/lib/pool/poold
```

The pool services and the `poold` daemon are now up.

6. Display the default resource pool configuration that is currently in use by using the `pooladm` command.

```
root@s11-testbed:~# pooladm

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

    pool pool_default
        int     pool.sys_id 0
        boolean pool.active true
        boolean pool.default true
        int     pool.importance 1
        string  pool.comment
```

```

pset      pset_default

pset pset_default
    int      pset.sys_id -1
    boolean  pset.default true
    uint     pset.min 1
    uint     pset.max 65536
    string   pset.units population
    uint     pset.load 64
    uint     pset.size 2
    string   pset.comment

    cpu

        int      cpu.sys_id 1
        string   cpu.comment
        string   cpu.status on-line

    cpu

        int      cpu.sys_id 0
        string   cpu.comment
        string   cpu.status on-line

root@s11-testbed:~#

```

Examine the default pool and the `pset` (processer set) configuration. Also note the number of available CPUs.

Task 2: Configure a Persistent Resource Pool

1. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.
2. Create the pool configuration file.

```

root@s11-testbed:~# ls -l /etc/pool*
/etc/pool*: No such file or directory

```

Currently, the `pooladm.conf` file does not exist.

```

root@s11-testbed:~# pooladm -s

```

You save the current pool configuration in the default `/etc/pooladm.conf` file.

```

root@s11-testbed:~# ls -l /etc/pool*
-rw-r--r-- 1 root root 1160 Dec 14 16:13 /etc/pooladm.conf
root@s11-testbed:~# file /etc/pooladm.conf
/etc/pooladm.conf:      XML document

```

The XML file has been created for you.

- Display the content of the pool configuration file by using the `more` command so that you can examine its content one page at a time.

```

root@s11-testbed:~# more /etc/pooladm.conf
<?xml version="1.0"?>
<!DOCTYPE system PUBLIC "-//Sun Microsystems Inc//DTD Resource
Management All//EN" "file:///usr/share/lib/xml/dtd/rm_pool.dtd.1">
<!--
Configuration for pools facility. Do NOT edit this file by hand - use
poolcfg(1) or libpool(3POOL) instead.
-->
<system ref_id="dummy" name="default" comment="" version="1" bind-
default="true">
  <property name="system.poold.objectives" type="string">wt-
load</property>
  <pool name="pool_default" active="true" default="true" importance="1"
comment="" res="pset_-1" ref_id="pool_0">
    <property name="pool.sys_id" type="int">0</property>
  </pool>
  <res_comp type="pset" sys_id="-1" name="pset_default" default="true"
min="1" max="65536" units="population" comment="" ref_id="pset_-1">
    <property name="pset.load" type="uint">388</property>
    <property name="pset.size" type="uint">2</property>
    <comp type="cpu" sys_id="1" comment="" ref_id="cpu_1">
      <property name="cpu.status" type="string">on-line</property>
    ...
  ...

```

The XML file contains the default pool configuration that you saved in step 2.

- Display the resource pool configuration from the `config` file by using the `poolcfg` command.

```

root@s11-testbed:~# poolcfg -c info

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

    pool pool_default
        int     pool.sys_id 0
        boolean pool.active true
        boolean pool.default true
        int     pool.importance 1
        string  pool.comment
        pset    pset_default
...

```

...

Note that this display is exactly the same as in step 6 of Task 1. The purpose of displaying it again is so that you can view it another time before you modify it.

5. Create a pset called `pset_1to2` by using the `poolcfg` command.

```
root@s11-testbed:~# poolcfg -c 'create pset pset_1to2
(uint pset.min=1; uint pset.max=2)'
```

The pset is defined with a range of two CPUs (1–2). For example, the kernel can use one or two CPUs based on the workload.

6. Create a pool called `pool_zgateway` and associate it with the `pset_1to2` pset by using the `poolcfg` command.

```
root@s11-testbed:~# poolcfg -c 'create pool pool_zgateway
(string pool.scheduler="FSS")'
```

While creating `pool_zgateway`, you also optionally indicate the Fair Share Scheduler (FSS) as your default scheduling class.

```
root@s11-testbed:~# poolcfg -c 'associate pool pool_zgateway
(pset pset_1to2) '
root@s11-testbed:~# ls -l /etc/pool*
-rw-r--r-- 1 root root 1645 Dec 14 16:17 /etc/pooladm.conf
```

The pool configuration file has been modified (as is evident from the time stamp).

7. View the modified pool configuration by using the `poolcfg -c info` command.

```
root@s11-testbed:~# poolcfg -c info | more

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

    pool pool_default
        int     pool.sys_id 0
        boolean pool.active true
        boolean pool.default true
        int     pool.importance 1
        string  pool.comment
        pset    pset_default

    pool pool_zgateway
        boolean pool.active true
        boolean pool.default false
        string  pool.scheduler FSS
        int     pool.importance 1
```

```

        string pool.comment
        pset      pset_1to2

pset pset_default
    int      pset.sys_id -1
    boolean pset.default true
    uint     pset.min 1
    uint     pset.max 65536
    string   pset.units population
    uint     pset.load 42
    uint     pset.size 2
    string   pset.comment

    cpu

        int      cpu.sys_id 1
        string   cpu.comment
        string   cpu.status on-line

    cpu

        int      cpu.sys_id 0
        string   cpu.comment
        string   cpu.status on-line

pset pset_1to2
    int      pset.sys_id -2
    boolean pset.default false
    uint     pset.min 1
    uint     pset.max 2
    string   pset.units population
    uint     pset.load 0
    uint     pset.size 0
    string   pset.comment

```

```
root@s11-testbed:~#
```

This is your new pool configuration. The pset, pool, and CPUs are all associated and displayed as you specified. Note that `pset_1to2` currently shows only one CPU, which is the minimum. Maximum CPUs are used as needed. Output may slightly differ.

8. Validate the configuration by using the `pooladm -n -c` command. Commit the changes by using the `-c` option.

```

root@s11-testbed:~# pooladm -n -c
root@s11-testbed:~# pooladm -c

```


9. Display the current pool configuration by using the `poolcfg -dc info` command.

```
root@s11-testbed:~# poolcfg -dc info | more

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

    pool pool_zgateway
        int     pool.sys_id 1
        boolean pool.active true
        boolean pool.default false
        string  pool.scheduler FSS
        int     pool.importance 1
        string  pool.comment
        pset    pset_1to2

    pool pool_default
        int     pool.sys_id 0
        boolean pool.active true
        boolean pool.default true
        int     pool.importance 1
        string  pool.comment
    ...
    ...
```

This display should include your modifications (for example, the `pool_zgateway` pool and its `pset_pset_1to2` shown here).

10. Display all the active resource pools by using the `poolstat` command.

```
root@s11-testbed:~# poolstat -r all
```

id	pool	type	rid	rset	min	max	size	used	load
1	pool_zgateway	pset	1	pset_1to2	1	2	1	0.00	0.00
0	pool_default	pset	-1	pset_default	1	66K	1	0.00	0.03

Summary: The output shows the default pool as well as your new pool, `pool_zgateway`.

Task 3: Bind the zgateway Zone to the Persistent Resource Pool

1. Log in to the **s11-testbed** VM as the `oracle` user. Use `oracle1` as the password. Assume administrator privileges.
2. List the current state of the zones by using the `zoneadm` command.

```
root@s11-testbed:~# zoneadm list -iv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	dbzone	running	/muzones/dbzone	solaris10	excl
2	zgateway	running	/muzones/zgateway	solaris	excl

The zgateway and dbzone zones are both up and running.

3. Allocate the pool to zgateway.

```
root@s11-testbed:~# zonecfg -z zgateway set pool=pool_zgateway
```

4. Confirm that the pool allocation is included in the zone configuration.

```
root@s11-testbed:~# zonecfg -z zgateway info | grep pool
pool: pool_zgateway
... ..
```

The `info` sub option displays the pool that is allocated to the zgateway zone.

5. Reboot zgateway to activate the resource pool binding. Check whether the zone has rebooted and is currently running.

```
root@s11-testbed:~# zlogin zgateway init 6
root@s11-testbed:~# zoneadm list -iv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	zgateway	running	/muzones/zgateway	solaris	excl
2	dbzone	running	/muzones/dbzone	solaris	excl

Note that the reboot process might take a while to complete.

6. Log in to zgateway to confirm the availability of the resource pool.

```
root@s11-testbed:~# zlogin zgateway
[Connected to zone 'zgateway' pts/1]
Oracle Corporation      SunOS 5.11      11.1      September 2012
```

7. View the modified pool configuration by using the `poolcfg -dc info` command.

```
root@zgateway:~# 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_zgateway
    int      pool.sys_id 1
    boolean  pool.active true
    boolean  pool.default false
    string   pool.scheduler FSS
    int      pool.importance 1
    string   pool.comment
    pset     pset_1to2

    pset pset_1to2
        int      pset.sys_id 1
        boolean  pset.default false
        uint     pset.min 1
        uint     pset.max 2
        string   pset.units population
        uint     pset.load 1827
        uint     pset.size 1
        string   pset.comment

        cpu
            int      cpu.sys_id 0
            string   cpu.comment
            string   cpu.status on-line

root@zgateway:~#

```

This is your new pool configuration. The pset, the pool, and the CPUs are all associated with the zgateway zone as you specified.

8. Exit from zgateway zone and return to the global zone.

```

root@zgateway:~# exit
logout

[Connection to zone 'zgateway' pts/1 closed]
root@s11-testbed:~#

```

Do not perform the steps in the following “Additional note” section. They are for reference only.

Additional note: If you want to remove the resource pool configuration, perform the following steps

- a. Remove the pool configuration of zgateway by using the zonecfg command.

```
root@s11-testbed:~# zonecfg -z zgateway clear pool
```

- b. Reboot zgateway and check the zone to see if it is up and running.

```
root@s11-testbed:~# zlogin zgateway init 6
root@s11-testbed:~# zoneadm list -cv
```

c. Log in to zgateway and use the `poolcfg -dc info` command to check the resource pool configuration.

```
root@s11-testbed:~# zlogin zgateway
root@zgateway:~# poolcfg -dc info
```

Practices for Lesson 7: Administering Privileges and RBAC

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

The testbed environment currently has three users: the default `root` account, `oracle` with administrative privileges, and `sadmin` with non-administrative privileges. You now need to create additional user accounts and then assign relevant permissions and privileges so that the new accounts can start using the system.

You first create two user accounts for the `zgateway` zone. Details for the user accounts and groups are specified in the following tables:

Group Name	Group ID
support	110
itgroup	120

User Account	Password	Shell	User ID	Primary Group	Secondary Group	Description
dkumar	Mypass1	default	1002	support	itgroup	Regular user
tshane	Mypass1	korn	1005	support		File system management, backup and recovery operations

In this practice, you perform the following tasks:

- Create user accounts.
- Configure disk quotas for users.
- Configure Role-Based Access Control (RBAC).

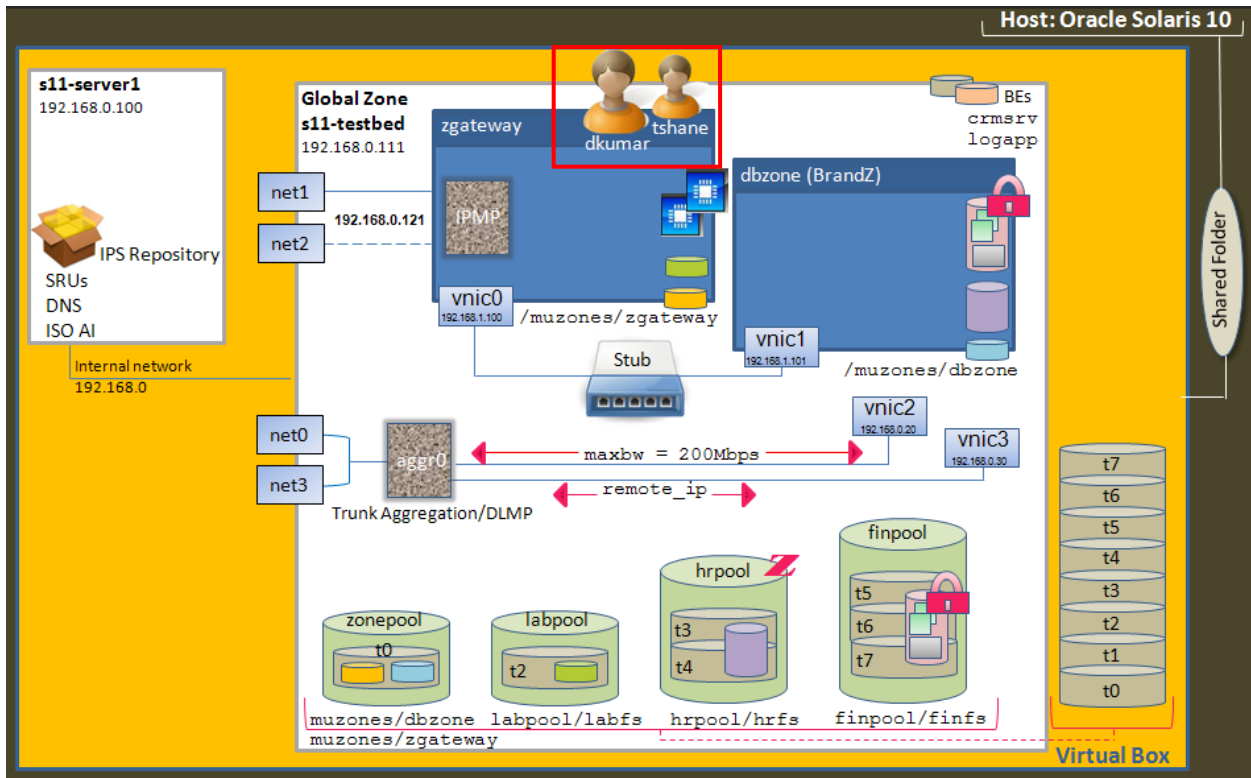
Notes

- Command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.

Practice 7-1: Creating User Accounts

Overview

Over a period of time, there will be a pool of user accounts. For now, you need to create two non-administrative users for the `zgateway` zone, `dkumar` and `tshane`. In addition to the default permissions, privileges, and authorizations assigned to a user account when it is created, you will assign additional attributes to the users based on specific requirements.



In this practice, you perform the following tasks:

- Set account defaults.
- Add a group.
- Add a user.
- Mount the user's home directory.
- Verify the user account setup.

Tasks

1. Verify that the **s11-server1** and **s11-testbed** VMs are running.
2. If the VMs are not running, start them now. Log in to the **s11-testbed** VM as the `oracle` user and then assume primary administrator privileges.
3. Verify that the `zgateway` zone is in running state.

```
root@s11-testbed:~# zoneadm list -cv
```

ID	NAME	STATUS	PATH	BRAND	IP
0	global	running	/	solaris	shared
1	dbzone	running	/muzones/dbzone	solaris10	excl
2	zgateway	running	/muzones/zgateway	solaris	excl

Note: If the zone is not in running state, use the following command to start the `zgateway` zone:

```
root@s11-testbed:~# zoneadm -z zgateway boot
```

4. Log in to the `zgateway` zone by using the `zlogin` command.

```
root@s11-testbed:~# zlogin zgateway
```

```
Oracle Corporation      SunOS 5.11      11.0      November 2011
root@zgateway:~#
```

5. Check whether the account default file exists. If it does not, create the account defaults by using the `useradd -D` command.

```
root@zgateway:~# ls /usr/sadm/defadduser
```

```
/usr/sadm/defadduser: No such file or directory
```

```
root@zgateway:~# useradd -D
```

```
group=staff,10 project=default,3 basedir=/export/home
skel=/etc/skel shell=/usr/bin/bash inactive=0
expire= auths= profiles= roles= limitpriv=
defaultpriv= lock_after_retries=
```

These default values can be modified for the new users you are creating based on requirements. For example, you can change the shell value to `/bin/ksh`.

6. Create the groups called `support` and `itgroup` according to the specifications.

```
root@zgateway:~# groupadd -g 110 support
```

```
root@zgateway:~# groupadd -g 120 itgroup
```

```
root@zgateway:~# grep support /etc/group
```

```
support::110:
```

```
root@zgateway:~# grep itgroup /etc/group
```

```
itgroup::120:
```

Recall that the groups are added to `/etc/group`.

7. Now you are ready to create the users `dkumar` and `tshane`. Use the `useradd` command and specify the user attributes listed in the specifications.

```
root@zgateway:~# useradd -u 1002 -g support -G itgroup
-d /export/home/dkumar -m -c "dileep kumar" dkumar
80 blocks
root@zgateway:~# useradd -u 1005 -g support
-d /export/home/tshane -m -c "tom shane" -s /bin/ksh tshane
80 blocks
```

What shell is assigned to `dkumar`? *The default `bash` shell*

8. Verify the creation of the `dkumar` and `tshane` users in the `/etc/passwd` file.

```
root@zgateway:~# grep dkumar /etc/passwd
dkumar:x:1002:110:dileep kumar:/export/home/dkumar:/usr/bin/bash
root@zgateway:~# grep tshane /etc/passwd
tshane:x:1005:110:tom shane:/export/home/tshane:/bin/ksh
```

9. Check whether the new users have entries in the `/etc/shadow` file.

```
root@zgateway:~# grep dkumar /etc/shadow
dkumar:UP:::::::
root@zgateway:~# grep tshane /etc/shadow
tshane:UP:::::::
```

When a new user is created, the account is flagged with `UP` for “undefined password.”

10. Create passwords for the users based on the specifications provided earlier.

```
root@zgateway:~# passwd dkumar
New Password: Mypass1
Re-enter new Password: Mypass1
passwd: password successfully changed for dkumar
```

```
root@zgateway:~# passwd tshane
New Password: Mypass1
Re-enter new Password: Mypass1
passwd: password successfully changed for tshane
```

Check how the password entries for these users have changed in `/etc/shadow`.

```
root@zgateway:~# grep dkumar /etc/shadow
dkumar:$5$kMYiNKAv$DJv6v2SEsfiePVHg8IDpIoqm0.DDRoCnoI4Xk7kXPJ1:16016::::::22640
root@zgateway:~# grep tshane /etc/shadow
tshane:$5$4HWZ90Rb$ocu70qzrhPDDcsjIumQ6XSOckIxzHifw9fy55d6xYcD:16016::::::22688
root@zgateway:~#
```

Now you see valid information for the users. One field is `16016` for both users. Can you tell what it represents? It represents the date on which the password was last modified, which is recorded as the number of days from January 1, 1970, to the modification date.

- ```
root@zgateway:~# grep support /etc/group
support::110:
root@zgateway:~# grep itgroup /etc/group
itgroup::120:dkumar
```

```
root@zgateway:~# id -G dkumar
110 120
root@zgateway:~# id -G tshane
110
```

12. Verify the home directories of the new users by using the `zfs list` command.

```
root@zgateway:~# zfs list -r rpool
```

| NAME                            | USED | AVAIL | REFER | MOUNTPOINT          |
|---------------------------------|------|-------|-------|---------------------|
| ...                             |      |       |       |                     |
| ...                             |      |       |       |                     |
| <b>rpool/export/home/dkumar</b> | 35K  | 32.1G | 35K   | /export/home/dkumar |
| <b>rpool/export/home/tshane</b> | 35K  | 32.1G | 35K   | /export/home/tshane |
| ...                             |      |       |       |                     |
| ...                             |      |       |       |                     |

**Note:** You can use the following steps to force `tshane` to create a new password on the next login.

```
root@zgateway:~# passwd -f tshane
passwd: password information changed for tshane
```

The `passwd` command is the main command to manage passwords. Here the `-f` option is used to expire `tshane`'s password. Check the effect of this command on the content of the `/etc/shadow` file.

```
root@zgateway:~# grep tshane /etc/shadow
tshane:$5$4HWZ90Rb$ocu70qzrhPDDcsjIumQ6XSOCkIxzHifw9fy55d6xYcD:0:::
```

Can you tell what information was changed in `/etc/shadow`? *The “last change” date was changed. It is now set to zero, indicating that the password has expired.*

Confirm it.

```
root@zgateway:~# su - oracle
```

**Note:** If you try to log in to the `tshane` account as an administrator user, you are not prompted for a password. Therefore, you must first switch to a non-administrator user, that is, `oracle`.

Use the new password `Newpass1`.

```
oracle@zgateway:~$ su - tshane
Password: Mypass1
su: Password for user 'tshane' has expired
New Password: Newpass1
Re-enter new Password: Newpass1
su: password successfully changed for tshane
Oracle Corporation SunOS 5.11 11.1 September 2012
tshane@zgateway:~$
```

Switch back to the `oracle` user and assume administrator privileges.

```
tshane@zgateway:~$ exit
oracle@zgateway:~$ exit
logout
root@zgateway:~#
```

How can you tell if `tshane`'s password expired on the first login? *By examining the first system message after login (for example, by using the `su` command, which is similar to logging in)*

```
root@zgateway:~# grep tshane /etc/shadow
tshane:5ldQR/Wec$8JvV3Mhfy5hc.xrD6HTJyhBZvuRq9UmAxoIX2kH6hG4:16016::::::22800
oot@zgateway:~#
```

Do you see any change in `tshane`'s entry in `/etc/shadow`? *Yes. The "last change" date is modified.*

## Practice 7-2: Setting User and Group Quotas on a ZFS File System

### Overview

Recall that you have already assigned quotas and reservations to datasets in ZFS file systems. Similarly, you can assign storage quotas for user accounts and groups. For now, the storage requirement is not concrete. It is therefore a good opportunity to try a few options for assigning quotas and observing the implications of setting storage space for users and groups. Although it looks like storage space is being allocated for the user, quotas are actually set at the directory level and not at the user level.

In this practice, you set user quotas for `dkumar` and `tshane` and the group quota for the `support` group.

### Task 1: Setting the User and Group Quotas on a ZFS File System

1. Log in to the `s11-testbed` VM as the `oracle` user and then assume administrator privileges.
2. Log in to the `zgateway` zone by using the `zlogin` command.

```
root@s11-testbed:~# zlogin zgateway
Oracle Corporation SunOS 5.11 11.0 November 2011
root@zgateway:~#
```

3. View the file systems of `labfs`.

```
root@zgateway:~# zfs list -r labfs
NAME USED AVAIL REFER MOUNTPOINT
labfs 84K 976M 34K /labfs
labfs@snap1 19K - 32K -
labfs/labfs1 31K 976M 31K /labfs/labfs1
root@zgateway:~#
```

4. Change the permission of the `labfs` file system.

```
root@zgateway:~# chmod -R 777 /labfs
```

5. Display the current user quotas for `dkumar` and `tshane`.

```
root@zgateway:~# zfs get userquota@dkumar labfs/labfs1
NAME PROPERTY VALUE SOURCE
labfs/labfs1 userquota@dkumar none local
root@zgateway:~# zfs get userquota@tshane labfs/labfs1
NAME PROPERTY VALUE SOURCE
labfs/labfs1 userquota@tshane none local
root@zgateway:~#
```

6. Display the current group quota for the `support` group.

```
root@zgateway:~# zfs get groupquota@support labfs/labfs1
NAME PROPERTY VALUE SOURCE
labfs/labfs1 groupquota@support none local
root@zgateway:~#
```

7. Display the user and group disk space usage by querying the following properties:

```
root@zgateway:~# zfs userspace labfs/labfs1
TYPE NAME USED QUOTA
POSIX User root 3K none
root@zgateway:~# zfs groupspace labfs/labfs1
TYPE NAME USED QUOTA
POSIX Group root 3K none
root@zgateway:~#
```

8. Set the user quota for dkumar to 2 MB by using the userquota command. Then display the result.

```
root@zgateway:~# zfs set userquota@dkumar=2M labfs/labfs1
root@zgateway:~# zfs get userquota@dkumar labfs/labfs1
NAME PROPERTY VALUE SOURCE
labfs/labfs1 userquota@dkumar 2M local
root@zgateway:~#
```

9. Set the user quota for tshane to 10 MB by using the userquota command. Then display the result.

```
root@zgateway:~# zfs set userquota@tshane=10M labfs/labfs1
root@zgateway:~# zfs get userquota@tshane labfs/labfs1
NAME PROPERTY VALUE SOURCE
labfs/labfs1 userquota@tshane 10M local
root@zgateway:~#
```

10. Set the group quota for the support group to 10 MB by using the groupquota command.

```
root@zgateway:~# zfs set groupquota@support=10M labfs/labfs1
root@zgateway:~# zfs get groupquota@support labfs/labfs1
NAME PROPERTY VALUE SOURCE
labfs/labfs1 groupquota@support 10M local
root@zgateway:~#
```

## Task 2: Verify User and Group Quotas

- Log in to the **s11-testbed** VM as the **oracle** user and then assume administrator privileges.
- Log in to the **zgateway** zone by using the **zlogin** command.

```
root@s11-testbed:~# zlogin zgateway
Oracle Corporation SunOS 5.11 11.0 November 2011
root@zgateway:~#
```

3. Switch to the **dkumar** user account.

```
root@zgateway:~# su - dkumar
Oracle Corporation SunOS 5.11 11.1 September 2012
```

Now, create **crm1** file with 1MB.

```
dkumar@zgateway:~$ mkfile 1m /labfs/labfs1/crm1
```

Note that **dkumar** has used 1 MB of the assigned quota of 2 MB disk space.

- Open another terminal window on **s11-testbed**. Log in to the **zgateway** zone as the **tshane** account.

```
oracle@s11-testbed:~$ su -
root@s11-testbed:~# zlogin zgateway
root@zgateway:~# su - tshane
Oracle Corporation SunOS 5.11 11.1 September 2012
```

Create a **crm2** file with 8 MB.

```
tshane@zgateway:~$ mkfile 8m /labfs/labfs1/crm2
```

Here, **tshane** has used 8 MB of the assigned quota of 10 MB disk space.

- Switch back to the terminal with the **dkumar** user account and try to create **crm3** with 1 MB.

```
dkumar@zgateway:~$ mkfile 1m /labfs/labfs1/crm3
```

Are you able to create **crm3**? Yes

Now try to create **crm4** with 5 MB.

```
dkumar@zgateway:~$ mkfile 5m /labfs/labfs1/crm4
```

Could not open /labfs/labfs1/crm4: Disc quota exceeded

```
dkumar@zgateway:~$ exit
```

Are you able to create **crm4**? No

Why? *The user quota of dkumar (2 MB) and the group quota of the support group have exceeded the allotted quota limit.*

Here, **dkumar** has used all the disk space (2 MB) from the assigned quota of 2 MB.

**tshane** has used 8 MB of disk space from the assigned quota of 10 MB. Because the group quota of the **support** group (10 MB) has been exceeded, **dkumar** is unable to create files.

- Close the terminal windows.

**Summary:** Because **dkumar** has exceeded the quota limit, the **crm4** file could not be created. As a system administrator, you must ensure that you assign sufficient quotas for user accounts so that their storage needs can be sufficiently addressed.

**Note:** Do not perform the steps in the following “Additional note.” They are for reference only.

**Additional note:** If you want to remove quotas, use the following commands.

```
root@zgateway:~# zfs set userquota@dkumar=None labfs/labfs1
```

```
root@zgateway:~# zfs set userquota@tshane=None labfs/labfs1
```

```
root@zgateway:~# zfs set groupquota@support=None labfs/labfs1
```

## Practice 7-3: Configuring RBAC

### Overview

Role-Based Access Control (RBAC) enables you to create roles and assign specific privileges or authorizations to them. You can then assign these roles to the appropriate users. This saves resources because you do not have to assign privileges to individual users.

In this practice, you create a role named `sdown` and configure a profile named `Shut` with authorization to execute the `shutdown` command on the `zgateway` zone. A non-administrative user usually cannot use the `shutdown` command. However, based on a user's job role, you can provision a role with the relevant profile and then assign it to a user.

In this practice, you perform the following tasks:

- Configure a role.
  - Create a role.
  - Configure a rights profile.
  - Assign the rights profile to the role.
  - Assume the role.
- Assign a profile directly to a user.
- Assign an authorization directly to a user.
- Create a system-wide RBAC policy.

### Task 1: Configure a Role

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use the password `oracle1` and then assume administrator privileges.
3. Log in to the `zgateway` zone by using the `zlogin` command.

```
root@s11-testbed:~# zlogin zgateway
Oracle Corporation SunOS 5.11 11.0 November 2011
root@zgateway:~#
```

4. Add a role called `sdown` by using the `roleadd` command. Create a password for the `sdown` role by using the `passwd` command.

```
root@zgateway:~# roleadd -u 3000 -g 10 -m -d /export/home/sdown sdown
80 blocks
root@zgateway:~# passwd sdown
New Password: sdown123
Re-enter new Password: sdown123
passwd: password successfully changed for sdown
```

The `sdown` role has been added and the password has been created.

5. Verify the entries that were created in various files.

```
root@zgateway:~# grep sdown /etc/passwd
sdown:x:3000:10:::/export/home/sdown:/usr/bin/pfbash
root@zgateway:~# getent user_attr | grep sdown
sdown:::type=role;profiles=All;roleauth=role
```

As you can see, an entry in `/etc/passwd` was created very much like an entry for a new user. Notice the default shell.

An entry is also made in `/etc/user_attr` for `sdown`, which is marked as a role.

6. Create a Shut profile by using the `profiles` command. When assigned to a user, this profile enables the user to shut down the `zgateway` zone.

```
root@zgateway:~# profiles -p Shut
profiles:Shut> set desc="Able to shutdown the system"
profiles:Shut> add cmd=/usr/sbin/shutdown
profiles:Shut:shutdown> set uid=0
profiles:Shut:shutdown> end
profiles:Shut> commit
profiles:Shut> exit
root@zgateway:~# getent prof_attr | grep Shut
Shut::Able to shutdown the system:
root@zgateway:~# getent exec_attr | grep Shut
Shut:solaris:cmd:::/usr/sbin/shutdown:uid=0
```

You have created a new rights profile called `Shut`.

7. Assign the Shut profile to the `sdown` role by using the `rolemod` command.

```
root@zgateway:~# rolemod -P Shut sdown
```

Verify that the changes have been made in the `/etc/user_attr` file.

```
root@zgateway:~# getent user_attr | grep sdown
sdown:::type=role;profiles=Shut;roleauth=role
root@zgateway:~#
```

Note the `profiles` entry in the `/etc/user_attr` file.

8. Assign the `sdown` role to `dkumar`. Confirm that an entry has been made in the `/etc/user_attr` file.

```
root@zgateway:~# usermod -R sdown dkumar
root@zgateway:~# getent user_attr | grep dkumar
dkumar::: roles=sdown
```

Note the entry in `/etc/user_attr` for `dkumar` with the `sdown` role.

9. Log in to the `dkumar` account and use the `shutdown` command to reboot the zone.

```
root@zgateway:~# su - dkumar
Oracle Corporation SunOS 5.11 11.1 September 2012
dkumar@zgateway:~$ /usr/sbin/shutdown -i 6 -g 0
/usr/sbin/shutdown: Only root can run /usr/sbin/shutdown
```

As expected, `dkumar` does not have the privileges to shut down the zone.



10. Use the `profiles` and `roles` commands to determine `dkumar`'s privileges.

```
dkumar@zgateway:~$ profiles
 Basic Solaris User
 All
dkumar@zgateway:~$ roles
sdown
```

*dkumar was assigned the sdown role. When? When you created the account*

11. Log in with the `sdown` role and use the `init` command to shut down the zone.

```
dkumar@zgateway:~$ su sdown
Password: sdown123
Oracle Corporation SunOS 5.11 11.0 November 2011
sdown@zgateway:~$ id
uid=3000(sdown) gid=10(staff)
sdown@zgateway:~$ /usr/sbin/init 6
Must be super-user
```

*Why is dkumar unable to reboot the zone? dkumar does not have the privilege of using the init command. dkumar can use only the shutdown command, as specified in profiles.*

12. Obtain the commands that `dkumar` is privileged to use by using the `profiles -l` command.

```
sdown@zgateway:~$ profiles -l
 Shut
 /usr/sbin/shutdown uid=0
 Basic Solaris User
 auths=solaris.mail.mailq,solaris.network.autoconf.read,solaris.admin.wusb
 .read
 profiles=All
 /usr/bin/cdrecord.bin
 privs=file_dac_read,sys_devices,proc_lock_memory,proc_prioctl,net_privaddr
 /usr/bin/readcd.bin
 privs=file_dac_read,sys_devices,net_privaddr
 /usr/bin/cdda2wav.bin
 privs=file_dac_read,sys_devices,proc_prioctl,net_privaddr
 All
 *
sdown@zgateway:~$
```

*Does the sdown role have the privilege to execute the init command? No*

*Can the sdown role execute the shutdown command? Yes, as part of the Shut profile*

13. Use the `shutdown` command to attempt to bring down the zone. To save time, respond with `n` when prompted to continue shutting down.

```
sdown@zgateway:~$ /usr/sbin/shutdown -i 6 -g 0

Shutdown started. Fri Nov 8 05:24:30 AM MDT

Do you want to continue? (y or n): n
Broadcast Message from root (pts/2) on zgateway Fri Dec 16 20
05:24:38...
False Alarm: The system zgateway will not be brought down.
Shutdown aborted.
sdown@zgateway:~$
```

Is the user `dkumar` able to execute the `shutdown` command? Yes

14. Display the profiles assigned to the `sdown` role by using the `profiles` command. When done, log out of the `sdown` role and `dkumar`'s account by using `exit`.

```
sdown@zgateway:~$ profiles
Shut
Basic Solaris User
All
sdown@zgateway:~$ exit
dkumar@zgateway:~$ exit
root@zgateway:~# exit
```

**Summary:** The `sdown` role has three profiles assigned to it: `Shut`, `Basic Solaris User`, and `All`. Therefore, when the `sdown` role is assigned to `dkumar`, the user is able to execute the `shutdown` command.

**Note:** Do not perform the step in the following "Additional note." It is for reference only.

**Additional note:** If you want to delete the `Shut` profile from the profiles assigned to the `sdown` role, use the `rolemod` command to delete the profile.

```
root@zgateway:~# rolemod -P "Basic Solaris User,All,Stop" sdown
```

## Task 2: Assign a Profile Directly to a User

In addition to assigning a profile to a role and then assigning the role to a user or a group of users, you can assign profiles directly to a user. Recall that the `tshane` user account has been specifically created for managing and scheduling data backups across various file systems. You therefore need to assign the `File System Management` profile to `tshane`, which is not one of the default profiles assigned to a user when a user account is created.

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use the password `oracle1` and then assume administrator privileges.
3. Log in to the `zgateway` zone by using the `zlogin` command.

```
root@s11-testbed:~# zlogin zgateway
Oracle Corporation SunOS 5.11 11.0 November 2011
root@zgateway:~#
```

- Assign the profile File System Management to the existing user tshane by using the usermod command. Verify the entry in the /etc/user\_attr file.

```
root@zgateway:~# usermod -P "File System Management" tshane
root@zgateway:~# getent user_attr | grep tshane
tshane:::profiles=File System Management
```

- Log in to the tshane account. Use the profiles command to display the current profiles assigned to tshane.

```
root@zgateway:~# su - tshane
Oracle Corporation SunOS 5.11 11.1 September 2012
tshane@zgateway:~$ profiles
 File System Management
 SMB Management
 VSCAN Management
 SMBFS Management
 Shadow Migration Monitor
 ZFS File System Management
 Basic Solaris User
 All

In addition to the default profiles for tshane, observe that the File System
Management profile is now included together with other profiles of the user.
```

- Attempt to create a directory in the root file system by using the mkdir command.

```
tshane@zgateway:~$ mkdir /holtdir
mkdir: Failed to make directory "/holtdir"; Permission denied

Can tshane create a directory in the root file system? No
```

- Execute the mkdir command by using the pfexec command. Confirm the directory creation.

```
tshane@zgateway:~$ pfexec mkdir /holtdir
tshane@zgateway:~$ cd /;ls -l | grep holt
drwxr-xr-x 2 root support 2 Dec 16 15:20 holtdir

tshane@zgateway:/$ exit
logout
root@zgateway:~# exit
```

**Summary:** The pfexec command temporarily enables the user to assume the privileges assigned in the profile.

### Task 3: Assign an Authorization Directly to a User

Just as a profile can be assigned to a user, authorizations can also be assigned to a user. Because the `tshane` user is responsible for managing data and scheduling backups, `tshane` might occasionally need to modify the scheduling details in the `crontab` file. However, a non-administrative user is not authorized to view or modify the `crontab` file of the root account. You need to make the necessary changes to the authorizations so that `tshane` is able to access the file when necessary.

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use the password `oracle1` and the assume administrator privileges.
3. Log in to the `zgateway` zone by using the `zlogin` command.

```
root@s11-testbed:~# zlogin zgateway
Oracle Corporation SunOS 5.11 11.0 November 2011
root@zgateway:~#
```

4. Log in to the `tshane` account. Use the `crontab` command to determine if the user has the authorization to display the `crontab` content for the superuser.

```
root@zgateway:~# su - tshane
Oracle Corporation SunOS 5.11 11.1 September 2012
tshane@zgateway:~$ crontab -l root
crontab: you must be super-user to access another user's crontab file
tshane@zgateway:~$ exit
root@zgateway:~#
```

As expected, the `tshane` account does not have the authorization to list the `root`'s `crontab` file.

5. As an administrator, assign Tom Shane the authorization for job administration by using the `usermod` command.

```
root@zgateway:~# usermod -A solaris.jobs.admin tshane
root@zgateway:~# getent user_attr |grep tshane
tshane:::auths=solaris.jobs.admin:profiles=File System Management
```

Does `tshane` have the right authorizations now? Yes.

6. Log in as `tshane` and issue the `crontab` command now. When done, log out of `tshane`'s account to return to the superuser account.

```
root@zgateway:~# su - tshane
Oracle Corporation SunOS 5.11 11.1 September 2012
tshane@zgateway:~$ crontab -l root
#ident "%Z%M% %I% %E% SMI"
#
Copyright 2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
#
The root crontab should be used to perform accounting data
collection.
```

```
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [-x /usr/lib/fs/nfs/nfsfind] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [-x /usr/lib/gss/gsscred_clean] &&
/usr/lib/gss/gsscred_clean
tshane@zgateway:~$ exit
root@zgateway:~# exit
```

**Summary:** You can see that `tshane` can now access the `crontab` file for the root account.

**Note:** Do not perform the steps in the following “Additional note.” They are for reference only.

**Additional note:** If you want to revoke (remove) the authorization from `tshane`, perform the following steps:

```
root@zgateway:~# usermod -A "" tshane
root@zgateway:~# getent user_attr | grep tshane
tshane:::auths=;profiles=File System Management
root@zgateway:~# su - tshane
Oracle Corporation SunOS 5.11 11.1 September 2012
tshane@zgateway:~$ crontab -l root
crontab: you must be super-user to access another user's crontab file
tshane@zgateway:~$ exit
logout
The authorization has been revoked. Now tshane cannot access the superuser's
crontab file.
```

#### Task 4: Create a System-Wide RBAC Policy

So far, you have applied profiles and authorizations at a granular level. Considering the number of users that the system or the zone will eventually have, and taking into account the security needs of the infrastructure, this is the right time to create a system-wide user access policy. To start, *all* users should be denied the permission to view processes other than their own. Currently, `dkumar` and `tshane` can view all processes of all users.

1. Verify that the **s11-testbed** VM is running. If it is not running, start it now.
2. Log in to the **s11-testbed** VM as the `oracle` user. Use the password `oracle1` and then assume administrator privileges.
3. Log in to the `zgateway` zone by using the `zlogin` command.

```
root@s11-testbed:~# zlogin zgateway
Oracle Corporation SunOS 5.11 11.0 November 2011
root@zgateway:~#
```

4. Log in to the `dkumar` account. Display the privilege sets by using the `ppriv` command.

```
root@zgateway:~# su - dkumar
Oracle Corporation SunOS 5.11 11.1 September 2012
dkumar@zgateway:~$ ppriv $$
16944: -bash
flags = <none>
```

```

E: basic
I: basic
P: basic
L:
basic,contract_event,contract_identity,contract_observer,file_chown,file_chown_self,file_dac_execute,file_dac_read,file_dac_search,file_dac_write,file_owner,file_setid,ipc_dac_read,ipc_dac_write,ipc_owner,net_bindmlp,net_icmpaccess,net_mac_aware,net_observability,net_privaddr,net_rawaccess,proc_audit,proc_chroot,proc_lock_memory,proc_owner,proc_setid,proc_taskid,sys_acct,sys_admin,sys_audit,sys_flow_config,sys_ip_config,sys_iptun_config,sys_mount,sys_nfs,sys_ppp_config,sys_resource,sys_share
dkumar@zgateway:~$

```

5. Display all the processes by using the `ps` command.

```

dkumar@zgateway:~$ ps -A -o user -o pid -o comm | more
USER PID COMMAND
root 0 sched
root 5 zpool-rpool
root 1 /usr/sbin/init
root 2 pageout
root 3 fsflush
root 6 intrd
root 7 vmtasks
root 427 /sbin/dhcpagent
root 10 /lib/svc/bin/svc.startd
root 12 /lib/svc/bin/svc.configd
daemon 75 /lib/crypto/kcfd
netadm 96 /lib/inet/ipmgmt
root 114 /lib/inet/in.mpathd
dladm 43 /usr/sbin/dlmgmt
netcfg 48 /lib/inet/netcfgd
root 2493 su
oracle 2356 /usr/lib/clock-applet
root 119 /usr/lib/pfexecd
daemon 1840 /usr/lib/nfs/nfs4cbd
root 756 lockd_kproc
oracle 2309 nautilus...
...
...
Can dkumar display the processes for any user? Yes

```

6. Exit the `dkumar` account. Then, as the administrator, modify the `/etc/security/policy.conf` file as follows:

```

dkumar@zgateway:~$ exit
logout
root@zgateway:~# vi /etc/security/policy.conf
root@zgateway:~# grep PRIV_DEFAULT /etc/security/policy.conf
There are two different settings; PRIV_DEFAULT determines the default

```

```
Similarly, PRIV_DEFAULT=basic,!file_link_any takes away only the
#PRIV_DEFAULT=basic
PRIV_DEFAULT=basic,!proc_info,!proc_session
```

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

The `policy.conf` file establishes a system-wide policy. The changes you made deny a non-administrative user the privilege to look at the processes of other users.

Now reboot the zone so that the policy can take effect.

```
root@zgateway:~# init 6
```

**Note:** The reboot may take a few minutes to complete. Log in to the `zgateway` zone after completion of the reboot.

7. Log in to the `dkumar` account and issue the same `ps` command to access the processes.

```
root@zgateway:~# su - dkumar
Oracle Corporation SunOS 5.11 11.1 September 2012
dkumar@zgateway:~$ ps -A -o user -o pid -o comm | more
 USER PID COMMAND
 dkumar 13721 -bash
 dkumar 13726 more
 dkumar 13725 ps
dkumar@zgateway:~$ ps -ef | more
 UID PID PPID C STIME TTY TIME CMD
 dkumar 13721 13720 0 07:47:06 pts/1 0:00 -bash
 dkumar 13727 13721 0 07:47:03 pts/1 0:00 ps -ef
```

**Summary:** The `dkumar` user can now view only the user's own processes. That would be the case for any user because a system-wide policy has been applied to the `policy.conf` file.

**Additional note:** To reset the process parameters in `/etc/security/policy.conf` to the original value, you can perform the following commands.

```
root@s11-testbed:~# zlogin zgateway
```

Remove the previously added line from the `policy.conf` file.

```
root@zgateway:~# vi /etc/security/policy.conf
```

```
...
PRIV_DEFAULT=basic,!proc_info,!proc_session
```

```
..
```

```
root@zgateway:~# grep PRIV_DEFAULT /etc/security/policy.conf
```

```
There are two different settings; PRIV_DEFAULT determines the default
```

```
Similarly, PRIV_DEFAULT=basic,!file_link_any takes away only the
```

```
#PRIV_DEFAULT=basic
```

```
root@zgateway:~#
```

Now reboot the zone to have the policy take effect.

```
root@zgateway:~# init 6
```

The reboot may take a few minutes to complete.

Log in to the zgateway zone. Then log in to the dkumar account.

```
root@s11-testbed:~# zlogin zgateway
```

```
root@zgateway:~# su - dkumar
```

```
Oracle Corporation SunOS 5.11 11.1 September 2012
```

```
dkumar@zgateway:~$ ps -ef | more
```

| UID  | PID | PPID | C | STIME    | TTY | TIME | CMD         |
|------|-----|------|---|----------|-----|------|-------------|
| root | 0   | 0    | 0 | 07:47:06 | ?   | 0:01 | sched       |
| root | 5   | 0    | 0 | 07:47:03 | ?   | 0:12 | zpool-rpool |
| root | 1   | 0    | 0 | 07:47:08 | ?   | 0:00 | /sbin/init  |
| root | 2   | 0    | 0 | 07:47:08 | ?   | 0:00 | pageout     |
| root | 3   | 0    | 0 | 07:47:08 | ?   | 0:18 | fsflush     |
| root | 6   | 0    | 0 | 07:47:08 | ?   | 0:00 | vmtasks     |

Now dkumar can display the processes for any user. Actually, any user can do this.

Exit the dkumar account.

```
dkumar@zgateway:~$ exit
```

```
logout
```

```
root@zgateway:~# exit
```

```
logout
```

```
[Connection to zone 'zgateway' pts/4 closed]
```

```
root@s11-testbed:~#
```



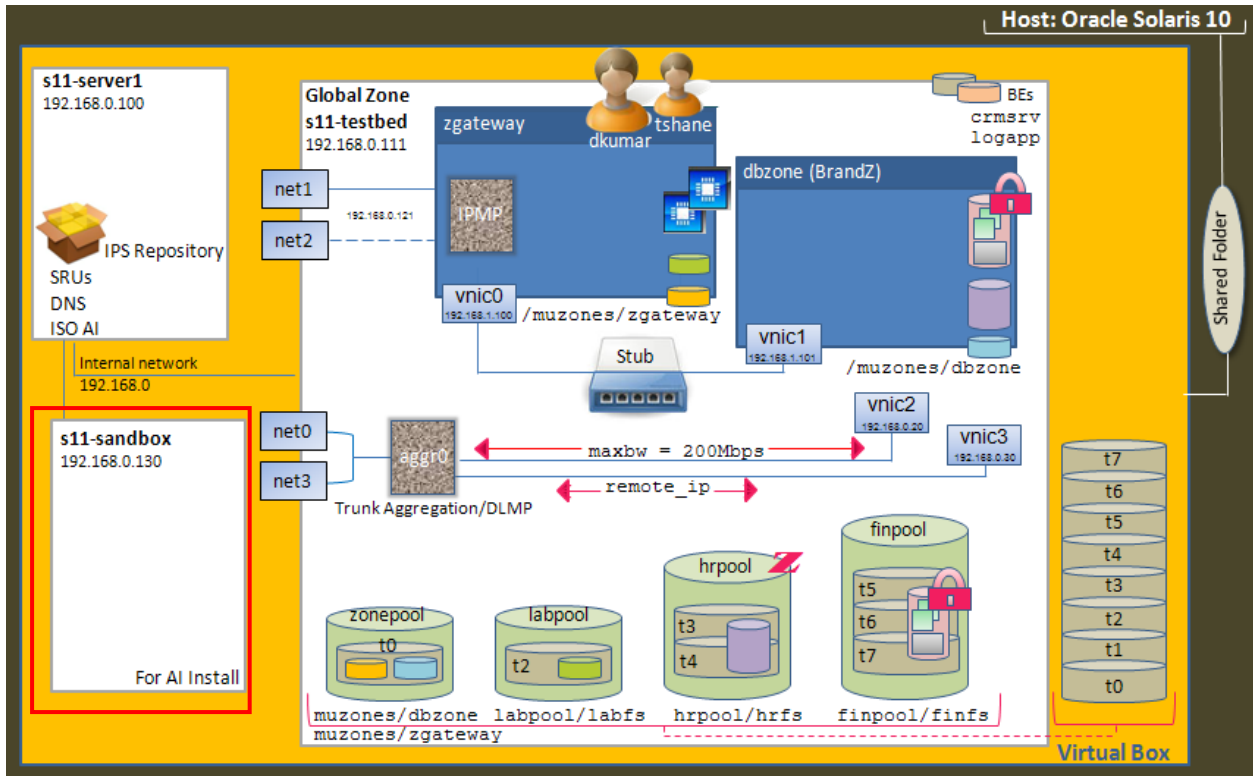
# **Practices for Lesson 8: Installing the Oracle Solaris 11.1 Operating System**

## **Chapter 8**

# Practices for Lesson 8: Overview

## Practices Overview

The initial framework of the testbed infrastructure is nearly complete. One of the last requirements is to install the Oracle Solaris 11.1 operating system (OS) on a separate system, **s11-sandbox**, and keep it ready for security-related testing and validation activities.



However, before you install the Oracle Solaris 11.1 OS by using the Automated Installer (AI), you must first download the Oracle Solaris 11.1 AI install image from the following site:

<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html>

The AI installation download is in an ISO image format that can be burned to a CD or DVD, or it can be used directly within Oracle VM Server or other virtualization software.

For this practice, the AI ISO image has already been downloaded for you. The ISO image file can be found in the `/opt/ora/iso` directory.

In this practice, you perform the following tasks:

- Verify the system AI requirements.
- Customize the AI service.
- Deploy the OS to a network client.

#### **Notes**

- Command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.
- Your system performance depends on the network speed and network load. If you find your VM too slow to proceed with, it is suggested that you restart the VM.

## Practice 8-1: Customizing the Automated Installation Service

### Overview

The Automated Installer (AI) enables you to customize your Oracle Solaris 11.1 installations by adding system configuration (SC) profiles. SC profiles are used to customize system attributes such as the host name, IP address, naming services, and use credentials of AI clients.

### Task 1: Verify the System AI requirements

Before installing Oracle Solaris 11.1, verify the system requirements for an AI configuration. You first need to configure an IPS repository so that you can minimize the package deployment time. Recall that you have already configured the **s11-server1** VM to be your local IPS repository. As a result, you need to verify only that the repository is available and accessible.

1. Verify that the **s11-server1** VM is running. If it is not running, start it at this time.
2. Log in to the **s11-server1** VM as the `oracle` user with the password `oracle1`.
3. Use the `su` command to assume primary administrator privileges.

```
oracle@s11-server1:~$ su -
Password: oracle1
Oracle Corporation SunOS 5.11 11.1 September 2012
root@s11-server1:~#
```

4. Verify that the operating system is Oracle Solaris 11.1.

```
root@s11-server1:~# cat /etc/release
 Oracle Solaris 11.1 X86

 Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights
 reserved.

 Assembled 19 September 2012
```

5. Verify that the OS is configured with a static IP address.

```
root@s11-server1:~# svcs network/physical:default
STATE STIME FMRI
online 0:24:39 svc:/network/physical:default
root@s11-server1:~# ipadm show-addr
ADDROBJ TYPE STATE ADDR
...
net0/v4 static ok 192.168.0.100/24
...
```

6. Verify that DNS is operational.

```
root@s11-server1:~# nslookup s11-server1.mydomain.com
Server: 192.168.0.100
Address: 192.168.0.100#53

Name: s11-server1.mydomain.com
Address: 192.168.0.100
```

7. Verify that the `application/pkg/server` service is enabled.

```
root@s11-server1:~# svcs application/pkg/server
STATE STIME FMRI
online 0:24:39 svc:/application/pkg/server:default
```

8. Verify that the publisher is set to `http://s11-server1.mydomain.com/`.

```
root@s11-server1:~# pkg publisher
PUBLISHER TYPE STATUS P LOCATION
solaris origin online F http://s11-server1.mydomain.com/
```

**Summary:** You verified that the IPS repository is properly configured for you to proceed with the remaining tasks of an AI installation.

## Task 2: Customize the AI Service

Now that you have AI working, you are ready to customize the AI service. In this task, you configure the AI server to automatically install an Oracle Solaris 11.1 client using the AI custom system configuration profile.

Perform the following steps on the **s11-server1** VM to customize the AI service:

1. On the **s11-server1** VM, check whether the `svc:/network/dns/multicast` service is online. If the service is not online, enable it.

```
root@s11-server1:~# svcs network/dns/multicast
STATE STIME FMRI
disabled 1:08:14 svc:/network/dns/multicast:default
root@s11-server1:~# svcadm enable network/dns/multicast
root@s11-server1:~# svcs network/dns/multicast
STATE STIME FMRI
online 1:32:27 svc:/network/dns/multicast:default
```

2. Verify that the `netmasks` file is configured appropriately for the DHCP service.

```
root@s11-server1:~# getent netmasks 192.168.0.0
```

Note that DHCP requires that the network mask for the local subnet be configured in the `/etc/netmasks` file. If an entry does not exist, update the `netmasks` file now.

```
root@s11-server1:~# vi /etc/netmasks
```

...

```
192.168.0.0 255.255.255.0
```

```
root@s11-server1:~# getent netmasks 192.168.0.0
```

```
192.168.0.0 255.255.255.0
```

3. Create a directory for the custom AI service.

```
root@s11-server1:~# mkdir -p /export/ai/custom_ai
```

4. Use the `installadm create-service` command to create another AI service based on the following information:
- Service name: `custom_ai`
  - DHCP base IP address: `192.168.0.135`
  - DHCP IP address range: `5`
  - AI ISO image location: `/opt/ora/iso/sol-11_1-ai-x86.iso`
  - Target directory: `/export/ai/custom_ai`

```
root@s11-server1:/# installadm create-service -n custom_ai -s
/opt/ora/iso/sol-11_1-ai-x86.iso -i 192.168.0.0 -c 5 -d
/export/ai/custom_ai

Creating service from: /opt/ora/iso/sol-11_1-ai-x86.iso
Setting up the image ...

Creating i386 service: custom_ai

Image path: /export/ai/custom_ai

Adding IP range to local DHCP configuration

Refreshing install services

Creating default-i386 alias

Setting the default PXE bootfile(s) in the local DHCP configuration
to:
bios clients (arch 00:00): default-i386/boot/grub/pxegrub2
uefi clients (arch 00:07): default-i386/boot/grub/grub2netx64.efi

Refreshing install services
root@s11-server1:/#
```

5. Use the `installadm list` command to verify that your AI service is installed.

```
root@s11-server1:/# installadm list

Service Name Alias Of Status Arch Image Path

custom_ai - on i386 /export/ai/custom_ai
default-i386 custom_ai on i386 /export/ai/custom_ai

root@s11-server1:/#
```

6. Use the `installadm create-client` command to add the MAC address of the s11-sandbox VM to the `custom_ai` service.
- Go to VirtualBox Manager.
  - Select **s11-sandbox**.
  - Click **Settings** on the menu bar of VirtualBox Manager.
  - In the Settings dialog box, select **Network** in the left pane.
  - On the Network tab, under Adapter 1, expand the Advanced section to view the MAC addresses.
  - Make a note of the MAC addresses. You use them frequently in this practice.

```
root@s11-server1:/# installadm create-client -e 08:00:27:85:C7:D6 -n
custom_ai
Adding host entry for 08:00:27:85:C7:D6 to local DHCP configuration.

root@s11-server1:/#
```

7. Use the `installadm list -c` command to verify that the client was added to the AI server `custom_ai`.

```
root@s11-server1:/# installadm list -c

Service Name Client Address Arch Image Path

custom_ai 08:00:27:85:C7:D6 i386 /export/ai/custom_ai

root@s11-server1:/#
```

8. Create a directory named `/var/tmp/manifests` to store the custom AI manifest files.

```
root@s11-server1:~# mkdir -p /var/tmp/manifests
```

9. Copy the `/export/ai/custom_ai/auto_install/manifest/default.xml` file to the `/var/tmp/manifests/custom_ai.xml` file.

```
root@s11-server1:~# cp
/export/ai/custom_ai/auto_install/manifest/default.xml
/var/tmp/manifests/custom_ai.xml
root@s11-server1:~# chmod 755 /var/tmp/manifests/custom_ai.xml
```

10. Modify the `/var/tmp/manifests/custom_ai.xml` file XML tag elements so that they reflect the following details:

- Change `<ai_instance name="default">` to `<ai_instance name="custom_ai" auto_reboot=true>`.
- Change `<origin name="http://pkg.oracle.com/solaris/release"/>` to `<origin name="http://s11-server1.mydomain.com"/>`.
- IPS package: `solaris-large-server` (Confirm that it uses the `solaris-large-server` package.)
- The modified file should look like this:

```
root@s11-server1:~# cat /var/tmp/manifests/custom_ai.xml
...
<auto_install>
 <ai_instance name="custom_ai" auto_reboot="true">
...
...
<source>
 <publisher name="solaris">
 <origin name="http://s11-server1.mydomain.com"/>
 </publisher>
...

```

11. Use the `diff` command to view the differences between the `custom_ai.xml` file and the `basic_ai.xml` file.

```
root@s11-server1:~# diff
/export/ai/custom_ai/auto_install/manifest/default.xml
/var/tmp/manifests/custom_ai.xml
9c9
< <ai_instance name="default">

> <ai_instance name="custom_ai" auto_reboot="true">
65c65
< <origin name="http://pkg.oracle.com/solaris/release"/>

> <origin name="http://s11-server1.mydomain.com"/>
root@s11-server1:~#

```

12. Create a MAC address-based criteria manifest named `criteria_custom_ai.xml` in the `/var/tmp/manifests` directory. Use the MAC addresses of the network client `s11-sandbox`.

```
root@s11-server1:~# vi /var/tmp/manifests/criteria_custom_ai.xml
root@s11-server1:~# cat /var/tmp/manifests/criteria_custom_ai.xml
<ai_criteria_manifest>
 <ai_criteria name="mac">
 <value>08:00:27:85:C7:D6</value>
 </ai_criteria>
</ai_criteria_manifest>

```



```
root@s11-server1:/#
```

**Note:** If the AI client does not match the criteria for a service (in this case, a specific MAC address), the AI service uses the default manifest when installing the OS.

13. Add the `custom_ai` manifest and criteria manifest to the `custom_ai` service. Then show the results.

```
root@s11-server1:/# installadm create-manifest -n custom_ai
-f /var/tmp/manifests/custom_ai.xml
-C /var/tmp/manifests/criteria_custom_ai.xml
root@s11-server1:/# installadm list -c -m
```

| Service Name | Client Address    | Arch | Image Path           |
|--------------|-------------------|------|----------------------|
| -----        | -----             | ---- | -----                |
| custom_ai    | 08:00:27:85:C7:D6 | i386 | /export/ai/custom_ai |

| Service/Manifest Name | Status  | Criteria                |
|-----------------------|---------|-------------------------|
| -----                 | -----   | -----                   |
| custom_ai             |         |                         |
| custom_ai             |         | mac = 08:00:27:85:C7:D6 |
| orig_default          | Default | None                    |
| default-i386          |         |                         |
| orig_default          | Default | None                    |

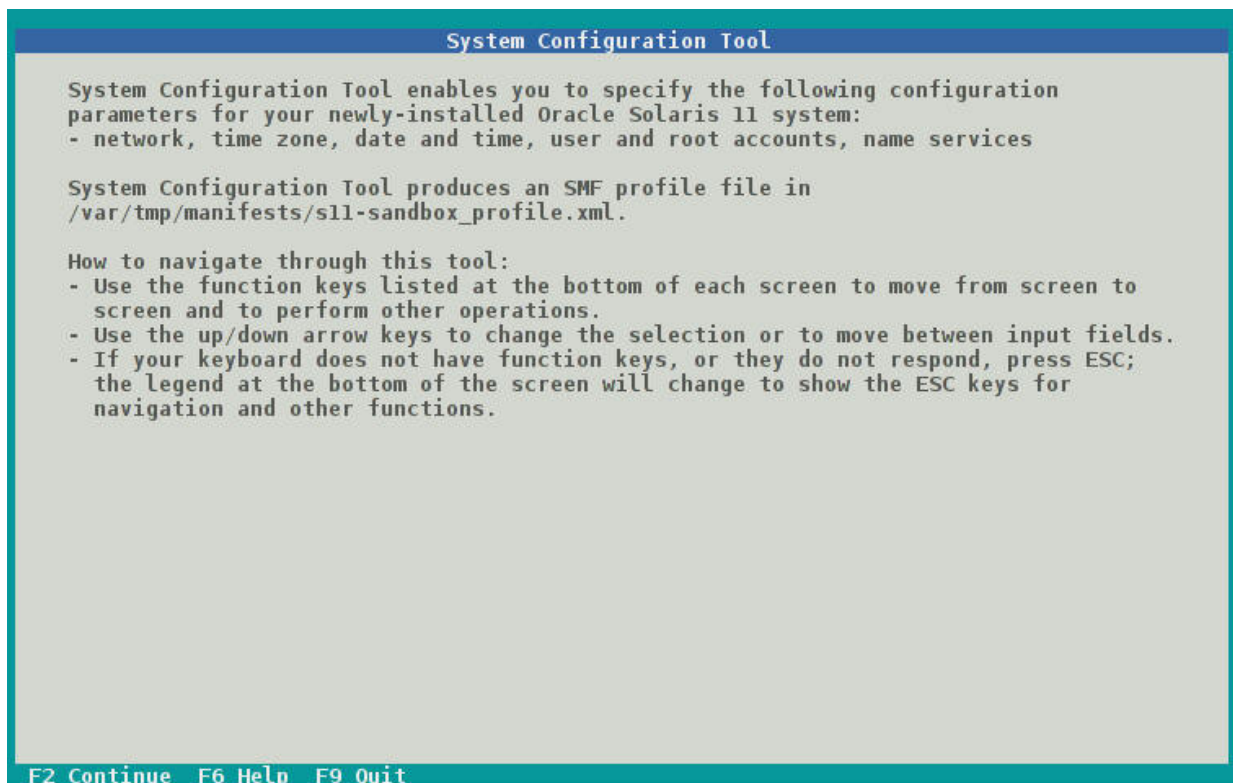
```
root@s11-server1:/#
```

14. Use the `sysconfig` utility to create a profile for **s11-sandbox** using the following properties:

- Host name: **s11-sandbox**
- Network type: **Manually**
- IPv4 interface name: **net0 (e1000g0)**
- Static IP address: **192.168.0.130**
- DNS Name Service: **Do not configure DNS.**
- Alternate Name Service: **None**
- Time zone: **Choose your local time zone.**
- Root password: **oracle1**
- Your real name: **oracle**
- Username: **oracle**
- User password: **oracle1**
- Support Registration: **Accept the default.**
- Support: Network Configuration: **Accept the default.**

```
root@s11-server1:~# sysconfig create-profile
-o /var/tmp/manifests/s11-sandbox_profile.xml
```

**Note:** The `sysconfig create-profile` utility launches a System Configuration Tool.



15. View the content of the **s11-sandbox** profile after the `sysconfig` utility has completed.

```
root@s11-server1:/# more /var/tmp/manifests/s11-sandbox_profile.xml
<!DOCTYPE service_bundle SYSTEM
"/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type="profile" name="sysconfig">
 <service version="1" type="service" name="system/config-user">
 <instance enabled="true" name="default">
 <property_group type="application" name="root_account">
 <propval type="astring" name="login" value="root"/>
 <propval type="astring" name="password"
value="5I9h9rzza$KBSBeSWUL6hBxOk8Kfdoh27TVgWYsTXXGP/D2bk3gO"/>
 <propval type="astring" name="type" value="role"/>
 </property_group>
 <property_group type="application" name="user_account">
 <propval type="astring" name="login" value="oracle"/>
 <propval type="astring" name="password"
value="5gZK7Xmqb$H1gSqEF2j3bXmONkdNaINxBIaSmc53n/9x/KuzuIbWB"/>
 <propval type="astring" name="type" value="normal"/>
 <propval type="astring" name="description" value="oracle1"/>
 <propval type="count" name="gid" value="10"/>
 <propval type="astring" name="shell" value="/usr/bin/bash"/>
 <propval type="astring" name="roles" value="root"/>
 <propval type="astring" name="profiles" value="System
Administrator"/>
 <propval type="astring" name="sudoers" value="ALL=(ALL) ALL"/>
 </property_group>
```

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

```

 </instance>
 </service>
 <service version="1" type="service" name="system/timezone">
 <instance enabled="true" name="default">
 <property_group type="application" name="timezone">
 <propval type="astring" name="localtime"
value="Asia/Kolkata"/>
 </property_group>
 </instance>
 </service>
 <service version="1" type="service" name="system/environment">
 <instance enabled="true" name="init">
 <property_group type="application" name="environment">
 <propval type="astring" name="LANG" value="en_US.UTF-8"/>
 </property_group>
 </instance>
 </service>
 <service version="1" type="service" name="system/identity">
 <instance enabled="true" name="node">
 <property_group type="application" name="config">
 <propval type="astring" name="nodename" value="s11-sandbox"/>
 </property_group>
 </instance>
 </service>
 <service version="1" type="service" name="system/keymap">
 <instance enabled="true" name="default">
 <property_group type="system" name="keymap">
 <propval type="astring" name="layout" value="US-English"/>
 </property_group>
 </instance>
 </service>
 <service version="1" type="service" name="system/console-login">
 <instance enabled="true" name="default">
 <property_group type="application" name="ttymon">
 <propval type="astring" name="terminal_type" value="sun-
color"/>
 </property_group>
 </instance>
 </service>
 <service version="1" type="service" name="network/physical">
 <instance enabled="true" name="default">
 <property_group type="application" name="netcfg">
 <propval type="astring" name="active_ncp"
value="DefaultFixed"/>
 </property_group>
 </instance>
 </service>

```

```

</service>
<service version="1" type="service" name="network/install">
 <instance enabled="true" name="default">
 <property_group type="application"
name="install_ipv4_interface">
 <propval type="astring" name="address_type" value="static"/>
 <propval type="net_address_v4" name="static_address"
value="192.168.0.130/24"/>
 <propval type="astring" name="name" value="net0/v4"/>
 </property_group>
 <property_group type="application"
name="install_ipv6_interface">
 <propval type="astring" name="stateful" value="yes"/>
 <propval type="astring" name="stateless" value="yes"/>
 <propval type="astring" name="address_type" value="addrconf"/>
 <propval type="astring" name="name" value="net0/v6"/>
 </property_group>
 </instance>
</service>
<service version="1" type="service" name="system/name-
service/switch">
 <property_group type="application" name="config">
 <propval type="astring" name="default" value="files"/>
 <propval type="astring" name="printer" value="user files"/>
 </property_group>
 <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="system/name-
service/cache">
 <instance enabled="true" name="default"/>
</service>
<service version="1" type="service" name="network/dns/client">
 <instance enabled="false" name="default"/>
</service>
<service version="1" type="service" name="system/ocm">
 <instance enabled="true" name="default">
 <property_group type="application" name="reg">
 <propval type="astring" name="user"
value="anonymous@oracle.com"/>
 <propval type="astring" name="password" value=""/>
 <propval type="astring" name="key" value=""/>
 <propval type="astring" name="cipher" value=""/>
 <propval type="astring" name="proxy_host" value=""/>
 <propval type="astring" name="proxy_user" value=""/>
 <propval type="astring" name="proxy_password" value=""/>
 <propval type="astring" name="config_hub" value=""/>
 </property_group>
 </instance>
</service>

```

```

 </instance>
 </service>
 <service version="1" type="service" name="system/fm/asr-notify">
 <instance enabled="true" name="default">
 <property_group type="application" name="autoreg">
 <propval type="astring" name="user"
value="anonymous@oracle.com"/>
 <propval type="astring" name="password" value=""/>
 <propval type="astring" name="index" value=""/>
 <propval type="astring" name="private-key" value=""/>
 <propval type="astring" name="public-key" value=""/>
 <propval type="astring" name="client-id" value=""/>
 <propval type="astring" name="timestamp" value=""/>
 <propval type="astring" name="proxy-host" value=""/>
 <propval type="astring" name="proxy-user" value=""/>
 <propval type="astring" name="proxy-password" value=""/>
 <propval type="astring" name="hub-endpoint" value=""/>
 </property_group>
 </instance>
 </service>
</service_bundle>
root@s11-server1:/#

```

16. Add the system configuration profile manifest `custom_ai` service. Then show the results.

```

root@s11-server1:/# installadm create-profile -n custom_ai
-f /var/tmp/manifests/s11-sandbox_profile.xml -p s11-sandbox_profile
-C /var/tmp/manifests/criteria_custom_ai.xml
Profile s11-sandbox_profile added to database.
root@s11-server1:/# installadm list -p -n custom_ai
Service/Profile Name Criteria

custom_ai
 s11-sandbox_profile mac = 08:00:27:85:C7:D6

root@s11-server1:/#

```

17. Validate the system configuration profile.

```

root@s11-server1:/# installadm validate -n custom_ai -p s11-
sandbox_profile
Validating static profile s11-sandbox_profile...
Passed
root@s11-server1:/#

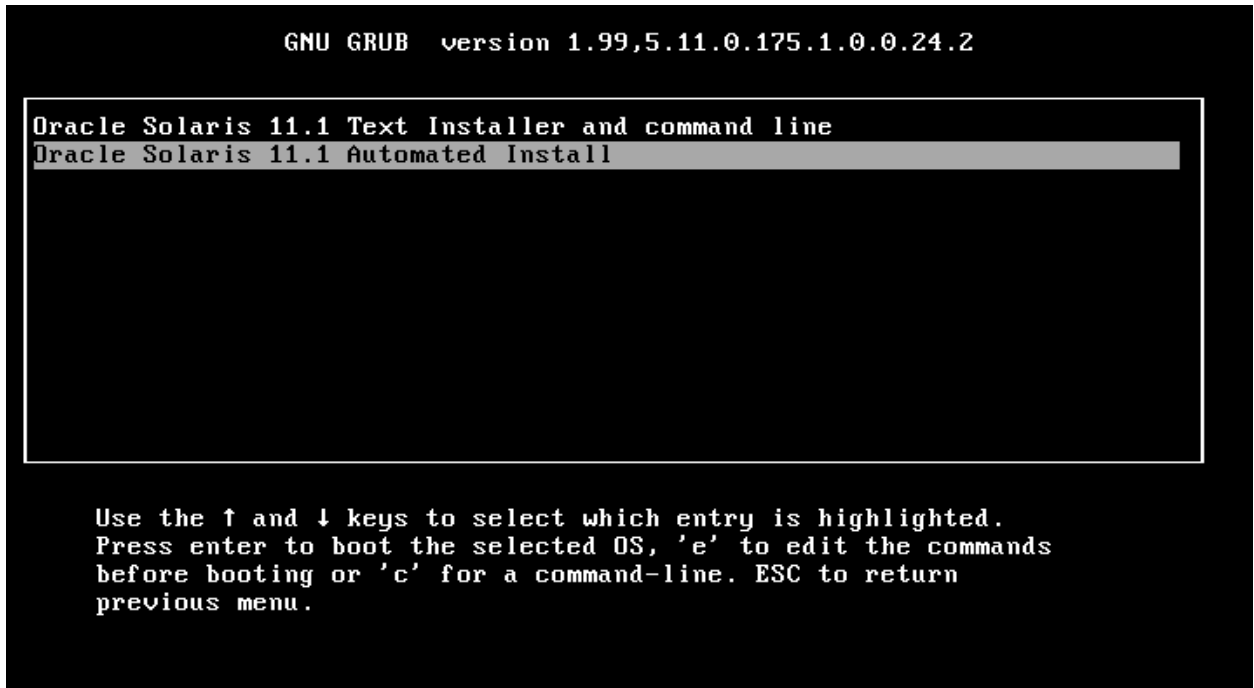
```

**Summary:** You successfully configured the AI server to automatically install an Oracle Solaris 11.1 client using the AI custom system configuration profile.

### Task 3: Deploying the OS to a Network Client

After completing the AI server configuration, you now test your work by deploying the Oracle Solaris 11.1 operating system to a network client.

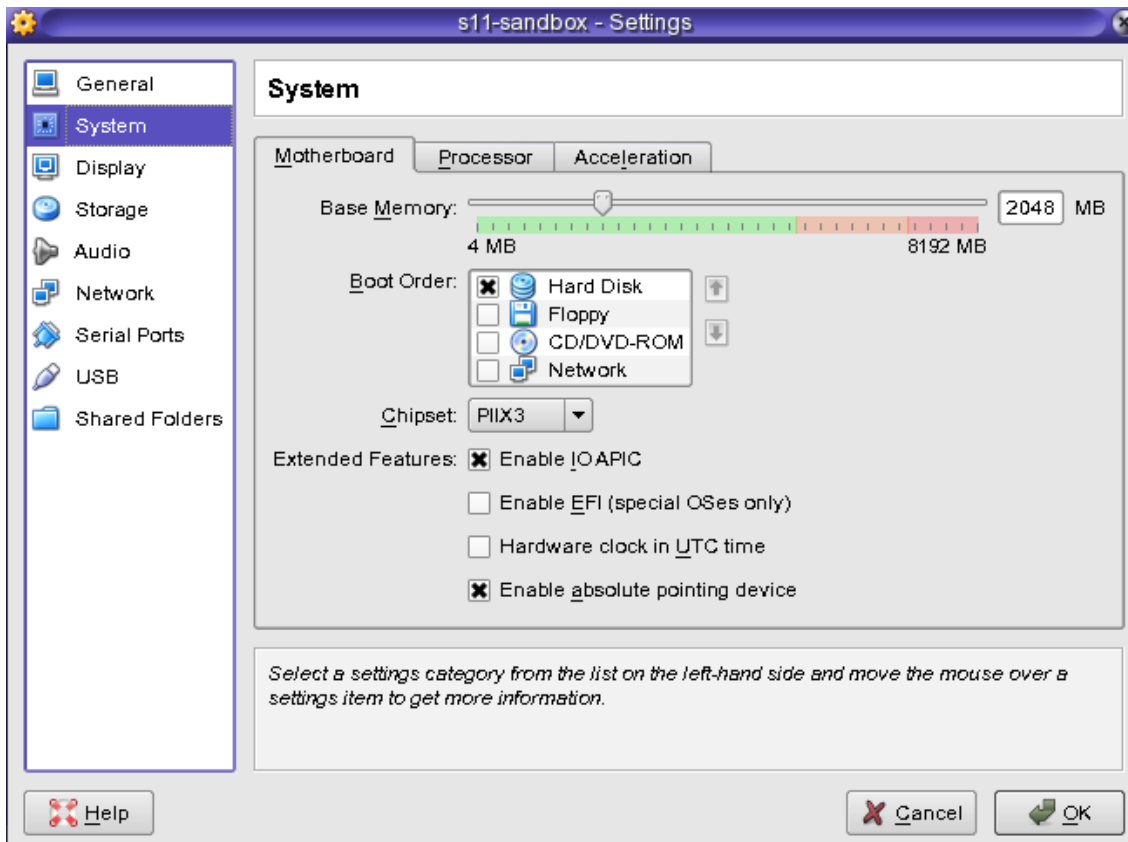
1. Verify that the **s11-server1** VM is running.
2. In VirtualBox Manager, start the **s11-sandbox** VM. If the AI server is configured correctly, you should see the OS installation begin in the VM.  
**Note:** Perform the next step as soon as possible.
3. When the **s11-sandbox** system starts the GNU GRUB menu, select the **Oracle Solaris 11.1 Automated Install** boot option.



#### Notes

- When you choose this boot option, the interactive system configuration is not available during this OS installation. Instead, IPS is used during the OS installation.
- The message traffic indicates that the IPS server is providing the installation package. When the AI installation completes, you should see messages similar to these.
- The installation takes some time to complete.

- After the OS installation is complete, reboot from the hard disk and log in as `oracle` with `oracle1` as the password.



- Check the system configuration to verify that the OS is configured according to the profile.

```
s11-sandbox console login: oracle
Password:
Last login: Sat Nov 9 18:36:14 on console
Oracle Corporation SunOS 5.11 11.1 September 2012
oracle@s11-sandbox:~$ hostname
s11-sandbox
oracle@s11-sandbox:~$ cat /etc/release
 Oracle Solaris 11.1 X86
 Copyright (c) 1983, 2012, Oracle and/or its affiliates. All rights reserved.
 Assembled 19 September 2012
oracle@s11-sandbox:~$ ipadm show-addr
ADDROBJ TYPE STATE ADDR
lo0/v4 static ok 127.0.0.1/8
net0/v4 static ok 192.168.0.130/24
lo0/v6 static ok ::1/128
net0/v6 addrconf ok fe80::a00:27ff:fe85:c7d6/10
oracle@s11-sandbox:~$
```

- Shut down and power off the s11-sandbox VM.

**Summary:** You successfully deployed the Oracle Solaris 11.1 OS on a client machine s11-sandbox using AI, and you verified its successful installation.





# **Practices for Lesson 9: Monitoring System Resources**

## **Chapter 9**

## Practices for Lesson 9: Overview

---

### Practices Overview

The fundamental framework of the testbed infrastructure is nearly complete. In the current environment, no system or resource issues need to be diagnosed. However, as the framework is deployed for validation and tuning applications, various issues will eventually develop and need to be addressed.

Now is a good time to explore the tools and utilities that Oracle Solaris 11.1 supports to monitor system resources (for example, system memory, CPU time, and data storage).

In this practice, you perform the following tasks:

- Evaluate system performance levels by using monitoring utilities.
- Monitor zone resource utilization by using the `zonestat` utility.
- Analyze applications by using DTrace one-liners and the DTrace Toolkit.

### Notes

- Command output or values may vary across systems.
- To accommodate complete command output, the font size of the output is reduced in a few places.
- Your system performance depends on the network speed and network load. If you find your VM too slow to proceed with, it is suggested that you restart the VM.

## Practice 9-1: Evaluating System Performance Levels

---

### Overview

Evaluating the system for memory, CPU, and disk usage is central to ensuring that the system performs at its best. Oracle Solaris 11.1 provides utilities to help you monitor your system resources and make informed decisions to address problems.

In this practice, you perform the following tasks:

- Display virtual memory statistics by using the `vmstat` utility.
- Display disk usage information.
- Monitor system activities.
- Collect system activity data automatically by using the `sar` command.

### Task 1: Display Virtual Memory Statistics

To obtain virtual memory statistics and information about system events (such as CPU load, paging, number of context switches, device interrupts, and system calls), use the `vmstat` command. You can also use `vmstat` to display statistics on swapping, cache flushing, and interrupts. In this task, you focus on using `vmstat` to display the following:

- Virtual memory statistics (`vmstat`)
  - System event information (`vmstat -s`)
  - Swapping statistics (`vmstat -S`)
1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1`.
  2. Right-click the desktop background and open a terminal window.
  3. In the terminal window, run the `su -` command to assume primary administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation SunOS 5.11 11.1 September 2012
root@s11-testbed:~#
```

4. Display the current virtual memory statistics by using the `vmstat` command.

```
root@s11-testbed:~# vmstat 5
```

| kthr |   |   | memory  |        |    | page |    |    |    | disk |    |    |    | faults |    |     | cpu  |      |    |    |    |
|------|---|---|---------|--------|----|------|----|----|----|------|----|----|----|--------|----|-----|------|------|----|----|----|
| r    | b | w | swap    | free   | re | mf   | pi | po | fr | de   | sr | s0 | s1 | s2     | s3 | in  | sy   | cs   | us | sy | id |
| 0    | 0 | 0 | 1113060 | 549572 | 6  | 35   | 0  | 0  | 0  | 0    | 1  | 1  | 0  | 0      | 0  | 743 | 3730 | 1694 | 1  | 5  | 94 |
| 0    | 0 | 0 | 1106844 | 529268 | 8  | 11   | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 737 | 3814 | 1843 | 1  | 6  | 93 |
| 0    | 0 | 0 | 1106844 | 529268 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 14 | 0  | 0      | 0  | 703 | 3440 | 1690 | 1  | 8  | 91 |
| 0    | 0 | 0 | 1106844 | 529268 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 27 | 0  | 0      | 0  | 742 | 3555 | 1766 | 1  | 6  | 93 |
| 0    | 0 | 0 | 1106844 | 529268 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 748 | 3719 | 1789 | 1  | 6  | 93 |
| 0    | 0 | 0 | 1106844 | 529264 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 726 | 3503 | 1708 | 1  | 7  | 92 |
| 0    | 0 | 0 | 1106844 | 529260 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 738 | 3831 | 1858 | 1  | 7  | 92 |
| 0    | 0 | 0 | 1106836 | 529260 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 923 | 3746 | 1720 | 1  | 8  | 91 |
| 0    | 0 | 0 | 1106808 | 529260 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 746 | 3612 | 1769 | 1  | 6  | 93 |
| 0    | 0 | 0 | 1106808 | 529260 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 28 | 0  | 0      | 0  | 775 | 3772 | 1843 | 1  | 5  | 94 |
| 0    | 0 | 0 | 1106808 | 529260 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 25 | 0  | 0      | 0  | 786 | 3966 | 1821 | 1  | 6  | 93 |
| 0    | 0 | 0 | 1106808 | 529260 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 17 | 0  | 0      | 0  | 810 | 3726 | 1638 | 1  | 7  | 92 |
| 0    | 0 | 0 | 1106808 | 529244 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 16 | 0  | 0      | 0  | 817 | 4469 | 2026 | 1  | 6  | 93 |
| 0    | 0 | 0 | 1106808 | 529244 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 745 | 3871 | 1857 | 1  | 6  | 93 |
| 0    | 0 | 0 | 1106808 | 529240 | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 15 | 0  | 0      | 0  | 721 | 3473 | 1707 | 1  | 6  | 94 |

```
^C
root@s11-testbed:~#
```

**CPU Load**

**us:** % of CPU time spent in user mode (1)

**sy:** % of CPU time spent in system mode (6)

**id:** % of CPU time when CPUs is idle (94)

5. Generate activity on the system by writing zeros to `/dev/null`.

```
root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[1] 5823
root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[2] 5824
root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[3] 5825
root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[4] 5826
root@s11-testbed:~#
```

## 6. Display the virtual memory statistics.

```

root@s11-testbed:~# vmstat 5
 kthr memory page disk faults cpu
 r b w swap free re mf pi po fr de sr s0 s1 s2 s3 in sy cs us sy
id
 0 0 0 1113048 549528 6 35 0 0 0 0 1 1 0 0 0 743 4456 1694 1 5
94
 8 0 0 1106064 528380 8 12 0 0 0 0 0 15 0 0 0 757 1540241 2086 58 42
0
 3 0 0 1106064 528368 0 0 0 0 0 0 0 16 0 0 0 738 1514733 1679 58 42
0
 6 0 0 1106064 528364 0 0 0 0 0 0 0 16 0 0 0 782 1447722 1843 57 43
0
 2 0 0 1106064 528364 0 5 0 0 0 0 0 15 0 0 0 832 1479844 1799 58 42
0
 5 0 0 1105372 527736 0 0 0 0 0 0 0 20 0 0 0 1017 1441694 1665 56 44
0
 5 0 0 1105936 528340 0 0 0 0 0 0 0 20 0 0 0 863 1471304 1895 57 43
0
 4 0 0 1105936 528340 0 0 0 0 0 0 0 18 0 0 0 768 1527266 1464 58 42
0
 5 0 0 1105936 528340 0 0 0 0 0 0 0 27 0 0 0 774 1560088 1482 59 41
0
 16 0 0 1105936 528336 0 0 0 0 0 0 0 26 0 0 0 828 1542343 1786 59 41
0
 9 0 0 1105936 528316 0 0 0 0 0 0 0 20 0 0 0 789 1590199 1929 59 41
0
 4 0 0 1105936 528372 0 0 0 0 0 0 0 24 0 0 0 710 1602377 1427 60 40
0
 4 0 0 1105936 528372 0 0 0 0 0 0 0 20 0 0 0 812 1543650 2127 58 42
0
^C
root@s11-testbed:~#
CPU Load
us: % of CPU time spent in user mode (58)
sy: % of CPU time spent in system mode (42)
id: % of CPU time when CPUs is idle (0)

```

7. Display the system events since the last reboot by using the `vmstat -s` command.

```

root@s11-testbed:~# vmstat -s | more
 0 swap ins
 0 swap outs
 0 pages swapped in
 0 pages swapped out
2182144 total address trans. faults taken
 3 page ins
 0 page outs
 3 pages paged in
 0 pages paged out
356073 total reclaims
356073 reclaims from free list

```

```

 0 micro (hat) faults
2182144 minor (as) faults
 3 major faults
284171 copy-on-write faults
984348 zero fill page faults
 68322 pages examined by the clock daemon
 0 revolutions of the clock hand
 0 pages freed by the clock daemon
2339 forks
 3595 vforks
 5883 execs
104981645 cpu context switches
 46071066 device interrupts
 2598544 traps
639768105 system calls
14523247 total name lookups (cache hits 92%)
 56691 user cpu
 814134 system cpu
31796395 idle cpu
 0 wait cpu
root@s11-testbed:~#

```

### Note

a. **356073 reclaims from free list:** Displays how many free pages of memory were reclaimed, which indicates how quickly the system was running out of memory. The memory is being used for programs, and this explains the load on the system memory.

b. **2339 forks:** Indicates how many processes are launching subprocesses. These processes create the workload that requires memory and CPU resources.

8. Display system memory pages swapping in and swapping out by using the `vmstat -S` command.

```

root@s11-testbed:~# vmstat -S
kthr memory page disk faults cpu
 r b w swap free si so pi po fr de sr s0 s1 s2 s3 in sy cs us sy id
 0 0 0 1113012 549428 0 0 0 0 0 0 1 2 0 0 0 744 11806 1693 1 5 93
root@s11-testbed:~#

```

Here you can check the swapping activity, for example, memory pages swapped in (`si`) and pages swapped out (`so`).

## 9. Terminate the dd processes.

```
root@s11-testbed:~# pkill dd
[1] Terminated dd if=/dev/zero of=/dev/null
[2] Terminated dd if=/dev/zero of=/dev/null
[3]- Terminated dd if=/dev/zero of=/dev/null
[4]+ Terminated dd if=/dev/zero of=/dev/null
root@s11-testbed:~#
```

## Task 2: Display Disk Usage Information

To monitor disk usage, use the `iostat` command in both normal and extended format.

To display disk space information, use the `df` command.

In this task, you do the following:

- Display general disk usage data (`iostat`).
  - Display extended disk statistics (`iostat -xtc`).
  - Display disk space information (`df -h`).
1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1`.
  2. Right-click the desktop background and open a terminal window.
  3. In the terminal window, use the `su` command to assume primary administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation SunOS 5.11 11.1 September 2012
root@s11-testbed:~#
```

4. Check the input/output activity on your disks and CPU by using the `iostat` command.

```
root@s11-testbed:~# iostat 5
```

| tty |      | sd0 |     |      | sd1 |     |      | sd2 |     |      | sd3 |     |      | cpu |    |    |    |
|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|----|----|----|
| tin | tout | kps | tps | serv | kps | tps | serv | kps | tps | serv | kps | tps | serv | us  | sy | wt | id |
| 0   | 5    | 13  | 2   | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 1   | 5  | 0  | 93 |
| 0   | 47   | 237 | 26  | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 1   | 3  | 0  | 97 |
| 0   | 16   | 38  | 15  | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 1   | 4  | 0  | 95 |
| 0   | 16   | 39  | 15  | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 0   | 3  | 0  | 97 |
| 0   | 16   | 239 | 27  | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 0   | 0   | 0    | 1   | 3  | 0  | 97 |

```
^C
root@s11-testbed:~#
```

**Note**

| Device Type | Field Name  | Description                                       |
|-------------|-------------|---------------------------------------------------|
| Terminal    | Device Type |                                                   |
|             | tin         | Number of characters in the terminal input queue  |
|             | tout        | Number of characters in the terminal output queue |
| Disk        | Device Type |                                                   |
|             | kps         | Blocks per second                                 |
|             | tps         | Transactions per second                           |
|             | serv        | Average service time (in milliseconds)            |
| CPU         | Device Type |                                                   |
|             | us          | In user mode                                      |
|             | sy          | In system mode                                    |
|             | wt          | Waiting for I/O                                   |
|             | id          | Idle                                              |

5. Obtain extended input/output statistics for each disk by using the `iostat -xtc` command.

```
root@s11-testbed:~# iostat -xtc
```

| extended device statistics |     |     |      |      |      |      |       |    |    | tty |      | cpu |    |    |    |
|----------------------------|-----|-----|------|------|------|------|-------|----|----|-----|------|-----|----|----|----|
| device                     | r/s | w/s | kr/s | kw/s | wait | actv | svc_t | %w | %b | tin | tout | us  | sy | wt | id |
| sd0                        | 0.8 | 0.7 | 7.9  | 5.4  | 0.0  | 0.0  | 0.3   | 0  | 0  | 0   | 5    | 1   | 5  | 0  | 93 |
| sd1                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.0   | 0  | 0  |     |      |     |    |    |    |
| sd2                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.1   | 0  | 0  |     |      |     |    |    |    |
| sd3                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.1   | 0  | 0  |     |      |     |    |    |    |
| sd4                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.1   | 0  | 0  |     |      |     |    |    |    |
| sd5                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.1   | 0  | 0  |     |      |     |    |    |    |
| sd6                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.1   | 0  | 0  |     |      |     |    |    |    |
| sd7                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.1   | 0  | 0  |     |      |     |    |    |    |
| sd8                        | 0.1 | 0.1 | 1.0  | 0.9  | 0.0  | 0.0  | 0.5   | 0  | 0  |     |      |     |    |    |    |
| sd9                        | 0.0 | 0.0 | 0.0  | 0.0  | 0.0  | 0.0  | 0.0   | 0  | 0  |     |      |     |    |    |    |

```
root@s11-testbed:~#
```

For example, consider the reads and writes for the `sd0` disk: 7.9 kilobytes worth of data read per second and 5.4 kilobytes worth of data written per second.

The `svc_t` column shows the service time in milliseconds. Look at the average service time of 0.3 milliseconds for the `sd0` disk. Compare this disk to the other disks. Why is its service time so high? *In the current environment, the default ZFS file system is on this disk.*

6. Display the amount of space that is available and used on the currently mounted file systems by using the `df` command.

```
root@s11-testbed:~# df -h | more
```

| Filesystem         | Size | Used | Available | Capacity | Mounted on       |
|--------------------|------|------|-----------|----------|------------------|
| rpool/ROOT/solaris | 39G  | 3.8G | 31G       | 11%      | /                |
| /devices           | 0K   | 0K   | 0K        | 0%       | /devices         |
| /dev               | 0K   | 0K   | 0K        | 0%       | /dev             |
| ctfs               | 0K   | 0K   | 0K        | 0%       | /system/contract |
| proc               | 0K   | 0K   | 0K        | 0%       | /proc            |
| mnttab             | 0K   | 0K   | 0K        | 0%       | /etc/mnttab      |



```

swap 1.1G 1.6M 1.1G 1% /system/volatile
objfs 0K 0K 0K 0% /system/object
sharefs 0K 0K 0K 0% /etc/dfs/sharetab
/usr/lib/libc/libc_hwcapi.so.1 35G 3.8G 31G 11%
/lib/libc.so.1
fd 0K 0K 0K 0% /dev/fd
rpool/ROOT/solaris/var 39G 604M 31G 2% /var
swap 1.1G 64M 1.1G 6% /tmp
rpool/VARSHARE 39G 120K 31G 1% /var/share
ora 426G 122G 304G 29% /opt/ora
--More--

```

This command is very useful because it displays the used and available storage information for all mounted file systems. In this example, the ZFS root file system has used 3.8 GB of a total of 39 GB.

### Task 3: Monitor System Activities

The `sar` (system activity reporter) utility helps you monitor system activities. In this task, you focus on using `sar` to monitor the following:

- File access (`sar -a`)
- Buffer activity (`sar -b`)
- System call statistics (`sar -c`)
- Disk activity (`sar -d`)
- Unused memory (`sar -r`)

You conclude by scheduling automatic data collection by modifying the `crontab` file.

1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1`.
2. Right-click the desktop background and open a terminal window.
3. In the terminal window, run the `su` command to assume primary administrator privileges.

```

oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation SunOS 5.11 11.1 September 2012
root@s11-testbed:~#

```

4. Check file access by using the `sar -a` command.

```

root@s11-testbed:~# sar -a 5 2

SunOS s11-testbed 5.11 11.1 i86pc 11/11/2013

20:40:21 iget/s namei/s dirbk/s
20:40:26 0 25 0
20:40:31 0 26 0

Average 0 25 0
root@s11-testbed:~#

```

You ran the command for two displays every five seconds. On average in the output, the system could not find many files (under column `namei/s`) that were being accessed. At the production system level, there may be problems if this number is high.

### Notes

- **iget/s**: The number of requests made for inodes that were not in the directory name look-up cache (DNLC)
- **namei/s**: The number of file system path searches per second. If `namei` does not find a directory name in the DNLC, it calls `iget` to get the inode for either a file or a directory. Therefore, most `igets` are the result of DNLC misses.
- **dirbk/s**: The number of directory block reads issued per second

### 5. Check buffer activity by using the `sar -b` command.

```
root@s11-testbed:~# sar -b 2 2
```

```
SunOS s11-testbed 5.11 11.1 i86pc 11/11/2013
```

```
20:41:42 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
20:41:44 0 0 100 0 0 100 0 0
20:41:46 0 0 100 0 0 100 0 0

Average 0 0 100 0 0 100 0 0
root@s11-testbed:~#
```

This command displays the reads from the buffer and the writes to the buffer. You can see 100% reads from the buffer and 100% writes to the buffer. You might be looking for particular issues. However, operations here are running smoothly in terms of buffer activity.

The following table describes the buffer activities that are displayed by the `-b` option.

| Field Name           | Description                                                                                                                            |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <code>bread/s</code> | Average number of reads per second that are submitted to the buffer cache from the disk                                                |
| <code>lread/s</code> | Average number of logical reads per second from the buffer cache                                                                       |
| <code>%rcache</code> | Fraction of logical reads that are found in the buffer cache (100 % minus the ratio of <code>bread/s</code> to <code>lread/s</code> )  |
| <code>bwrit/s</code> | Average number of physical blocks (512 bytes) that are written per second from the buffer cache to disk                                |
| <code>lwrit/s</code> | Average number of logical writes per second to the buffer cache                                                                        |
| <code>%wcache</code> | Fraction of logical writes that are found in the buffer cache (100 % minus the ratio of <code>bwrit/s</code> to <code>lwrit/s</code> ) |
| <code>pread/s</code> | Average number of physical reads, per second, that use character device interfaces                                                     |
| <code>pwrit/s</code> | Average number of physical write requests per second that use character device interfaces                                              |

## 6. Check system call activity by using the `sar -c` command.

```
root@s11-testbed:~# sar -c 2 2
```

```
SunOS s11-testbed 5.11 11.1 i86pc 11/11/2013
```

```
20:42:20 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
20:42:22 1234 46 47 0.00 0.00 10007 9763
20:42:24 1240 47 44 0.00 0.00 9583 9759

Average 1237 47 46 0.00 0.00 9796 9761
root@s11-testbed:~#
```

This command displays system calls for reads, writes, forks, and other system call information. This information is useful when you are developing metrics or tracking down a high number of system calls by using `dtrace`.

The following table describes the system call categories that are reported by the `-c` option.

| Field Name | Description                                                                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| scall/s    | The number of all types of system calls per second, which is usually about 30 per second on a system with four to six users                                                         |
| sread/s    | The number of <code>read</code> system calls per second                                                                                                                             |
| swrit/s    | The number of <code>write</code> system calls per second                                                                                                                            |
| fork/s     | The number of <code>fork</code> system calls per second, which is about 0.5 per second on a system with four to six users. This number increases if shell scripts are running.      |
| exec/s     | The number of <code>exec</code> system calls per second. If <code>exec/s</code> divided by <code>fork/s</code> is greater than 3, look for inefficient <code>PATH</code> variables. |
| rchar/s    | The number of characters (bytes) transferred per second by <code>read</code> system calls                                                                                           |
| wchar/s    | The number of characters (bytes) transferred per second by <code>write</code> system calls                                                                                          |

## 7. Check disk activity by using the `sar -d` command.

```
root@s11-testbed:~# sar -d 2 2
```

```
SunOS s11-testbed 5.11 11.1 i86pc 11/11/2013
```

```
20:43:37 device %busy avque r+w/s blks/s await avserv
20:43:39 ahci0 0 0.0 0 0 0.0 0.0
 iscsi0 0 0.0 0 0 0.0 0.0
 scsi_vhc 0 0.0 0 0 0.0 0.0
 sd0 0 0.0 42 188 0.0 0.1
 sd0,a 0 0.0 1 0 0.0 0.0
 sd0,b 0 0.0 41 188 0.0 0.1
 . . .
 . . .
Average ahci0 0 0.0 0 0 0.0 0.0
 iscsi0 0 0.0 0 0 0.0 0.0
 scsi_vhc 0 0.0 0 0 0.0 0.0
```

|       |   |     |    |    |     |     |
|-------|---|-----|----|----|-----|-----|
| sd0   | 0 | 0.0 | 21 | 94 | 0.0 | 0.1 |
| sd0,a | 0 | 0.0 | 1  | 0  | 0.0 | 0.0 |
| sd0,b | 0 | 0.0 | 20 | 94 | 0.0 | 0.1 |
| sd0,h | 0 | 0.0 | 0  | 0  | 0.0 | 0.0 |
| sd0,i | 0 | 0.0 | 0  | 0  | 0.0 | 0.0 |
| sd0,q | 0 | 0.0 | 0  | 0  | 0.0 | 0.0 |
| . . . |   |     |    |    |     |     |
| . . . |   |     |    |    |     |     |

root@s11-testbed:~#

This command displays disk-related activity: for example, reads and writes (as shown in the `r+w/s` column), average wait time, and average service time in milliseconds. What do the numbers indicate? *If any of these numbers are too high for your application, there may be a disk issue.*

**Note:** The following table describes the disk device activities that are reported by the `-d` option.

| Field Name | Description                                                                                                                                                                        |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| device     | Name of the disk device that is being monitored                                                                                                                                    |
| %busy      | Portion of time the device was busy servicing a transfer request                                                                                                                   |
| avque      | Average number of requests during the time the device was busy servicing a transfer request                                                                                        |
| r+w/s      | Number of read-and-write transfers to the device per second                                                                                                                        |
| blks/s     | Number of 512-byte blocks that are transferred to the device per second                                                                                                            |
| avwait     | Average time, in milliseconds, that transfer requests wait idly in the queue. This time is measured only when the queue is occupied.                                               |
| avserv     | Average time, in milliseconds, for a transfer request to be completed by the device. For disks, this value includes seek times, rotational latency times, and data transfer times. |

8. Check available physical and swap memory by using the `sar -r 2 2` command.

|                                              |
|----------------------------------------------|
| root@s11-testbed:~# <code>sar -r 2 2</code>  |
| SunOS s11-testbed 5.11 11.1 i86pc 11/11/2013 |
| 20:45:04 freemem freeswap                    |
| 20:45:06 132839 2218328                      |
| 20:45:08 132839 2218328                      |
| Average 132839 2218328                       |
| root@s11-testbed:~#                          |

This command displays the available physical and swap memory. The benefit of tracking these numbers is that it helps you take corrective action if memory is running out. For example, if very little swap memory is left, you can increase the swap memory allocation.

### Notes

- `freemem`: The average number of memory pages that are available to user processes over the intervals sampled by the command. Page size is machine-dependent.
- `freeswap`: The number of 512-byte disk blocks that are available for page swapping

9. Exit the terminal.

## Practice 9-2: Monitoring Zone Resource Utilization

### Overview

Oracle Solaris 11 provides a powerful new zone-monitoring utility called `zonestat`. The `zonestat` utility reports on the CPU, memory, and resource utilization of currently running zones. Each zone's utilization is reported as a percentage of both system resources and the zone's configured limits. The `zonestat` utility prints a series of reports and one or more summary reports at specified intervals.

### Task: Monitor Zone Resource Utilization

Perform the following steps in the **s11-testbed** VM:

1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1`.
2. Run the `su` command to assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation SunOS 5.11 11.1 September 2012
root@s11-testbed:~#
```

3. Display a summary of memory utilization every five seconds by using the `zonestat` utility.

```
root@s11-testbed:~# zonestat -r physical-memory 5
Collecting data for first interval...
Interval: 1, Duration: 0:00:05
PHYSICAL-MEMORY SYSTEM MEMORY
mem_default 2559M

 ZONE USED %USED CAP %CAP
 [total] 1153M 45.0% - -
 [system] 426M 16.6% - -
 global 640M 25.0% - -
 zgateway 54.7M 2.13% - -
 dbzone 32.3M 1.26% - -

Interval: 2, Duration: 0:00:10
PHYSICAL-MEMORY SYSTEM MEMORY
mem_default 2559M

 ZONE USED %USED CAP %CAP
 [total] 1154M 45.0% - -
 [system] 426M 16.6% - -
 global 640M 25.0% - -
 zgateway 54.7M 2.13% - -
 dbzone 32.3M 1.26% - -

Interval: 3, Duration: 0:00:15
PHYSICAL-MEMORY SYSTEM MEMORY
mem_default 2559M

 ZONE USED %USED CAP %CAP
 [total] 1154M 45.0% - -
```

```

[system] 426M 16.6% - -
global 640M 25.0% - -
zgateway 54.7M 2.13% - -
dbzone 32.3M 1.26% - -

```

```
^Croot@s11-testbed:~#
```

```
...
```

Press Ctrl + C to escape the zonestat command.

**Note:** Command output may vary across systems.

4. Report on the default processor set (pset) once a second for one minute.

```

root@s11-testbed:~# zonestat -r default-pset 1 1m
Collecting data for first interval...
Interval: 1, Duration: 0:00:01
PROCESSOR_SET TYPE ONLINE/CPUS MIN/MAX
pset_default default-pset 2/2 2/2
 ZONE USED %USED CAP %CAP SHRS %SHR %SHRU
 [total] 0.34 17.3% - - - - -
 [system] 0.09 4.78% - - - - -
 global 0.24 12.0% - - - - -
 dbzone 0.00 0.14% - - - - -
 zgateway 0.00 0.38% - - - - -

Interval: 2, Duration: 0:00:02
PROCESSOR_SET TYPE ONLINE/CPUS MIN/MAX
pset_default default-pset 2/2 2/2
 ZONE USED %USED CAP %CAP SHRS %SHR %SHRU
 [total] 0.30 15.3% - - - - -
 [system] 0.00 0.00% - - - - -
 global 0.29 14.9% - - - - -
 dbzone 0.00 0.18% - - - - -
 zgateway 0.00 0.21% - - - - -

...

```

5. Monitor silently at 10-second intervals for one minute. Then produce a report showing both total utilization and high utilization.

**Note:** To view the output, you need to wait for a minute.

```

root@s11-testbed:~# zonestat -q -R total,high 10s 1m

Report: Total Usage
 Start: Mon Nov 11 06:04:44 UTC 2013
 End: Mon Nov 11 06:05:44 UTC 2013
 Intervals: 6, Duration: 0:01:00
SUMMARY
 Cpus/Online: 2/2 PhysMem: 2559M VirtMem: 3583M
 ---CPU--- --PhysMem-- --VirtMem-- --PhysNet--
 ZONE USED %PART USED %USED USED %USED PBYTE %PUSE
 [total] 0.15 7.70% 1154M 45.0% 1658M 46.2% 6592 0.00%
 [system] 0.05 2.62% 426M 16.6% 1014M 28.3% - -

```

```

 global 0.09 4.94% 640M 25.0% 565M 15.7% 577K 0.00%
 dbzone 0.00 0.02% 32.3M 1.26% 32.9M 0.92% 0 0.00%
 zgateway 0.00 0.10% 54.7M 2.13% 45.0M 1.25% 103K 0.00%

Report: High Usage
 Start: Mon Nov 11 06:04:44 UTC 2013
 End: Mon Nov 11 06:05:44 UTC 2013
 Intervals: 6, Duration: 0:01:00
SUMMARY Cpus/Online: 2/2 PhysMem: 2559M VirtMem: 3583M
 ---CPU--- --PhysMem-- --VirtMem-- --PhysNet--
 ZONE USED %PART USED %USED USED %USED PBYTE %PUSE
[total] 0.16 8.46% 1154M 45.0% 1658M 46.2% 728K 0.00%
[system] 0.06 3.19% 426M 16.6% 1014M 28.3% - -
 global 0.09 4.87% 640M 25.0% 565M 15.7% 596K 0.00%
 dbzone 0.00 0.02% 32.3M 1.26% 32.9M 0.92% 0 0.00%
 zgateway 0.00 0.12% 54.7M 2.13% 45.0M 1.25% 104K 0.00%

root@s11-testbed:~#

```

6. (Optional) Halt the zgateway and dbzone zones to release the system resources.

```

root@s11-testbed:~# zoneadm -z zgateway halt
root@s11-testbed:~# zoneadm -z dbzone halt
root@s11-testbed:~# zoneadm list -cv

```

| ID | NAME     | STATUS    | PATH             | BRAND     | IP     |
|----|----------|-----------|------------------|-----------|--------|
| 0  | global   | running   | /                | solaris   | shared |
| -  | dbzone   | installed | /muzones/dbzone  | solaris10 | excl   |
| -  | zgateway | installed | /muzones/zgatway | solaris   | excl   |



## Practice 9-3: Analyzing Applications at Run Time by Using DTrace One-Liners

### Overview

DTrace is an observability technology that helps you examine the behavior of user programs and applications as well as the OS in development and in production. DTrace does the following:

- Examines the entire software stack
- Determines the root cause of performance problems
- Tracks the source of aberrant behavior

In this practice, you work with DTrace one-liners.

### Notes

1. Although the printed output from the DTrace commands in this practice are examples, your own output should be similar.
2. When executing the DTrace one-liners in this practice, let them run for several seconds to enable DTrace to collect a good data sample. Press Ctrl + C to terminate DTrace.

### Tasks

Perform the following steps to practice using DTrace one-liners:

1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1`.
2. Run the `su` command to assume primary administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation SunOS 5.11 11.1 September 2012
root@s11-testbed:~#
```

3. Create a list of DTrace providers.

```
root@s11-testbed:~# dtrace -l | perl -pe 's/^\s*\s+(\S+?)'
([0-9]|\s).*\/\1/' | sort | uniq
39125 fbt cpu_ms.GenuineIntel
gintel_gentopo_ereport_create_resource_elem entry
39126 fbt cpu_ms.GenuineIntel
gintel_gentopo_ereport_create_resource_elem return
39127 fbt cpu_ms.GenuineIntel gintel_ereport_create_resource_elem entry
39128 fbt cpu_ms.GenuineIntel gintel_ereport_create_resource_elem return
39129 fbt cpu_ms.GenuineIntel
nehalem_ep_ereport_add_memory_error_counter entry
39130 fbt cpu_ms.GenuineIntel
nehalem_ep_ereport_add_memory_error_counter return
PROVIDER
Xserver1401
cpu_ms.GenuineIntel
dtrace
fbt
fsinfo
io
ip
```

```

iscsi
kerberos966
libpython2.6.so.1.0
lockstat
mib
nfsmapi1311
nfsv4
pcbe.GenuineIntel.6
perl5800
proc
profile
sched
scsi_vhci_f_asym_emc
scsi_vhci_f_asym_lsi
scsi_vhci_f_asym_sun
scsi_vhci_f_sym_emc
scsi_vhci_f_sym_hds
sdt
shadowfs
syscall
sysevent
sysinfo
tcp
udp
vm
vminfo
root@s11-testbed:~#

```

4. Generate activity on the system by writing zeros to /dev/null.

```

root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[1] 5823
root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[2] 5824
root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[3] 5825
root@s11-testbed:~# dd if=/dev/zero of=/dev/null &
[4] 5826
root@s11-testbed:~#

```

5. Determine the number of system calls that are executed by the CPUs.

```

root@s11-testbed:~# dtrace -n 'syscall:::entry { @[probefunc] = count
(); }'

dtrace: description 'syscall:::entry ' matched 213 probes
^C

fcntl

```

1

```

mmap 1
schedctl 1
sigpending 1
fstatat64 2
getdents64 2
mkdirat 2
modctl 2
setgid 2
statvfs 2
lseek 3
setcontext 3
sysconfig 3
waitsys 3
getmsg 4
putmsg 4
setuid 4
sigaction 4
brk 6
getpid 9
close 11
openat 11
lwp_sigmask 13
privsys 20
getuid 22
gtime 30
writev 34
setitimer 38
fstatat 39
doorfs 44
lwp_cond_wait 52
lwp_park 73
nanosleep 101
pollsys 108
recv 189
p_online 256
clock_gettime 260
ioctl 874
write 1131451
read 1131504
root@s11-testbed:~#

```

6. Determine the number of interrupts that are handled by each CPU.

```

root@s11-testbed:~# dtrace -n 'sdt:::interrupt-start { @num[cpu] =
count(); } '
dtrace: description 'sdt:::interrupt-start ' matched 1 probe
^C

```

```

0 1517
1 1618
root@s11-testbed:~#

```

### 7. Terminate the dd processes.

```

root@s11-testbed:~# pkill dd
[1] Terminated dd if=/dev/zero of=/dev/null
[2] Terminated dd if=/dev/zero of=/dev/null
[3] Terminated dd if=/dev/zero of=/dev/null
[4] Terminated dd if=/dev/zero of=/dev/null

```

**Note:** Command output or values may vary.

### 8. Generate disk I/O activity by using the following script:

```

root@s11-testbed:~# while [1]
> do
> tar cf /var/tmp/disk1.tar /opt/ora/lab/*
> rm /var/tmp/disk1.tar
> done&
[1] 10288

```

### 9. Determine the write size distribution by process.

```

root@s11-testbed:~# dtrace -n 'sysinfo::writech { @dist[execname] =
quantize(arg0); }'
dtrace: description 'sysinfo::writech ' matched 4 probes
^C

dtrace
value ----- Distribution ----- count
0 | 0
1 | @@ 1
2 | 0
...
tar
value ----- Distribution ----- count
1024 | 0
2048 | 1
4096 | 0
8192 | @@ 5939
16384 | 0

root@s11-testbed:~#

```

## 10. Determine the read size distribution by process.

```

root@s11-testbed:~# dtrace -n 'sysinfo:::readch { @dist[execname] =
quantize(arg0); }'
dtrace: description 'sysinfo:::readch ' matched 4 probes
^C

. . .
tar

```

| value | ----- Distribution -----                                 | count |
|-------|----------------------------------------------------------|-------|
| -1    |                                                          | 0     |
| 0     | @@                                                       | 235   |
| 1     |                                                          | 0     |
| 2     |                                                          | 0     |
| 4     |                                                          | 0     |
| 8     |                                                          | 0     |
| 16    |                                                          | 1     |
| 32    |                                                          | 4     |
| 64    |                                                          | 6     |
| 128   |                                                          | 16    |
| 256   |                                                          | 29    |
| 512   |                                                          | 26    |
| 1024  | @                                                        | 85    |
| 2048  | @                                                        | 82    |
| 4096  |                                                          | 40    |
| 8192  | @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ | 5183  |
| 16384 |                                                          | 0     |

```

root@s11-testbed:~#

```

## 11. Determine the physical I/O to ZFS pools, physical disks, disk volumes, and NFS.

```

root@s11-testbed:~# dtrace -n 'io:::start { @[pid, execname] =
sum(args[0] ->b_bcount); }'
dtrace: description 'io:::start ' matched 6 probes
^C

5 zpool-rpool 15973888
root@s11-testbed:~#

```

## 12. Determine the file I/O (measured in blocks), summarized per zone.

```

root@s11-testbed:~# dtrace -n 'io:::start { @size[zonename] =
quantize(args[0] ->b_bcount); }'
dtrace: description 'io:::start ' matched 6 probes
^C

global

```

| value | ----- Distribution ----- | count |
|-------|--------------------------|-------|
| 256   |                          | 0     |
| 512   | @@@@                     | 22    |

|         |                 |    |
|---------|-----------------|----|
| 1024    | @@@@@@@         | 38 |
| 2048    | @@@@@@@         | 42 |
| 4096    | @@@@            | 24 |
| 8192    | @@@             | 15 |
| 16384   | @               | 6  |
| 32768   |                 | 1  |
| 65536   | @@              | 9  |
| 131072  |                 | 1  |
| 262144  |                 | 2  |
| 524288  | @               | 3  |
| 1048576 | @@@@@@@@@@@@@@@ | 68 |
| 2097152 |                 | 0  |

root@s11-testbed:~#

13. Kill the disk I/O script.

```
root@s11-testbed:~# pkill bash
...
```

14. Generate network activity by running the following command:

```
root@s11-testbed:~# ping -s 192.168.0.255 > /dev/null&
[1] 6200
root@s11-testbed:~#
```

15. Determine the number of received packets by host address.

```
root@s11-testbed:~# dtrace -n 'ip:::receive { @[args[2]->ip_saddr] =
count(); }'
dtrace: description 'ip:::receive ' matched 5 probes
^C

192.168.0.121 3
192.168.0.10 36
192.168.0.30 36
192.168.0.20 42
192.168.0.100 77
root@s11-testbed:~#
```

16. Kill the ping process.

```
root@s11-testbed:~# pkill ping
[1]+ Terminated ping -s 192.168.0.255 > /dev/null
root@s11-testbed:~#
```

## Practice 9-4: Analyzing Applications at Run Time by Using the DTrace Toolkit

### Overview

In this practice, you use the DTrace Toolkit.

**Note:** Although the printed output from the commands in this practice are examples, your own output should be similar.

### Task

Perform the following steps:

1. Log in to the **s11-testbed** VM as the `oracle` user with the password `oracle1`.
2. Run the `su` command to assume administrator privileges.

```
oracle@s11-testbed:~$ su -
Password: oracle1
Oracle Corporation SunOS 5.11 11.1 September 2012
root@s11-testbed:~#
```

3. Verify the DTrace Toolkit installation.

```
root@s11-testbed:~# pkg info dtrace-toolkit
 Name: system/dtrace/dtrace-toolkit
 Summary: DTraceToolkit 0.99
 Description: The DTraceToolkit is a collection of useful, documented
 DTrace
 scripts
 Category: Development/System
 State: Installed
 Publisher: solaris
 Version: 0.99
 Build Release: 5.11
 Branch: 0.175.1.0.0.24.2
 Packaging Date: Wed Sep 19 18:47:48 2012
 Size: 3.00 MB
 FMRI: pkg://solaris/system/dtrace/dtrace-toolkit@0.99,5.11-
 0.175.1.0.0.24.2:20120919T184748Z
root@s11-testbed:~# pkg install dtrace-toolkit
No updates necessary for this image.
Planning linked: 1/1 done
root@s11-testbed:~#
```

4. Add the DTrace Toolkit to the user profile.

```
root@s11-testbed:~# pwd
/root
root@s11-testbed:~# vi .profile
...
export PATH=/usr/bin:/usr/sbin:/usr/dtrace/DTT/Bin
export MANPATH=/usr/dtrace/DTT/Man:/usr/share/man
..
```

```
:wq!
root@s11-testbed:~# . .profile
```

Verify the path by using echo command.

```
root@s11-testbed:~# echo $PATH
/usr/bin:/usr/sbin:/usr/dtrace/DTT/Bin
root@s11-testbed:~# echo $MANPATH
/usr/dtrace/DTT/Man:/usr/share/man
root@s11-testbed:~#
```

5. In the `/usr/dtrace/DTT/Docs` directory, look at the content of the `Contents` file.

```
root@s11-testbed:~# cd /usr/dtrace/DTT/Docs
root@s11-testbed:/usr/dtrace/DTT/Docs# cat Contents
Contents - Command Summary
```

The following is a list of commands found in the DTraceToolkit, along with their directory location.

Generally commands that end in a ".d" are DTrace scripts, and commands that don't are DTrace scripts wrapped in another language (eg, shell or Perl). See the Docs/Readme for instructions for finding their docs.

```
DTraceToolkit/
 dexplorer run a series of scripts and archive output
 dtruss process syscall info. DTrace truss
 dvmstat vmstat by PID/name/command
 errinfo report syscall failures with details
 execsnoop snoop process execution as it occurs
 iosnoop snoop I/O events as they occur
 iopattern print disk I/O pattern
 iotop display top disk I/O events by process
 opensnoop snoop file opens as they occur
 procsysystem analyse process system call times
 rwsnoop snoop read/write events
 rwtop display top read/write bytes by process
 statsnoop snoop file stats as they occur
Apps/
 httpdstat.d realtime httpd statistics
 nfswizard.d NFS client activity wizard
 shellsnoop snoop live shell activity
 weblatency.d website latency statistics
Cpu/
 cputypes.d list CPU types
 cpuwalk.d measure which CPUs a process runs on
 dispqlen.d dispatcher queue length by CPU
```



```

 intbycpu.d interrupts by CPU
 intoncpu.d interrput on-cpu usage
 inttimes.d interrput on-cpu time total
 loads.d print load averages
 runocc.d run queue occupancy by CPU
 xcallsbypid.d CPU cross calls by PID
Disk/
 bitesize.d print disk event size report
 diskhits disk access by file offset
 hotspot.d print disk event by location
 iofile.d I/O wait time by filename and process
 iofileb.d I/O bytes by filename and process
 iopending plot number of pending disk events
 pathopens.d pathnames successfully opened count
 seeksize.d print disk seek size report
Docs/
 oneliners.txt DTrace oneliners
FS/
 fsrw.d file system read/write event tracing
 fspaging.d file system read/write and paging tracing
 rfsio.d read FS I/O stats, with cache miss rate
 rfileio.d read file I/O stats, with cache miss rate
 vopstat vnode interface statistics
Java/
 j_*.d 18 scripts for tracing Java using the hotspot
provider
 JavaScript/
 js_*.d 14 scripts for JavaScript with the Mozilla
provider
Kernel/
 cputimes print time by Kernel/Idle/Process
 cpudists time distribution by Kernel/Idle/Process
 cswstat.d context switch time statistics
 dnlcps.d DNLC stats by process
 dnlcsnoop.d snoop DNLC activity
 dnlcstat DNLC statistics
 kstat_types.d trace kstat reads with type info
 modcalls.d kernel function calls by module name
 priclass.d priority distribution by scheduling class
 pridist.d process priority distribution
 putnexts.d trace who is putting to which streams module
 whatexec.d examine the type of files executed
Locks/
 lockbyproc.d lock time by process name
 lockbydist.d lock time distribution by process name
Mem/

```

```

anonpgpid.d anonymous memory paging info by PID on CPU
minfbypid.d minor faults by PID
minfbyproc.d minor faults by process name
pgpgginbypid.d pages paged in by PID
pgpgginbyproc.d pages paged in by process name
swapinfo.d print virtual memory info
vmbypid.d virtual memory stats by PID
vmstat.d vmstat demo using DTrace
vmstat-p.d vmstat -p demo using DTrace
xvmstat extended vmstat demo using DTrace
Misc/
 guess.d guessing game
 wpm.d words per minute tracing
 woof.d audio alert for new processes
Net/
 connections print inbound TCP connections by process
 icmpstat.d print ICMP statistics
 tcpsnoop snoop TCP network packets by process, Solaris 10 3/05
 tcpsnoop_snv snoop TCP network packets by process, Solaris
Nevada
 tcpsnoop.d snoop TCP network packets by process, Solaris 10 3/05
 tcpsnoop_snv.d snoop TCP network packets by process, Solaris
Nevada
 tcpstat.d print TCP statistics
 tcptop display top TCP network packets by PID, Solaris
10 3/05
 tcptop_snv display top TCP network packets by PID, Solaris
Nevada
 tcpwdist.d simple TCP write distribution by process
 udpstat.d print UDP statistics
Perl/
 pl_*.d 12 scripts for tracing Perl
Php/
 php_*.d 12 scripts for tracing Php
Proc/
 crash.d crashed application report
 creatbyproc.d snoop file creat() by process name
 dappprof profile user and lib function usage
 dapptrace trace user and lib function usage
 fddist file descriptor usage distribution
 fileproc.d snoop files opened by process
 kill.d snoop process signals
 lastwords print syscalls before exit
 mmapfiles.d mmap'd files by process
 newproc.d snoop new processes
 pfilestat show I/O latency break down by FD
 pidpersec.d print new PIDs per sec

```

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

```

readbytes.d read bytes by process name
readdist.d read distribution by process name
rwbbypid.d read/write bytes by PID
rwbypid.d read/write calls by PID
rwbytype.d read/write bytes by vnode type
sampleproc sample processes on the CPUs
shortlived.d check short lived process time
sigdist.d signal distribution by process name
stacksize.d measure stack size for running threads
sysbypid.d system stats by PID
syscallbyproc.d system calls by process name
syscallbypid.d system calls by process ID
threaded.d sample multi-threaded CPU usage
topsysproc display top syscalls by process name
writebytes.d write bytes by process name
writedist.d write distribution by process name

Python/
 py_*.d 14 scripts for tracing Python
Shell/
 sh_*.d 15 scripts for tracing the Bourne shell
System/
 sar-c.d sar -c demo using DTrace
 syscallbysc.d system calls by system call
 topsyscall display top system call type
 uname-a.d uname -a demo using DTrace
Tcl/
 tcl_*.d 15 scripts for tracing Tcl
User/
 setuids.d snoop setuid calls
Zones/
 zvmstat vmstat info by zone

Total: 230 scripts
root@s11-testbed:/usr/dtrace/DTT/Docs#

```

What does the `dtruss` command do?

*dtruss prints details on process system calls. It is like a DTrace version of `truss` but has been designed to be less intrusive than `truss`.*

Look at the man page for `dtruss`. Which option prints the elapsed time in microseconds? `-e`

6. View the man pages for the following scripts:

- `iosnoop`
- `iotop`
- `iopattern`
- `fddist`

- vopstat
- threaded.d
- sar-c.d
- readdist.d
- writedist.d
- stacksize.d
- rwttop
- sampleproc
- dexplorer

7. Shut down the VMs.