Lift and Shift Guide – Migrating Application Data From Older SPARC Systems to Newer SPARC Systems

Part No: F12883-01
February 2020

ORACLE

Lift and Shift Guide – Migrating Application Data From Older SPARC Systems to Newer SPARC Systems

Part No: F12883-01

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Référence: F12883-01

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` ou le site `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` si vous êtes malentendant.

Contents

Using This Documentation

- **Overview** – This document provides instructions for lifting and shifting application data that is running on Oracle Solaris 10 (or earlier) to Oracle Solaris 11.3 (or later) running on more modern sun4v hardware.
- **Audience** – Experienced Oracle Solaris system administrators
- **Required knowledge** – Experience administering Oracle Solaris computer systems.

Product Documentation Library

Documentation and resources for this product and related products are available at https://docs.oracle.com/cd/E94980_01.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

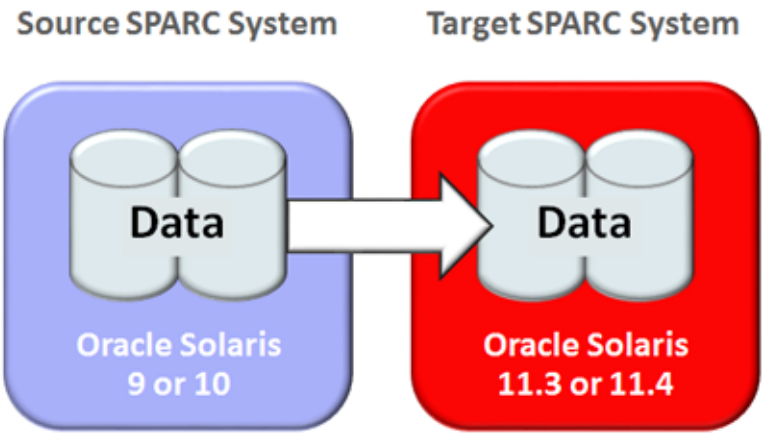
Understanding Data Migration Scenarios

These topics provide an overview of the migration processes that are covered in this guide:

For additional lift and shift guides, visit the Oracle Solaris on SPARC - Lift and Shift Documentation Library at `https://docs.oracle.com/cd/E94980_01/`

Data Migration Overview

This document describes how to use various commands and techniques to migrate applications and data from older SPARC systems to newer SPARC systems running Oracle Solaris 11.3 (or later). The older system is referred to as the *source system*, and the newer system is referred to as the *target system*.



Before you can migrate data, you need to assess the data and in some cases, prepare shared storage (used to temporarily hold compressed data), and prepare the target. These actions are described in "Preparing to Migrate Data" on page 13.

Assumptions

The migration techniques described in this document assume that these conditions are met:

- **Source system data** (to be migrated) – Is on a SPARC system running Oracle Solaris 10 (or later) either on bare metal, a guest logical domain, or a non-global zone.
- **Target system** – can have a Solaris 11.3 (or later) non-global zone or S10 branded zone or logical guest domain that is already prepared and ready to receive the migrated data. The OS and zone or domain can be newly created on the target, or it can be migrated from an older source system.

  For information about how to lift and shift the Oracle Solaris OS to newer SPARC systems, refer to these documents (entire collection is available at `https://docs.oracle.com/cd/E94980_01`).

  - *Lift and Shift Guide - Moving Oracle Solaris 10 Guest Domains to SPARC Servers Running Oracle Solaris 11*
  - *Lift and Shift Guide - Migrating Workloads from Oracle Solaris 10 (ZFS) SPARC Systems to Oracle Solaris 10 Guest Domains*
  - *Lift and Shift Guide - Migrating Workloads from Oracle Solaris 10 SPARC Systems to Oracle Solaris 10 Branded Zones*

Data Migration Scenarios

The data migration scenarios are categorized as follows:

- **Migrating Data with Oracle Solaris Commands** –Oracle Solaris commands can be used to migrate data based on the type of file system or data management that is on the source and target systems. For example, you can use the `ufsdump` and `ufsrestore` commands to migrate data from a UFS file system to another UFS file system, or modernize by migrating to a ZFS file system. To improve migration efficiency, the migration commands are used in conjunction with compression and decompression commands. This requires the use of shared storage to temporarily hold the compressed data. For more information, see "Migrating Data with Oracle Solaris Commands" on page 25.
- **Connect Source System Storage to the Target System** –This scenario involves unplugging a data storage device from the source system and plugging it into the target system. After the device is connected to the target system, you need to configure the target

system to recognize storage device. For example, When you move Fibre Channel devices from one system to another, on the target system, you need to change the FC initiators that are associated with the LUNs on the storage. For more information, see “Move Storage Between Systems” on page 37.

- **Additional Data Migration Methods** –The last chapter describes additional data migration methods. For example, if you are migrating an Oracle Database, you can take advantage of built-in features such as Oracle Recovery Manager (RMAN) to back up the data on the target system.

Example Prompts

In the screen output examples, the command line prompt indicates on which system (target or source), and which zone (global or non-global), or logical domain a command is executed. This table lists the prompts.

Prompt	Description
`root@Source#`	Superuser on the source system
`root@Target#`	Superuser on the target system

Preparing to Migrate Data

These topics describe common activities that are used to plan and prepare storage for the data migration:

- “Determine the Data Storage Requirements” on page 13
- “Configure the Target's Virtual Storage (Logical Domain Virtualization)” on page 18
- “Configure the Target's Virtual Storage (Zone Virtualization)” on page 20
- “Prepare Shared Storage” on page 22

Alternatively, you can disconnect the source storage devices and reconnect them to the target system. In this scenario, refer to “Move Storage Between Systems” on page 37.

▼ Determine the Data Storage Requirements

Before you can migrate data from a source system to a target system, you need to determine the amount of storage required to hold the data. The assessment approach differs based on the source system configuration, but these are typical determinations:

- Total storage capacity and topology – Used to configure equivalent storage on the target system.
- Data management type (or file system type) – Used to determine which commands you can use to gather storage capacity (described in this section) It is also used to determine which commands to use for the migration (see “Migrating Data with Oracle Solaris Commands” on page 25).
- Disk capacity used by the data – Used to calculate the amount of space needed in the shared storage that is used during the migration (see “Prepare Shared Storage” on page 22).

To keep track of the storage information, you can create a worksheet like this example:

Storage Device	Total Storage Capacity of Device	Storage Used by Application	Data Mgt. Type	Notes
Mirrored: c0t1d0 & c1t1d0	600 GB	200 GB	UFS	application abc
Mirrored: c2t600144F0E635D8C700005AC56AB30013d0 c2t600144F0E635D8C700005AC56ADA0014d0	1200 GB	600 GB	ZFS	data
Totals	**1800 GB**	**800 GB**		

1. Identify the storage on the source system.

You can use the `format` utility to list the disks and the disk names that are used in other commands.

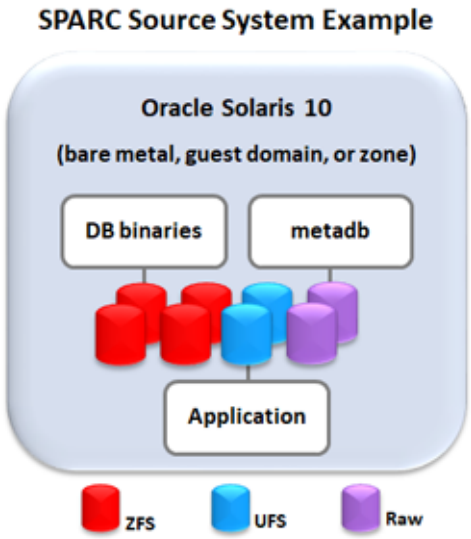
```
root@Source# format

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
          /pci@0,600000/pci@0/scsi@1/sd@0,0
       1. c0t1d0 <SEAGATE-ST960004SSUN600G-0115 cyl 64986 alt 2 hd 27 sec 668>
          /pci@0,600000/pci@0/scsi@1/sd@1,0
       2. c1t0d0 <SUN600G cyl 64986 alt 2 hd 27 sec 668>
          /pci@24,600000/pci@0/scsi@1/sd@0,0
       3. c1t1d0 <SUN600G cyl 64986 alt 2 hd 27 sec 668>
          /pci@24,600000/pci@0/scsi@1/sd@1,0
       4. c2t600144F0E635D8C700005AC7C46F0017d0 <SUN-ZFSStorage7420-1.0 cyl 8124 alt 2
 hd 254 sec 254>
          /scsi_vhci/ssd@g600144f0e635d8c700005ac7c46f0017
       5. c2t600144F0E635D8C700005AC7C4920018d0 <SUN-ZFSStorage7420-1.0 cyl 8124 alt 2
 hd 254 sec 254>
          /scsi_vhci/ssd@g600144f0e635d8c700005ac7c4920018
       6. c2t600144F0E635D8C700005AC56A6A0012d0 <SUN-ZFS Storage 7420-1.0-300.00GB>
          /scsi_vhci/ssd@g600144f0e635d8c700005ac56a6a0012
       7. c2t600144F0E635D8C700005AC56A460011d0 <SUN-ZFS Storage 7420-1.0-300.00GB>
          /scsi_vhci/ssd@g600144f0e635d8c700005ac56a460011
       8. c2t600144F0E635D8C700005AC56AB30013d0 <SUN-ZFS Storage 7420-1.0-1200.00GB>
          /scsi_vhci/ssd@g600144f0e635d8c700005ac56ab30013
       9. c2t600144F0E635D8C700005AC56ADA0014d0 <SUN-ZFS Storage 7420-1.0-1200.00GB>
          /scsi_vhci/ssd@g600144f0e635d8c700005ac56ada0014
      10. c2t600144F0E635D8C700005AC56B2E0016d0 <SUN-ZFS Storage 7420-1.0-200.00GB>
          /scsi_vhci/ssd@g600144f0e635d8c700005ac56b2e0016
      11. c2t600144F0E635D8C700005AC56B080015d0 <SUN-ZFS Storage 7420-1.0-200.00GB>
```

```
          /scsi_vhci/ssd@g600144f0e635d8c700005ac56b080015
CTRL-C
```

2. **Determine the file system types used by the application data.**

   The application data might use a combination of file system or data management types. For example, the application might be on UFS, the database binaries on ZFS, and the metadb data on a raw file system.



   These commands list the file system types.

   - `fstyp /dev/rdsk/`*device_name* (where *device_name* is the `c`*n*`t`*n*`d`*n*`s`*n* such as `c2t600144F0E635D8C700005AC7C46F0017d0s6`)
   - `mount -v`

   Refer to the fstyp(1M) man page and the mount(1M) man page.

   You can also use any available commands and utilities that provided by your data application.

3. **Identify the amount of storage used by the application data.**

   The commands used to gather storage use depend on the file system type. The following table lists Oracle Solaris commands that are used to gather storage usage and topology information. Some examples are shown after the table.

Data Management Type	Space Assessment Commands	Man Page

UFS	`df -h`
	`mount`
	`prtvtoc /dev/rdsk/`*device_name*
		df(1M)
		mount(1M)
		prtvtoc(1M)

SVM	`df -h`
	`metastat`
	`metastat -p` *metaset*
	`mount`
		df(1M)
		metastat(1M)
		mount(1M)

ZFS	`df -h`
	`mount`
	`zpool list`
		df(1M)
		mount(1M)
		zpool(1M)
		Also see:
		Resolving Data Problems in a ZFS Storage Pool
		Understanding How ZFS Calculates Used Space (Doc ID 1369456.1) from `https://support.oracle.com`

ASM and raw disks	`asmcmd`
	`ASMCD> lsdg`
	`ASMCD> lsdsk -p -G DATA`
	`iostat -En` *device_name*
	`prtvotc /dev/rdsk/`*device*
	`sqlplus / as sysasm`
	`SQL> select path, name from v$asm_disk;`
		ASMCMD
		iostat(1M)
		prtvtoc(1M)

Examples

- **Identifying the amount of storage used by the ZFS file systems.**

  Sum the sizes under the SIZE column to determine total amount of storage allocated.

The values in the ALLOC column are used to calculate the amount of storage that is needed in shared storage and on the target system.

```
root@SourceGlobal #zpool list
NAME              SIZE  ALLOC  FREE  CAP  HEALTH  ALTROOT
dbzone            298G  16.4G  282G   5%  ONLINE  -
dbzone_db_binary  149G  22.2G  127G  14%  ONLINE  -
rpool             556G   199G  357G  35%  ONLINE  -
```

- **Identifying the amount of storage used by the UFS file systems.**

  In this example, the database zone logs are on UFS file systems on disks 4 and 5.

  The values in the used column are used to calculate the amount of storage that is needed in shared storage and on the target system.

```
root@Source# df -h -F ufs
Filesystem           size   used    avail   capacity    Mounted on
/logs/archivelogs    197G    13G    182G    7%          /logs/archivelogs
/logs/redologs       9.8G   1.2G    8.6G    13%         /logs/redologs
```

- **Identifying the amount of storage used by ASM raw disks.**

  In this example, the ASM raw disks are on slice 0 of disks 10 and 11 (as numbered by the format utility).

  The capacities in GB can be extracted from the disk partition table using the following scripts.

  In this example, each ASM raw disk is approximately 199 GB.

```
root@Source# echo $((`prtvtoc /dev/rdsk/c2t600144F0E635D8C700005AC56B2E0016d0s2 |
 grep " 0 " | awk '{ print $5 }'`*512/1024/1024/1024))
199
root@Source# echo $((`prtvtoc /dev/rdsk/c2t600144F0E635D8C700005AC56B080015d0s2 |
 grep " 0 " | awk '{ print $5 }'`*512/1024/1024/1024))
199
```

- **To list the SVM metadevice state database (metadb) configuration.**

  In this example, two internal disks (c0t0d0s4 and c1t0d0s4) are used for redundancy of the metadevice state database.

```
root@Source# metadb
        flags           first blk       block count
     a m  pc luo        16              8192            /dev/dsk/c0t0d0s4
     a    pc luo        8208            8192            /dev/dsk/c0t0d0s4
     a    pc luo        16400           8192            /dev/dsk/c0t0d0s4
     a    pc luo        16              8192            /dev/dsk/c1t0d0s4
```

```
a    pc luo        8208           8192           /dev/dsk/c1t0d0s4
a    pc luo        16400          8192           /dev/dsk/c1t0d0s4
```

- **To display disk capacities.**
  You can use the `iostat -En` command with each disk name that was provided in the previous step. For example:
  `iostat -En` *disk_name*
  Repeat the `iostat` command for each disk.
  Note that the sizes shown represent the raw whole disk capacity including a reserved area. The actual usable capacity is less.
  This example shows that the target system is configured to provide the exact same virtual disks and capacities that the guest domain had on the source system.

```
root@Target# iostat -En c0t600144F09F2C0BFD00005BE4C424000Cd0 | grep -i size
Size: 1.07GB <1073741824 bytes>
```

4. **On the Target system, use similar commands to ensure there is equal or greater storage available for the incoming data.**
   If needed, configure virtual storage for the incoming data.
   To configure a LUN for a guest domain, see “Configure the Target's Virtual Storage (Logical Domain Virtualization)” on page 18.
   To configure a LUN for a zone, see “Configure the Target's Virtual Storage (Zone Virtualization)” on page 20.

▼ Configure the Target's Virtual Storage (Logical Domain Virtualization)

Target system storage space is needed for the incoming application data. In this procedure, storage space is prepared on a target system that supports virtualization using Oracle Solaris logical domains. If the target system uses zone virtualization, instead see “Configure the Target's Virtual Storage (Zone Virtualization)” on page 20.

In this example, a LUN (virtual disk) is provisioned for the target guest domain in preparation of the incoming application data. Only one LUN is provisioned, but the commands can be repeated to provision additional LUNs.

1. **Log into the target control domain as superuser.**
2. **Provision a virtual disk .**

Repeat these two steps for each virtual disk that you want to provision.

a. Export the virtual disk backend from a service domain.

`ldm add-vdsdev` *disk_pathname vol_name*`@`*vdisk_server*

Where:

- *disk_pathname* is the path name of the physical device.
- *vol_name* is a unique name of the device being added to the virtual disk server.
- *vdisk_server* is the name of the virtual disk server to which this device is added.

Example:

```
root@Target# ldm add-vdsdev /dev/rdsk/c0t600144F09F2C0BFD00005BE4A9A90005d0s2 data-
vol2@vds0
```

b. Assign the virtual disk to a guest domain.

`ldm add-vdisk` *vdisk_name vol_name*`@`*vdisk_server guest_domain*

Where:

- *vdisk_name* is the name of the new vdisk.
- *vol_name* is the name of an existing volume in which to connect.
- *vdisk_server* is the name of the virtual disk server to which this device is added.
- *guest_domain* is the name of the guest domain to which this vdisk is added.

```
root@Target# ldm add-vdisk vdisk2 data-vol2@vds0 data-gdom
```

3. Verify the disk configuration.

```
root@Target# ldm list -o disk primary

NAME
primary

VDS
    NAME         VOLUME         OPTIONS     MPGROUP     DEVICE
    vds0         solaris-vol0                           /dev/rdsk/
c0t600144F09F2C0BFD00005BE4A8500003d0s2
                 solaris-vol1                           /dev/rdsk/
c0t600144F09F2C0BFD00005BE4A90F0004d0s2
                 data-vol2                              /dev/rdsk/
c0t600144F09F2C0BFD00005BE4A9A90005d0s2
```

4. List the virtual disk bindings.

```
root@Target# ldm ls -o disk data-gdom
NAME
solaris10

DISK
    NAME         VOLUME             TOUT ID   DEVICE    SERVER         MPGROUP
    vdisk0       solaris-vol0@vds0      0     disk@0    primary
    vdisk1       solaris-vol1@vds0      1     disk@1    primary
    vdisk2       data-vol2@vds0         2     disk@2    primary
```

Configure the Target's Virtual Storage (Zone Virtualization)

Target system storage space is needed for the incoming application data. In this procedure, storage space is prepared on a target system that uses Oracle Solaris zones for virtualization. If the target system uses logical domain virtualization, instead see “Configure the Target's Virtual Storage (Logical Domain Virtualization)” on page 18.

These are the key activities:

- Provision a LUN (virtual disk)
- Add a file system to the zone.
- Add a zpool dataset

1. Provision LUNs.

The following commands create two LUNs for an Oracle Solaris 11 zone called `s11zone` to support the incoming application data.

You can obtain the physical disk name (for example, `/dev/rdsk/c0t6000B08414B3030313035353135O167Bd0s6`) from the `format` utility. See “Determine the Data Storage Requirements” on page 13.

Note that the file system is not available until the zone is rebooted. A reboot is performed later in this procedure.

```
root@Target# zonecfg -z s11zone
zonecfg:s11zone>add device
zonecfg:s11zone>set match=/dev/rdsk/c0t6000B08414B3030313035353135O167Bd0s6
zonecfg:s11zone>set allow-partition=true
zonecfg:s11zone>set allow-raw-io=true
zonecfg:s11zone>end

zonecfg:s11zone>add device
```

```
zonecfg:s11zone>set match=/dev/rdsk/c0t6000B08414B3030313035353531350167Dd0s6
zonecfg:s11zone>set allow-partition=true
zonecfg:s11zone>set allow-raw-io=true
zonecfg:s11zone>end
zonecfg:s11zone>commit
zonecfg:s11zone>exit
```

2. **Add a file system to a zone.**

   The following commands add a raw UFS file system (`/u01`) to an Oracle Solaris 11 zone called `s11zone`.

   Note that the file system is not available until the zone is rebooted. A reboot is performed after the zpool is added.

```
root@Target# zonecfg -z s11zone
zonecfg:s11zone> add fs
zonecfg:s11zone> set dir=/u01
zonecfg:s11zone> set special=/dev/md/db_set/dsk/d0
zonecfg:s11zone> set raw=/dev/md/db_set/rdsk/d0
zonecfg:s11zone> set type=ufs
zonecfg:s11zone> end
zonecfg:s11zone> commit
zonecfg:s11zone> exit
```

3. **Add a zpool to the zone.**

```
root@Target# zonecfg -z s11zone
zonecfg:s11zone> add dataset
zonecfg:s11zone> set name = data
zonecfg:s11zone> end
zonecfg:s11zone> commit
zonecfg:s11zone> exit
```

4. **From the global zone, reboot the non-global zone.**

```
root@Target# zoneadm -z s11zone reboot
```

5. **Verify the LUNs.**

```
root@Target# zlogin z11zone 'find /dev/*dsk'
/dev/dsk
/dev/dsk/c0t6000B08414B3030313035353531350167Bd0s6
/dev/dsk/c0t6000B08414B3030313035353531350167Dd0s6
/dev/rdsk
/dev/rdsk/c0t6000B08414B3030313035353531350167Bd0s6
/dev/rdsk/c0t6000B08414B3030313035353531350167Dd0s6
```

6. **From the global zone, verify the zpool.**

```
root@Target# zpool list
NAME SIZE ALLOC FREE CAP DEDUP HEALTH ALTROOT
rpool 1.09T 200G 912G 18% 1.00x ONLINE -
data 1.09T 284G 828G 25% 1.00x ONLINE -
```

7. **From the global zone, verify the file system.**

```
root@Target# df -k /data
Filesystem   1024-blocks  Used       Available  Capacity  Mounted on
data         154828800    26928890   127892854  18%       /data
```

▼ Prepare Shared Storage

Several of the data migration methods in this guide make use of shared storage between the source and target systems. See "Data Migration Commands" on page 25. If you plan to use those methods, you can use the steps in this section to help you prepare shared storage.

In this section, a network file system (NFS) is exported from the target system and mounted by the source system. During the data migration process, the data is compressed and transferred to the shared storage, then uncompressed and transferred to the source system.

1. **Ensure that there is enough available storage capacity on the planned shared storage.**

   You can use this rule of thumb:

   Take the amount of storage used by the data and divide by 2 (this takes into account that the data will be compressed). Then add 30% to that value (ensures adequate headroom).

2. **On the target control domain, create and export a file system.**

```
root@Target# zfs create -o mountpoint=/SharedStorage vpool/SharedStorage
root@Target# zfs set share.nfs=on vpool/SharedStorage
root@Target# zfs set share.nfs.root=\* vpool/SharedStorage
root@Target# zfs set share.nfs.rw=\* vpool/SharedStorage
root@Target# exportfs
vpool_SharedStorage      /SharedStorage   sec=sys,root,rw
```

3. **As superuser on the source system, create a mount point and mount the shared storage.**

   In this example, `TargetControlDom:/SharedStorage` is an NFS share exported from the target system.

```
root@Source# mkdir /SharedStorage
```

```
root@Source# mount -F nfs TargetControlDom:/SharedStorage /SharedStorage
```

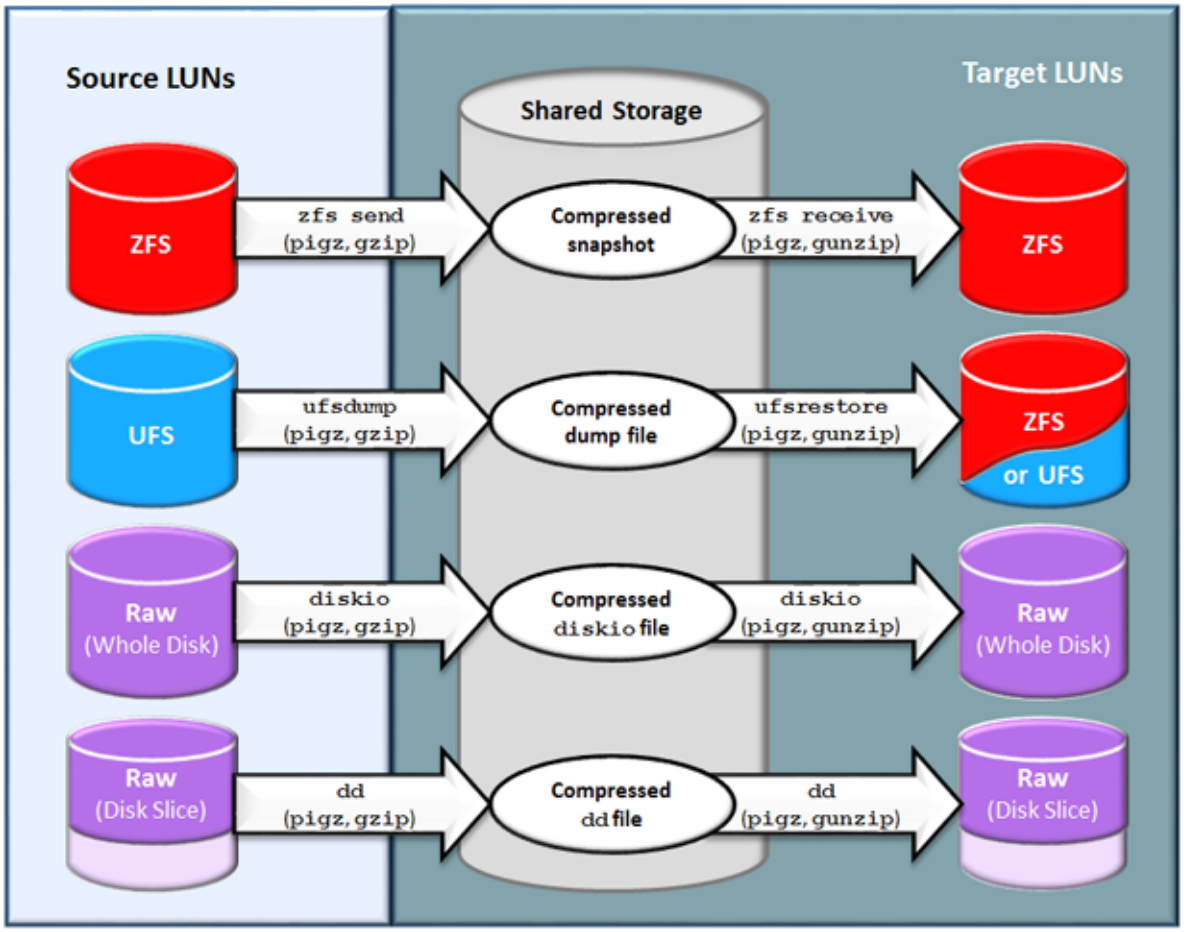
Migrating Data with Oracle Solaris Commands

These topics describe how to use various Oracle Solaris commands to migrate data from a SPARC source system to a SPARC target system:

Description	Links
Learn about using Oracle Solaris commands for migration.	“Data Migration Commands” on page 25 “Minimize Downtime with Incremental Copies” on page 28
Migrate a ZFS file system.	“Example 1a – Migrate a ZFS File System (Full)” on page 28 “Example 1b – Migrate a ZFS File System (Incremental)” on page 30
Migrate a UFS file system.	“Example 2a – Migrate a UFS File System (Full)” on page 31 “Example 2b – Migrate a UFS File System (Incremental)” on page 32
Migrate a whole raw disk.	“Example 3 - Migrate a Whole Raw Disk” on page 34
Migrate a raw disk slice.	“Example 4 – Migrate a Raw Disk Slice” on page 35
Migrate any type of file system.	“Example 5 – Migrate Any File System With `rsync`” on page 36

Data Migration Commands

This diagram shows which Oracle Solaris commands can be used to migrate data based on the type of file system or data management that is on the source and target systems. The commands in parenthesis are secondary commands that are used with the main command to compress and decompress the data, shortening the migration duration.



This table describes the primary commands.

TABLE 1 Primary Data Migration Commands

Command	Description	Link to Example
`zfs send(1M)` man page	Sends a copy of a snapshot stream to another pool on the same system or in another pool on a different system. Can be used to create incremental copies.	"Example 1a – Migrate a ZFS File System (Full)" on page 28
`zfs receive(1M)` man page	Receives a snapshot stream.	"Example 1b – Migrate a ZFS File System (Incremental)" on page 30
`ufsdump(1M)` man page	Creates a copy of a UFS file system. Can be used to create incremental copies.	"Example 2a – Migrate a UFS File System (Full)" on page 31

Command	Description	Link to Example
`ufsrestore(1M)` man page	Restores files that were created with the `ufsdump` command.	"Example 2b – Migrate a UFS File System (Incremental)" on page 32
`diskio`	Creates a copy of an entire raw disk. The `diskio` command is available under these circumstances: ■ In Oracle Solaris 11.3 and 11.4 ■ In Oracle Solaris 10 when patch 151934-06 (or later) is installed. Note that `diskio` is only available in the global zone.	"Example 3 - Migrate a Whole Raw Disk" on page 34
`dd` man page	Creates a copy of an individual disk slice.	"Example 4 – Migrate a Raw Disk Slice" on page 35
(Not shown in diagram) `rsync(1)` man page	Copies files to local and remote systems. Can be used to create incremental copies. This is a versatile command because it can be used to copy files to and from UFS and ZFS file systems. However, performance might be better when using the other commands.	"Example 5 – Migrate Any File System With `rsync`" on page 36

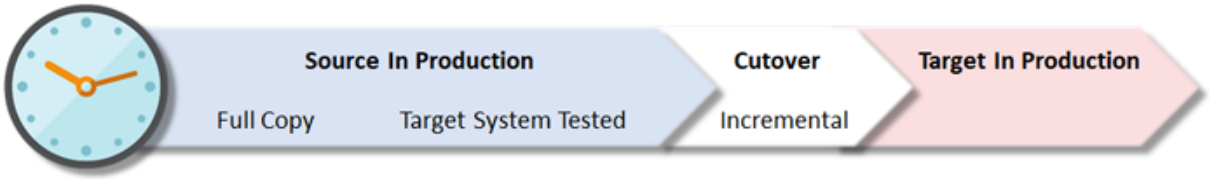
TABLE 2 Secondary Commands

Command	Description
`pigz(1)` man page	Efficiently compresses or decompresses files through parallel processing. The `pigz` command is available under these circumstances: ■ Available in Oracle Solaris 11.3 and 11.4. It is in the `/usr/bin` directory. ■ Available in Oracle Solaris 10, when patch 151934-06 (or later) is installed. Note that `pigz` is only available in the global zone in the `/opt/SUNWldm/lib/contrib` directory.
`gzip(1)` man page	Compresses or decompresses files. Use this command when `pigz` is not available.
`gunzip(1)` man page	Compresses or decompresses files. Use this command when `pigz` is not available.

Minimize Downtime with Incremental Copies

You can take advantage of incremental copy features of certain commands to minimize the downtime associated with data migration. This method migrates the data in multiple phases:

- Phase 1 – A full copy of the source data is migrated to the target system while the source system remains in production.
- Phase 2 – The target system is configured and tested while the source system remains in production.
- Phase 3 – At the cut-over phase, an incremental copy of the source data is captured and copied to the target system.

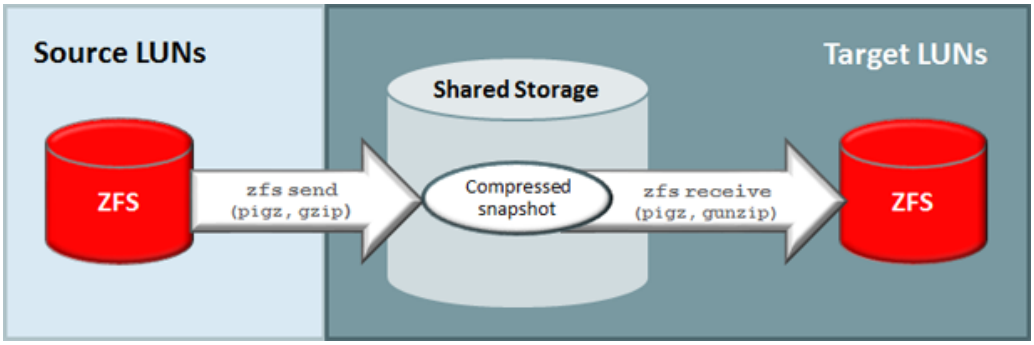


The following Oracle Solaris commands provide options for creating full and incremental copies:

- zfs send – See:
  "Example 1a – Migrate a ZFS File System (Full)" on page 28
  "Example 1b – Migrate a ZFS File System (Incremental)" on page 30
- ufsdump – See:
  "Example 2a – Migrate a UFS File System (Full)" on page 31
  "Example 2b – Migrate a UFS File System (Incremental)" on page 32

▼ Example 1a – Migrate a ZFS File System (Full)

This example migrates data from a ZFS file system to an already prepared ZFS file system.



1. **Ensure that shared storage is prepared.**

   See "Prepare Shared Storage" on page 22.

2. **From the source global zone, create a ZFS snapshot of the source system's zpool.**

```
root@Source# zfs snapshot -r zone01@snap01

root@Source# zfs list -t snapshot
NAME USED AVAIL REFER MOUNTPOINT
zone01@snap01 0 - 16.8G -
```

3. **Send the ZFS snapshots to the shared storage.**

   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Source# zfs send zone01@snap01 | pigz > /SharedStorage/zone01.gz
```

4. **From the target, transfer the compressed snapshot from the shared storage to the target zpool.**

   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Target# pigz -dc -f /SharedStorage/zone01.gz | zfs receive -F zone01@snap01
```

Example 1b – Migrate a ZFS File System (Incremental)

This example migrates an incremental ZFS dataset from the source system to shared storage, then to an already prepared ZFS file system on the target system.

This procedure is used in situations when there is a lapse between the time that the initial full ZFS data is captured on the source and then fully restored on the target. If the source system remained in production during the initial migration, you can perform an incremental `zfs send` to migrate only the changed data.

1. **Ensure that shared storage is prepared.**

   See "Prepare Shared Storage" on page 22.

2. **From the source global zone, create a ZFS snapshot of the source system.**

```
root@Source# zfs snapshot -r zone01@snap02

root@Source# zfs list -t snapshot
NAME          USED    AVAIL    REFER    MOUNTPOINT
zone01@snap01 0       -        35K      -
```

3. **Use specific `zfs send` options to only transfer changes that occurred after the full snapshot.**

   The `zfs send` command offers these options for generating incremental streams:

   - `-R` – Replicates the file system up to the named snapshot.
   - `-I` – Generates a stream package that sends all intermediary snapshots from the first snapshot to the second snapshot.

   This example sends the incremental snapshot to shared storage.

   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Source# zfs send -R -I zone01@snap01 zone01@snap02 | pigz > /SharedStorage/
zone02.gz
```

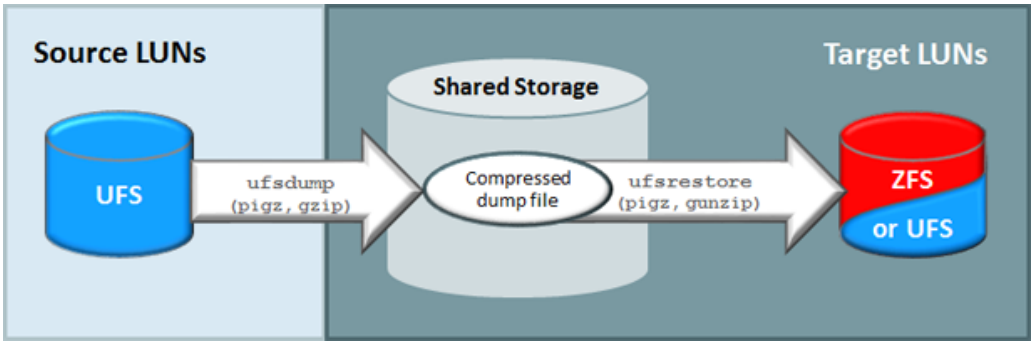
4. **From the target guest domain, transfer the compressed snapshot from the shared storage to the target zpool.**

   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Target#  pigz -dc -f /SharedStorage/zfs_send/test_snapI2.gz | zfs receive zone01
```

Example 2a – Migrate a UFS File System (Full)

This is an example that migrates all of the data from a UFS file system to an already prepared ZFS or UFS file system.



1. **Ensure that shared storage is prepared.**

   See "Prepare Shared Storage" on page 22.

2. **From the source global zone, create a copy of the UFS file system in a compressed format on shared storage.**

   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Source# ufsdump 0cf - /dev/md/rdsk/d20 | pigz > /SharedStorage/redo.ufsdump.gz
DUMP: Date of this level 0 dump: Mon Jul 30 11:09:29 2018
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/md/rdsk/d20 (SourceGlobal:/zones/dbzone/root/logs/redologs) to
standard output.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 1232116 blocks (601.62MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 1232026 blocks (601.58MB) on 1 volume at 6315 KB/sec
DUMP: DUMP IS DONE
```

3. **Check to see that the ufsdump file is starting to show up in the shared storage.**

```
root@Source# cd /SharedStorage
```

```
root@Source# ls -rtlh *.gz
-rw-r--r-- 1 root root 59M Jul 30 11:09 redo.ufsdump.gz
```

4. **From the target guest domain, uncompress and restore the dump file.**

   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Target# pigz /SharedStorage/redo.ufsdump.gz | ufsrestore xvf -
Verify volume and initialize maps
Dump date: Mon Jul 30 11:09:29 2018
Dumped from: the epoch
Level 0 dump of /zones/dbzone/root/logs/redologs on SourceGlobal:/dev/md/dsk/d20
Label: none
Extract directories from tape
Initialize symbol table.
Extract requested files
extract file ./redo04.log
extract file ./redo05.log
extract file ./redo06.log
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] y
```

▼ Example 2b – Migrate a UFS File System (Incremental)

This example migrates an incremental subset of the data from a UFS file system to an already prepared ZFS or UFS file system.

This procedure is used in situations when there is a lapse between the time that the initial full dump is captured on the source and then restored on the target . If the source system remained in production, it is likely that data has changed on the source system. In such cases, you can perform an incremental dump and restore to migrate only the changed data to the target system.

1. **Ensure that shared storage is prepared.**

   See "Prepare Shared Storage" on page 22.

2. **From the source global zone, create an incremental copy of the UFS file system in a compressed format on shared storage.**

   The following examples show two types of incremental dumps. The difference between the examples is the way the source disk device is specified.

   - **UFS running on SVM (raw disk specified)**

Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Source# ufsdump 0cf - /dev/md/rdsk/d20 | pigz > /SharedStorage/redo.ufsdump.gz
```

- **UFS (raw physical disk specified)**

  Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Source# ufsdump 1cf - /dev/rdsk/c0t5000CCA0804092ECd0s6 | pigz
 > /SharedStorage/ufs1_ufsdump1.gz
DUMP: Date of this level 1 dump: Wed Sep 25 14:45:01 2019
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t5000CCA0804092ECd0s6 (opc-s02-s72-06:/ufs1) to standard
 output.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Writing 63 Kilobyte records
  DUMP: Estimated 930 blocks (465KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: 880 blocks (440KB) on 1 volume at 22000 KB/sec
  DUMP: DUMP IS DONE
```

3. **From the target guest domain, uncompress and restore the incremental dump file.**

   The following examples show two types of incremental restores.

   - **UFS running on SVM**

     Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Target# pigz -dc -f /SharedStorage/redo.ufsdump.gz | ufsrestore xvf -
```

   - **UFS**

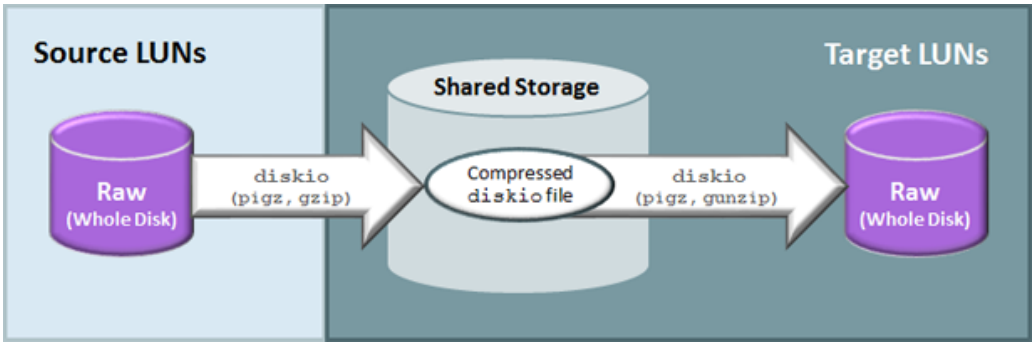
     Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Target# pigz -dc -f /SharedStorage/ufs1_ufsdump1.gz | ufsrestore xvf -
Verify volume and initialize maps
Dump   date: Wed Sep 25 14:45:01 2019
Dumped from: the epoch
Level 1 dump of /ufs1 on opc-s02-s72-06:/dev/dsk/c0t5000CCA0804092ECd0s6
Label: none
```

```
Extract directories from tape
Initialize symbol table.
Warning: ./lost+found: File exists
Extract requested files
extract file ./hosts
extract file ./nsswitch.conf
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] y
Directories already exist, set modes anyway? [yn] n
```

▼ Example 3 - Migrate a Whole Raw Disk

This is an example that migrates data from a whole raw disk to an already prepared whole raw disk.



1. **Ensure that shared storage is prepared.**
   See "Prepare Shared Storage" on page 22.

2. **On the source system, create a compressed `diskio` file of the whole raw disk on shared storage.**
   In this example, the compressed `diskio` file is created on shared storage.
   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

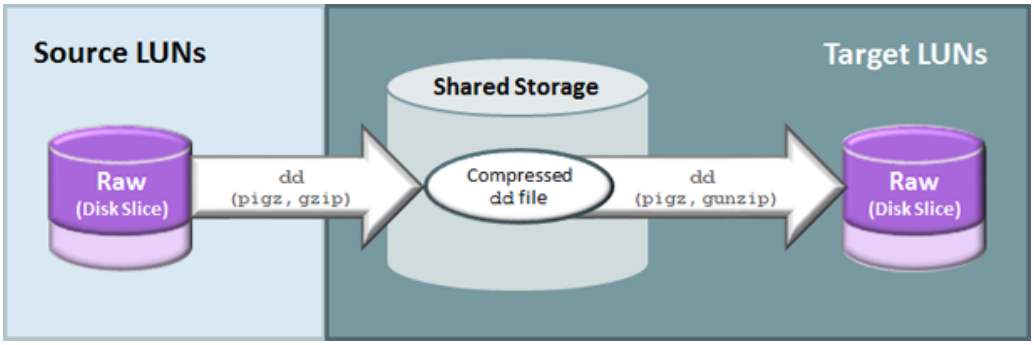
```
root@Source# BUFFERSIZE=104857600
root@Source# /opt/SUNWldm/lib/contrib/diskio -i /dev/rdsk/
c0t600144F0CD152C9E000057F54D4D0026d0s2 -b ${BUFFERSIZE} | pigz > /SharedStorage/
data_disk1.img.gz
```

3. **On the target system, uncompress the `diskio` file.**

```
root@Target# cd /SharedStorage
root@Target# pigz -dc -f /SharedStorage/data_disk1.img.gz |/opt/SUNWldm/lib/contrib/
diskio -o /dev/rdsk/c0t600144F09F2C0BFD00005A2865C30006d0s2
```

▼ Example 4 – Migrate a Raw Disk Slice

This is an example that migrates data from an individual raw disk slice to an already prepared raw disk slice.



1. **Ensure that shared storage is prepared.**
   See "Prepare Shared Storage" on page 22.

2. **On the source system, use the `dd` command piped to a compression command to copy the data to shared storage.**
   The `dd bs` option is used to specify the block size in bytes. Using a large block size improves performance.
   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Source# dd if=/dev/rdsk/c2t600144F0E635D8C700005AC56B080015d0s0 bs=104857600 | pigz
 > /SharedStorage/slice-data.img.gz
```

3. **On the target system, use a decompression command piped to the `dd` command to copy the data from shared storage to the target system.**
   Note – For the appropriate path to the `pigz` command, see the Secondary Commands table in "Data Migration Commands" on page 25.

```
root@Target# pigz -c /SharedStorage/asm1.img.gz | dd of=/dev/rdsk/c0d6s0 bs=104857600
```

Example 5 – Migrate Any File System With `rsync`

In this example, the rsync command is used to migrate data any file system type to a target system.

- **From the source, create an archive copy of the data.**

  The following `rsync` syntax can be used to create full and incremental copies:

  `rsync -avSM --delete` *source destination*

  Where:

  - `a` – Creates a copy in archival mode (maintains recursive directory structure and symbolic links).
  - `v` – Increases verbosity.
  - `S`– Handles sparse files efficiently
  - `M`– Sends OPTION to the remote side only.
  - `--delete`– Deletes extraneous files from the receiving side (files that aren't on the sending side).
  - *source*– The source directory to copy.
  - *destination*– The target destination directory.

  Example:

  ```
  root@Source# rsync -avSH --delete  /export/projects target_name:/export
  ```

Additional Data Migration Methods and Resources

These topics provide additional techniques, methods, and examples for migrating data to a target system:

Move Storage Between Systems

You can lift and shift data by moving the storage from the original source system and reconnected it to the target system. This task can be accomplished by physically recabling the devices, or by using multiported devices such as the devices on a SAN or iSCSI network. Each storage type requires a particular set of administrative actions.

1. **Prepare the source system for the removal of the storage.**

   Stop any applications that might attempt to access the storage that you plan to remove.

   Backup the source system data.

   Additional administrative tasks are usually needed to logically remove the storage from the source system. The tasks differ based on the type of storage and file systems. Here are some common tasks:

   - **ZFS** – Run the `zpool export` command. This unmounts file systems and removes the zpool from the source system. For more information, refer to: `https://docs.oracle.com/cd/E19253-01/819-5461/gbchy/index.html`
   - **UFS and NFS** – Use the `umount` command to unmount the file systems. Also consider removing the file systems from the `/etc/vfstab` file to prevent the system from attempting to mount the removed file systems. For more information, refer to How to Unmount a File System at `https://docs.oracle.com/cd/E26505_01/html/E37385/fscreate-6.html#scrolltoc`

2. **If needed, physically recable the storage devices to the target system.**

3. **Configure the storage connections on the target system.**

   - **ZFS** – Run the `zpool import` command. to discover and import the zpool. For more information, refer to: https://docs.oracle.com/cd/E19253-01/819-5461/gbchy/index.html
   - **Fibre Channel configured LUNs** – Perform the ZFS SAN Fibre Channel Configuration procedure that is described here:

     https://docs.oracle.com/cd/E71909_01/html/E71919/gooud.html
   - **iSCSI configured LUNs** – Perform one of the Configuring iSCSI procedures that are described here:
     - ZFS appliance BUI – https://docs.oracle.com/cd/E37831_01/html/E52872/gocub.html
     - ZFS appliance CLI – https://docs.oracle.com/cd/E37831_01/html/E52872/gocux.html

4. **Configure volume manager.**

   Follow one of the following migration procedures before starting SVM, ZFS or VxVM configurations on the target system.

   - **ZFS** –

     https://docs.oracle.com/cd/E19253-01/819-5461/gbchy/index.html
   - **NFS mounts** – Create target NFS mounts that are same NFS mounts seen in source system.

     On ZFS Storage side, add NFS node exceptions for accessing NFS file system, that are similar to source config.

     Update the target system `/etc/vfstab` file for NFS mounts that match with source system. and create mount points and mount file system.

     https://docs.oracle.com/cd/E37831_01/html/E52872/configuration__services__nfs__sharing_a_filesystem_over_nfs.html
   - **SVM** –

     With metasets:

     https://docs.oracle.com/cd/E19120-01/open.solaris/819-2789/efnri/index.html

     Without metasets:

     https://docs.oracle.com/cd/E26505_01/html/E29409/troubleshoottasks-84.html

     https://docs.oracle.com/cd/E19683-01/806-6111/6jf2ve3ne/index.html
   - **VxVM** –

     Follow the Veritas/Symantec documentation for exporting Veritas disk groups from source system and importing them on the target system.

Additional Data Migration Resources

ZFS Shadow Migration

For migrating data from one ZFS storage appliance to another ZFS storage appliance, you can use the shadow migration feature provided by the ZFS storage appliance. Shadow migration supports migration only from NFS sources.

For more details, refer to the *Oracle ZFS Storage Appliance Administration Guide* at: https://docs.oracle.com/cd/E51475_01/html/E52872/shares__shadow_migration___shadow_migration_.html

ZFS Storage Appliance Replication

Replication can be used to migrate data and configuration information from one ZFS storage appliance to another ZFS storage appliance.

For more details, refer to the *Oracle ZFS Storage Appliance Administration Guide* at: https://docs.oracle.com/cd/E37831_01/html/E52872/gocuj.html

Oracle Database Utilities for Data Migration

- **Oracle Recovery Manager (RMAN)** – Is Oracle's preferred solution for database backup and recovery, and can also be used to transport databases and tablespaces to target systems. You can use RMAN to create a database and tablespace copies that can be imported into another database, or move an entire database from one system to another.

  For more information including access to the RMAN documentation, go to https://www.oracle.com/database/technologies/high-availability/rman.html
- **Active Data Guard** – Is a set of services that create, manage, and monitor one or more standby databases to enable production Oracle databases to survive disasters and data corruptions. This technology can be used to create a physical standby database on the target system while the source system remains in production. Once the standby database is

created, Oracle Data Guard automatically maintains the standby database by transmitting primary database redo data to the standby system where the redo data is applied to the standby database. When the target standby database is ready to go into production, use the Active Data Guard switchover feature to switch the standby database to the primary database.

Oracle Data Guard transport services are also used by other Oracle features such as Oracle Streams and Oracle GoldenGate for efficient and reliable transmission of redo from a source database to one or more remote destinations.

For more information including access to the documentation, go to https://www.oracle.com/database/technologies/high-availability/dataguard.html

Built-in Application Utilities

The applications on your system might include utilities that augment data migration.

Index