4

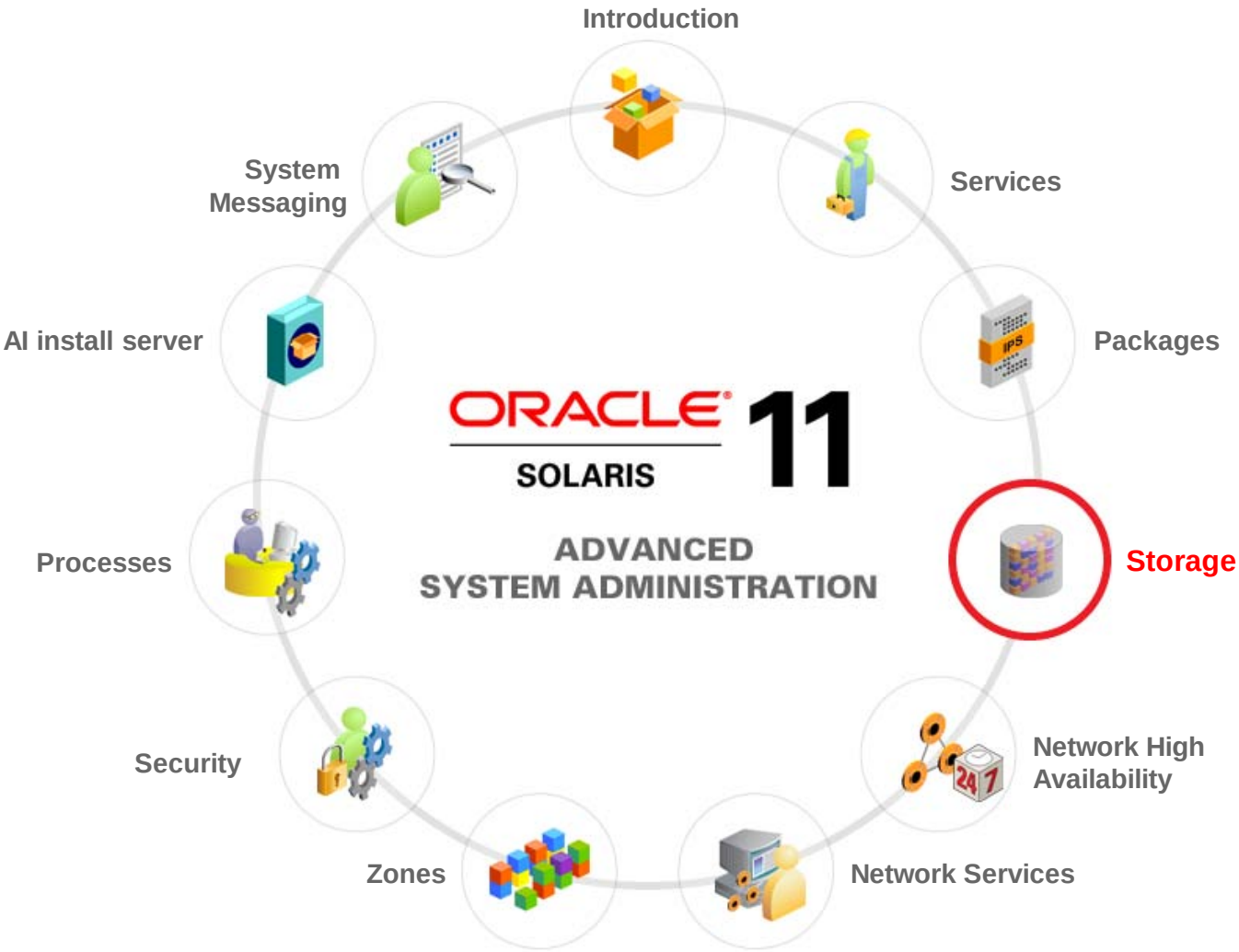
Managing Data Backup and Restore by Using ZFS

Objectives

After completing this lesson, you should be able to:

- Plan for data storage configuration and backup
- Manage data redundancy with mirrored storage pools
- Back up and restore data by using ZFS snapshots
- Manage data storage space with ZFS file system properties
- Troubleshoot ZFS failures

Job Workflow



Agenda

- Planning for data storage configuration and backup
- Managing data redundancy with mirrored storage pools
- Administering data backup and restore by using ZFS snapshots
- Managing data storage space with ZFS file system properties
- Troubleshooting ZFS failures

Planning for Data Storage Configuration and Backup

- Ensure that critical business application data is protected, backed up, recoverable, and accessible.
- You can support business applications by:
  - Providing data redundancy by using mirrored storage pools
  - Setting up file systems to store the data
  - Backing up the file systems by using snapshots and saving them on a remote system
  - Minimizing storage space by using the ZFS file system `compression` property

Identifying Data Backup and Restore Requirements

As part of planning for data backup and restore, it is important to consider the following questions:

- How will data be backed up?
- How often will data be backed up?
- How can these backups be used to recover data in case of data loss?

Determining Storage Pool Requirements

As part of planning, the following storage pool requirements should be identified:

- Devices
  - Disks that are at least 128 MB in size
  - Disks that are not in use by other parts of the operating system
  - Individual slices on a preformatted disk or entire disks formatted as a single, large slice
  - Use of log and cache devices for improved performance
- Level of data redundancy
  - Non-redundant (striped) configurations
  - Mirrored
  - RAID-Z

Identifying Replication Features of a ZFS Storage Pool

Data replication features:

- Mirrored storage pool configuration
  - Requires at least two disks
  - Makes it possible to create more than one mirror in each pool
  - Provides data redundancy through its configuration options and self-healing data features
- RAID-Z storage pool configuration
  - **Single-parity RAID-Z (`raidz` or `raidz1`):** Similar to RAID-5
  - **Double-parity RAID-Z (`raidz2`):** Similar to RAID-6
  - **Triple-parity RAID-Z (`raidz3`):** Similar to `raidz2` with an additional parity protection level
- Self-healing data
  - Supported in a mirrored or RAID-Z configuration
  - Automatically detects and repairs bad data blocks

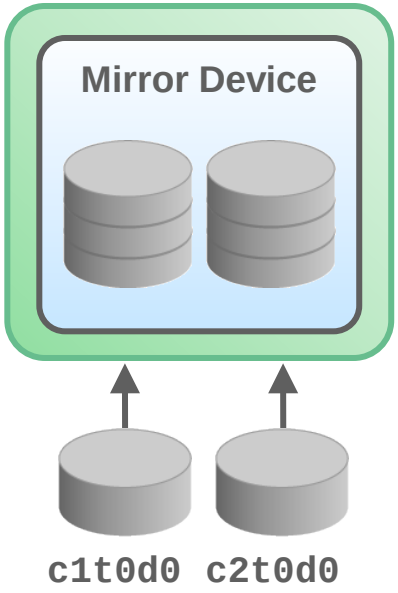
Identifying Replication Features of a ZFS Storage Pool

Data replication features:

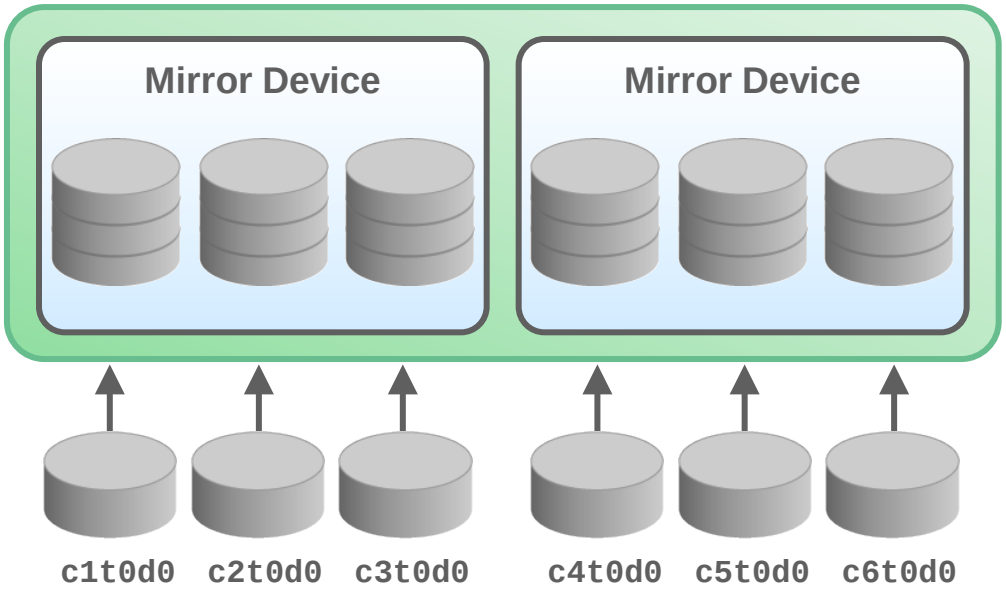
- ZFS hybrid storage pool
  - Is available in the Oracle ZFS Storage Appliance
  - Is a special storage pool that combines DRAM, SSDs, and HDDs
  - Improves performance and increases capacity while reducing power consumption
  - Enables you (through this product's management interface) to select the ZFS redundancy configuration of the storage pool and easily manage other configuration options
- Dynamic striping

Mirrored Storage Pool Configuration

- At least two disks are required.
- Many disks can be used.
- Multiple mirrors can be created in each pool.



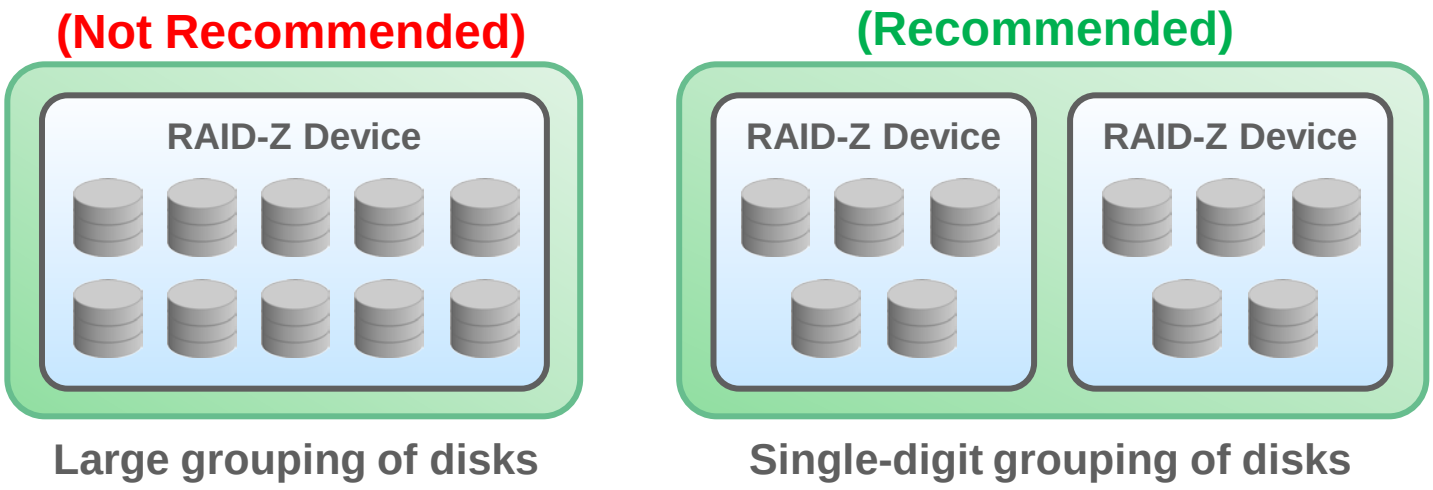
Simple Mirrored Configuration



Complex Mirrored Configuration

RAID-Z Storage Pool Configuration

- Minimum disk usage recommendations by RAID-Z level
  - **`raidz` or `raidz1`:** Use at least three disks (2 + 1).
  - **`raidz2`:** Use at least five disks (3 + 2).
  - **`raidz3`:** Use at least eight disks (5 + 3).
- For better performance, configure storage pools with single-digit groupings of disks.



Self-Healing Data

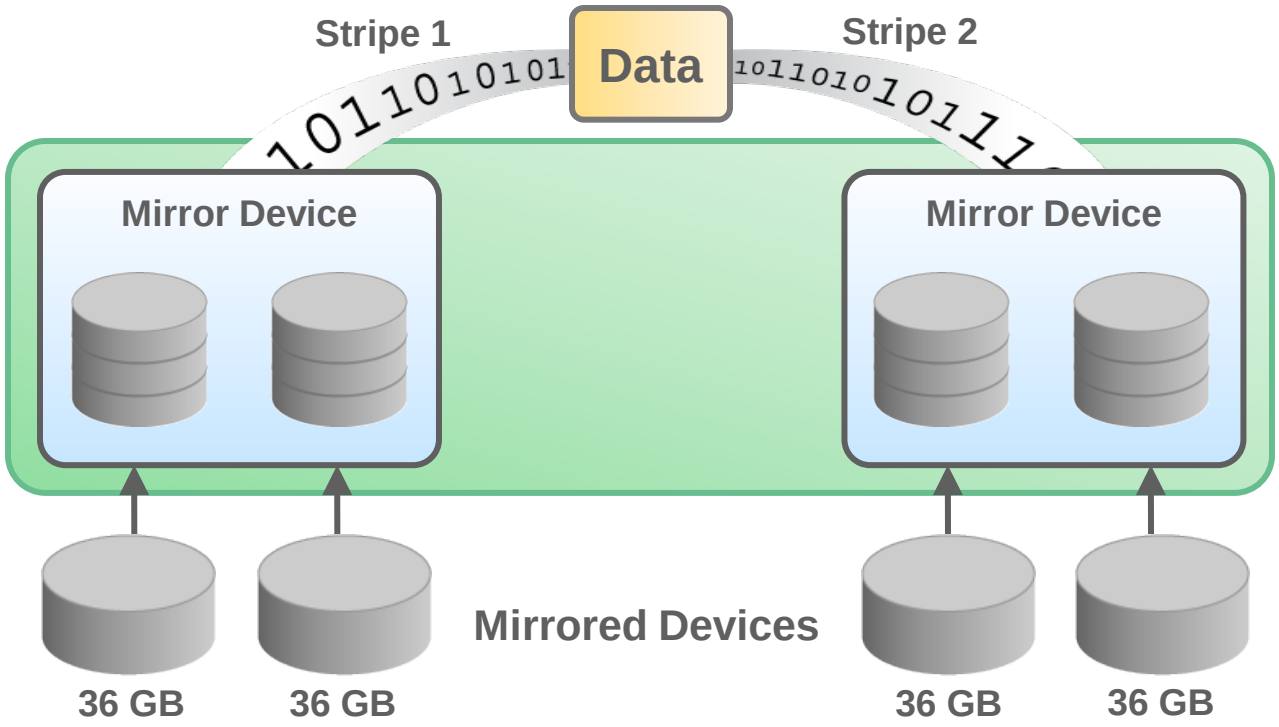
- A mirrored or RAID-Z configuration provides support for self-healing data.
- When a bad data block is detected:
  - Correct data is fetched from another replicated copy
  - Bad data is repaired by replacement with the good copy

Dynamic Striping

- Data is dynamically striped across all top-level virtual devices.
- Data placement is done at write time.
- When a new virtual device is added, data is gradually allocated to the new device.

Note: Although ZFS supports combining different types of virtual devices within the same pool, the recommended practice is to use top-level virtual devices of the same type with the same redundancy level in each virtual device.

Dynamic Striping in a Mirrored Pool



Recommended Storage Pool Practices

- Keep pool capacity below 90% for best performance.
- For random read/write workloads, mirrored pools are recommended instead of RAID-Z pools.
- Separate your log devices.
- Monitor the following:
  - Pool health
  - Storage pool space
  - Pool and device failures

Determining File System Requirements

- Determine how to set up your file systems to:
  - Store business application data efficiently
  - Facilitate data backup and restore operations
- One recommended approach is to:
  - Create one file system for the main application
  - Create subfile systems for each subapplication
- This approach ensures that the following actions are easy:
  - Backing up the entire file system
  - Backing up the data on each subfile system
  - Restoring the entire file system
  - Setting file system properties at the highest level and having subfile systems inherit the property values

Identifying Your Data Backup and Restore Strategy

As part of planning, you should identify your data backup and restore strategy:

- Use ZFS snapshots to create file system backups and save them on a remote system.
- Use send and receive commands for saving incremental changes between snapshots or for remote replication.

Determining Ways to Save Data Storage Space

ZFS offers a file system `compression` property that:

- Is used to enable or disable compression for a file system
- Compresses only new data on an existing file system if it is enabled after file system creation
- The following compression algorithms are available:

Compression Algorithms	Description
`gzip`	Standard UNIX compression
`gzip-`*`N`*	Selects a specific `gzip` level. `gzip-1` provides the fastest gzip compression, `gzip-9` provides the best data compression and `gzip-6` is the default
`lz4`	Provides better compression with lower CPU overhead
`lzjb`	Optimized for performance while providing decent compression
`zle`	Zero length encoding is useful for datasets with large blocks of zeros

Implementing the Data Storage Configuration and Backup Plan

- Configure and test the functionality of a mirrored storage pool.
- Create snapshots of the file systems in the mirrored storage pool to use as backups.
- Set and test the ZFS `compression` property on the file systems.
- Troubleshoot ZFS device and data issues.

Quiz

In a mirrored storage pool configuration, ZFS supports data redundancy but does not support dynamic striping.

a. True
b. False

Quiz

If you want better performance when working with RAID-Z pools, you should configure the pools with single-digit groupings of disks.

a. True
b. False

Quiz

With a RAID-Z storage pool, the data is dynamically striped across all the virtual devices in the pool and redundant among the devices in a single vdev.

a. True
b. False

Quiz

Which RAID-Z parity option does ZFS support?

a. `raidz / raidz1` only

b. `raidz / raidz1` and `raidz2` only

c. `raidz / raidz1, raidz2,` and `raidz3`

d. `raidz / raidz1, raidz2, raidz3,` and `raidz4`

Quiz

Which ZFS file system property enables you to minimize the amount of data storage space that is used in a storage pool?

a. minimize
b. restrictsize
c. compressratio
d. compression

Agenda

- Planning for data storage configuration and backup
- Managing data redundancy with mirrored storage pools
- Administering data backup and restore by using ZFS snapshots
- Managing data storage space with ZFS file system properties
- Troubleshooting ZFS failures

Managing Data Redundancy with Mirrored Storage Pools

This section covers the following topics:

- Creating a mirrored storage pool
- Adding log and cache devices to a storage pool
- Managing devices in a ZFS storage pool

Creating a Mirrored Storage Pool

To create a new ZFS mirrored storage pool, use `zpool create` followed by the pool name, the `mirror` keyword, and the storage devices that will compose the mirror.

```
# zpool create hrpool mirror c1t0d0 c2t0d0 mirror c3t0d0 c4t0d0
```

Data is:

- Dynamically striped across both mirrors
- Redundant between each disk within a mirror

Adding Log Devices to a Storage Pool

A log device:

- Can be added as part of, or after, pool creation
- Can be removed
- Is designated by the `log` keyword

```
# zpool create datapool mirror c1t1d0 c1t2d0 log mirror c1t5d0 c1t8d0
```

Adding Cache Devices to a Storage Pool

Cache devices:

- Can be added as part of, or after, pool creation
- Can be removed
- Cannot be mirrored and cannot be part of a RAID-Z configuration
- Are designated with the `cache` keyword

```
# zpool create appool mirror c0t2d0 c0t4d0 cache c0t0d0
```

Note: You can monitor cache statistics with `zpool iostat`.

Managing Devices in ZFS Storage Pools

The tasks that you can perform with the devices in a pool include:

- Adding top-level virtual devices
- Attaching and detaching devices
- Taking a device offline
- Bringing a device online
- Designating hot spares

Adding Devices to a Storage Pool

To add a new virtual device to a pool, use the `zpool add` command.

```
# zpool add appool mirror c2t1d0 c2t2d0
```

By adding a new top-level virtual device, space is:

- Dynamically added to the pool
- Immediately available to all the datasets in the pool

Note: A dataset is a generic name for the following ZFS entities:

- File systems
- Snapshots
- Volumes

Attaching Devices to a Storage Pool

To attach a new device to an existing mirrored or nonmirrored pool, use the `zpool attach` command.

```
# zpool attach appool c1t1d0 c2t1d0
```

Attaching Devices to a Storage Pool

```
# zpool attach appool c1t1d0 c2t1d0
# zpool status
  pool: appool
 state: ONLINE
 scrub: resilver completed after 0h0m with 0 errors on Fri Dec
13 14:11:33 2013
config:
    NAME         STATE     READ   WRITE  CKSUM
    appool       ONLINE       0      0      0
      mirror-0   ONLINE       0      0      0
       c0t1d0    ONLINE       0      0      0
       c1t1d0    ONLINE       0      0      0
       c2t1d0    ONLINE       0      0      0 73.5K resilvered
```

Resilvering: The process of transferring data from one device to another device

Taking Devices Offline in a Storage Pool

To take a device offline, use `zpool offline` followed by the pool name and the device name.

```
# zpool offline hrpool c1t0d0
```

When a device is offline:

- ZFS does not send it any requests
- It remains offline after a system reboot

  Note: Use `zpool offline -t` to take a device offline temporarily.
- It is not detached from the storage pool

Detaching Devices from a Storage Pool

To detach a device from a mirrored storage pool, use the `zpool detach` command.

```
# zpool detach appool c2t1d0
```

Note: This operation is refused if there are no other valid replicas of the data.

Bringing Devices Online in a Storage Pool

To bring a device online, use `zpool online` followed by the pool name and the device name.

```
# zpool online hrpool c1t0d0
bringing device c1t0d0 online
```

When a device is brought back online, data that was added to the storage pool while the device was offline resilvers to the device.

Note: You cannot use `zpool online` to replace a disk.

Replacing Devices in a Storage Pool

To replace a failed device with another device in the same location, use `zpool replace` followed by the pool name and the device name.

```
# zpool replace hrpool c1t1d0
```

If the device is in a different location, specify both devices.

```
# zpool replace hrpool c1t1d0 c1t2d0
```

Note: The size of the replacement device must be greater than or equal to the minimum size of all the devices in a mirror or RAID-Z configuration.

Designating Hot Spares in a Storage Pool

With the ZFS hot spares feature, you can:

- Identify disks to replace a failed or faulted device in one or more storage pools
- Designate these devices as hot spares:
  - When you create the pool (`zpool create`)
  - After you creating a pool (`zpool add`)

  Note: The size of the designated device must be equal to or larger than the size of the largest disk in the pool.
- After a failed device is replaced and resilvered, the spare is automatically detached and made available.
- An in-progress spare replacement can be canceled.
- If the faulted device is detached, the spare assumes its place and is removed from the spare's list of all active pools.

Designating Hot Spares in a Storage Pool

To designate hot spares in a pool that you are creating, use `zpool create` followed by the pool name, the configuration, the keyword `spare`, and the names of the spares.

```
# zpool create appool mirror c1t1d0 c2t1d0 spare c1t2d0 c2t2d0
# zpool status appool
pool: appool
 state: ONLINE
 scrub: none requested
config:
        NAME        STATE     READ   WRITE  CKSUM
        appool      ONLINE       0      0      0
          mirror-0  ONLINE       0      0      0
            c1t1d0  ONLINE       0      0      0
            c2t1d0  ONLINE       0      0      0
        spares
          c1t2d0    AVAIL
          c2t2d0    AVAIL
```

Designating Hot Spares in a Storage Pool

To add hot spares to an existing pool, use `zpool add` followed by the pool name, the keyword `spare`, and the names of the hot spares.

```
# zpool add appool spare c1t3d0 c2t3d0
# zpool status appool
pool: appool
 state: ONLINE
 scrub: none requested
config:
    NAME          STATE     READ   WRITE     CKSUM
    appool        ONLINE       0       0         0
      mirror-0    ONLINE       0       0         0
        c1t1d0    ONLINE       0       0         0
        c2t1d0    ONLINE       0       0         0
    spares
      c1t3d0      AVAIL
      c2t3d0      AVAIL
```

Designating Hot Spares in a Storage Pool

Example of a hot spare replacing a faulted device:

```
# zpool status appool
pool: appool
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas
    exist for the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
 scrub: resilvered completed 0h12m with 0 errors on Fri Dec 13 14:16:04
    2013
config:
    NAME          STATE     READ      WRITE    CKSUM
    appool        DEGRADED     0          0        0
      mirror-0    DEGRADED     0          0        0
       c1t1d0      ONLINE      0          0        0
       spare-1     UNAVAIL     0          0        0
         c2t1d0    UNAVAIL     0          0        0 cannot open
         c1t3d0    ONLINE      0          0        0 58.5K resilvered
    spares
      c1t3d0      INUSE      currently in use
      c2t3d0      AVAIL
```

Removing Hot Spares in a Storage Pool

To remove a hot spare, use `zpool remove` followed by the pool name and the name of the hot spare.

```
# zpool remove appool c1t2d0
# zpool status appool
pool: appool
(output omitted)
    spares
      c1t3d0      AVAIL
```

Note: You cannot remove a hot spare if it is currently being used by the storage pool as an active device.

Practice 4-1 Overview: Managing Data Redundancy with a ZFS Mirrored Pool

This practice covers the following topics:

- Creating ZFS mirrored pools
- Adding disks to a ZFS storage pool
- Adding a cache device to a ZFS storage pool
- Destroying a ZFS storage pool

Agenda

- Planning for data storage configuration and backup
- Managing data redundancy with mirrored storage pools
- Administering data backup and restore by using ZFS snapshots
- Managing data storage space with ZFS file system properties
- Troubleshooting ZFS failures

Backing Up and Recovering Data with ZFS Snapshots

This section covers the following topics:

- Creating and destroying a ZFS snapshot
- Displaying a ZFS snapshot
- Sending ZFS snapshot data
- Receiving ZFS snapshot data
- Remote replication of ZFS snapshot data

Creating and Destroying a ZFS Snapshot

To create a snapshot, use `zfs snapshot` followed by the snapshot name.

```
# zfs snapshot hrpool/home/qarpt@friday
```

To destroy a snapshot, use `zfs destroy` followed by the snapshot name.

```
# zfs destroy hrpool/home/qarpt@friday
```

Displaying a ZFS Snapshot

- To display only snapshots, use `zfs list -t snapshot`.

```
# zfs list -t snapshot
NAME                           USED AVAIL REFER  MOUNTPOINT
hrpool/home/qarpt@tuesday      18K      -    21K  -
hrpool/home/qarpt@wednesday    19K      -   280K  -
hrpool/home/qarpt@thursday      0       -   538K  -
```

- To list the snapshots created for a specific file system, enter `zfs list -r -t snapshot` followed by the file system name.

```
# zfs list -r -t snapshot -o name,creation hrpool/home
NAME                                CREATION
hrpool/home/qarpt@tuesday      Wed Dec 11 10:03 2013
hrpool/home/qarpt@wednesday    Thu Dec 12 10:03 2013
hrpool/home/qarpt@thursday     Fri Dec 13 10:03 2013
hrpool/home/bonus@now          Sat Dec 14 11:04 2013
```

- The snapshots of a file system are accessible in the `.zfs/snapshot` directory in the root of the file system.

```
# ls /home/qarpt/.zfs/snapshot
tuesday wednesday thursday
```

Identifying ZFS Snapshot Differences

- You can determine ZFS snapshot differences by using the `zfs diff` command.
- The `zfs diff` command gives a high-level description of the differences between a snapshot and a descendant dataset.
- The type of change is described along with the name of the file:
  - "+" indicates that the file was added in the later dataset.
  - "–" indicates that the file was removed in the later dataset.
  - "M" indicates that the file was modified in the later dataset.
  - "R" indicates that the file was renamed in the later dataset.

Identifying ZFS Snapshot Differences: Example

```
# zfs snapshot newpool/mydata@before
# touch /newpool/mydata/newfile
# zfs snapshot newpool/mydata@after

# zfs list -r -t snapshot -o name,creation
NAME                                                  CREATION
newpool/mydata@before                                 Fri Sep 11  4:02 2015
newpool/mydata@after                                  Fri Sep 11  4:02 2015
rpool/ROOT/solaris@install                            Tue Sep  1  5:14 2015
rpool/ROOT/solaris@2015-09-07-21:44:29                Mon Sep  7 13:44 2015
rpool/ROOT/solaris/var@install                        Tue Sep  1  5:14 2015
rpool/ROOT/solaris/var@2015-09-07-21:44:29            Mon Sep  7 13:44 2015
rpool/zones/zone1@send-to-nfs                         Thu Sep 10  9:47 2015
rpool/zones/zone1/rpool@send-to-nfs                   Thu Sep 10  9:47 2015
rpool/zones/zone1/rpool/ROOT@send-to-nfs              Thu Sep 10  9:47 2015
rpool/zones/zone1/rpool/ROOT/zbe-0@send-to-nfs        Thu Sep 10  9:47 2015
rpool/zones/zone1/rpool/ROOT/zbe-0/var@send-to-nfs    Thu Sep 10  9:47 2015
rpool/zones/zone1/rpool/export@send-to-nfs            Thu Sep 10  9:47 2015
rpool/zones/zone1/rpool/export/home@send-to-nfs       Thu Sep 10  9:47 2015
rpool/zones/zone2/rpool/ROOT/zbe-0@zflash.150910.09.34.33  Thu Sep 10  8:34 2015
rpool/zones/zone2/rpool/ROOT/zbe-0@zflash.150910.09.39.35  Thu Sep 10  8:39 2015

# zfs diff newpool/mydata@before newpool/mydata@after
M         /newpool/mydata/
+         /newpool/mydata/newfile
```

Sending ZFS Snapshot Data

The `zfs send` command:

- Is used to send ZFS snapshot data for backup purposes
- Sends a copy of a snapshot to another pool
  - On the same system
  - On a different system
- Creates a stream representation of a snapshot that is written to standard output
  - By default, a full stream is generated.
  - The output can be redirected to a file, to a different system, or to a device.
- Displays an estimate of the size of the send stream

Sending ZFS Snapshot Data

Remember the following key points when you send a file system snapshot:

- Use the `zfs send -I` option to send all incremental streams from one snapshot to a cumulative snapshot, or to send an incremental stream from the original snapshot to create a clone.
  The original snapshot must already exist on the receiving side to accept the incremental stream.
- When you use the `zfs send -r` option without the `-c` option and when you use the `zfs send -R` option, stream packages omit the origin of clones in some circumstances.
- Use the `zfs send -n` option with the `-v` option to see what snapshots would have been sent.
- Use the `zfs send -s` *`streamsize`* option to specify the size of the stream in bytes.
  - `-s` cannot be used with `–v`.
  - *`streamsize`* is the only valid option to `-s`.

Sending ZFS Snapshot Data

To send a ZFS snapshot, enter `zfs send` followed by the snapshot name and destination.

```
# zfs send hrpool/data@snap1 > /backup/hrdata
```

To send incremental ZFS snapshot data, use `zfs send -i`.

```
# zfs send -i hrpool/data@snap1 hrpool/data@snap2 > \
/backup/hrdata
```

Receiving ZFS Snapshot Data

The `zfs receive` command:

- Is used to receive ZFS snapshot data
- Receives the snapshot from:
  - Another pool
    - On the same system
    - On a different system
  - A file or device
- Creates a snapshot whose content is specified in the stream that is provided on standard input
- Has the alias `recv`

Note: If a full stream is received, a new file system is created as well.

Receiving ZFS Snapshot Data

Remember the following key points when you receive a file system snapshot:

- The snapshot and the file system are received.
- The file system and all the descendant file systems are unmounted.
- The file systems are inaccessible while they are being received.
- The original file system to be received must not exist while it is being transferred.
- If a conflicting file system name exists, `zfs rename` can be used to rename the file system.
- Use the `zfs send -R` option to send a replication stream of all descendant file systems. When the replication stream is received, all properties, snapshots, descendant file systems, and clones are preserved.

Receiving ZFS Snapshot Data

To receive a ZFS file system snapshot, use `zfs receive` followed by the snapshot name and the location from which you want to retrieve the file system.

```
# zfs send hrpool/jobdesc@1215 > /bkups/jobdesc.121511
# zfs receive hrpool/jobdesc2 < /bkups/jobdesc.121511
# zfs rename hrpool/jobdesc hrpool/jobdesc.old
# zfs rename hrpool/jobdesc2 hrpool/jobdesc
```

Remote Replication of ZFS Snapshot Data

To remotely copy snapshot data from one system to another system, use `zfs send` and `zfs receive`.

```
# zfs send hrpool/report@today | ssh newsys zfs recv sandbox/restfs
```

Practice 4-2 Overview: Using ZFS Snapshots for Backup and Recovery

This practice covers the following topics:

- Creating ZFS snapshots
- Sending ZFS snapshot data
- Receiving and saving snapshot data
- Rolling back to recover lost data

Agenda

- Planning for data storage configuration and backup
- Managing data redundancy with mirrored storage pools
- Administering data backup and restore by using ZFS snapshots
- Managing data storage space with ZFS file system properties
- Troubleshooting ZFS failures

Managing Data Storage Space with ZFS File System Properties

This section covers the following topics:

- Mounting and sharing ZFS file systems
  - Overriding a default ZFS mount point
  - Managing legacy mount points
  - Sharing and unsharing ZFS file systems
- Setting ZFS quotas and reservations

Overriding Default ZFS Mount Points

- By default, all ZFS file systems are mounted:
  - By ZFS at boot by using an SMF service
  - Under */path*, where *path* is the name of the file system
- The default mount point can be overridden by setting the `mountpoint` property to a specific path by using `zfs set`.
- When a default mount point is overridden, ZFS automatically:
  - Creates the mount point if needed
  - Mounts the file system to the new mount point

Note: There is no need to edit the `/etc/vfstab` file.

mountpoint Property

- Is inherited
- Can be set to none to prevent the file system from being mounted automatically
- Can be set to legacy to manage through legacy mount interfaces
  - This setting prevents ZFS from automatically mounting and managing the file system.
  - The file system must be managed by using legacy tools (mount, umount) and /etc/vfstab.

Note: To determine whether a file system can be mounted, check the value of the canmount property and the mountpoint property.

Automatic Mount Point Behavior

- When changing from a `legacy` or `none` mount point, ZFS automatically mounts the file system.
- If ZFS is managing the file system but is currently unmounted, and if the `mountpoint` property is changed, the file system remains unmounted.
- When the `mountpoint` property is changed, ZFS automatically reassigns the mount point.
- Mount point directories are created as needed.
- A dataset whose `mountpoint` property is not legacy is managed by ZFS.

Note: A default mount point can be created by using `zpool create -m`.

Legacy Mount Point Behavior

- Legacy file systems must be managed by using the `mount` and `umount` commands and the `/etc/vfstab` file.
- ZFS does not automatically mount legacy file systems on boot.
- The ZFS `mount` and `unmount` commands do not operate on legacy file systems.
- To automatically mount a legacy file system on boot, you must add an entry to the `/etc/vfstab` file.

Managing Legacy Mount Points

To manage ZFS file systems with legacy tools, use `zfs set` followed by `mountpoint=legacy` and the file system name.

```
# zfs set mountpoint=legacy hrpool/home/reports
```

To mount the file system, use `mount -F` followed by the file system type, the file system name, and a mount point.

```
# mount -F zfs hrpool/home/reports /mnt
```

share.nfs Property: Introduction

- ZFS automatically shares file systems by using the share.nfs property.
- The share.nfs property is a comma-separated list of options to pass to the share command.
  - The value on:
    - Is an alias for the default share options
    - Provides read/write permissions to anyone
  - The value off indicates that the file system is no longer shared.
- All file systems whose share.nfs property is not off are shared during boot.

Setting the share.nfs Property

To share a new file system, use the zfs set syntax (similar to the following example):

```
# zfs set share.nfs=on hrpool/home/reports
```

- The share.nfs property is inherited.
- Setting share.nfs to off:
  - Prevents a file system from automatically being shared
  - Allows the file system to be shared by using legacy methods

Unsharing ZFS File Systems

To explicitly unshare a file system, use `zfs unshare` followed by the file system name or mount point.

```
# zfs unshare hrpool/home/reports
```

To unshare all ZFS file systems, use `zfs unshare` with the `-a` option.

```
# zfs unshare -a
```

Sharing ZFS File Systems

To share a file system, use `zfs share` followed by the file system name.

```
# zfs share hrpool/home/reports
```

To share all ZFS file systems, use `zfs share` with the `-a` option.

```
# zfs share -a
```

Setting ZFS Quotas and Reservations

This section covers the following topics:

- Setting quotas on ZFS file systems
  - User quotas
  - Group quotas
- Setting default user and group quotas
- Displaying user and group space usage
- Removing user and group quotas
- Setting reservations on ZFS file systems

quota, reservation, refquota, refreservation, and used Properties

Property	Description
quota	Sets a limit on the pool space used by a file system
refquota	Sets the amount of disk space that a dataset can consume
reservation	Guarantees a specified amount of space for a file system from a pool
refreservaton	Sets the minimum amount of disk space that is guaranteed to a dataset, not including descendants (such as snapshots and clones)
used	Read-only property that identifies the amount of disk space consumed by a dataset and all its descendants

Setting Quotas for ZFS File Systems

To set a quota on a file system, use `zfs set` followed by `quota=`, the space amount, and the file system name.

```
# zfs set quota=10g hrpool/home/reports
```

To display the quota setting for a file system, use `zfs get` followed by `quota` and the file system name.

```
# zfs get quota hrpool/home/reports
NAME                  PROPERTY       VALUE    SOURCE
hrpool/home/reports   quota          10g      local
```

Note: The quota cannot be less than the current dataset usage.

Setting Quotas for ZFS File Systems

To limit the amount of disk space that a dataset can consume, use `zfs set` followed by `refquota=`, the space amount, and the dataset name.

```
# zfs set refquota=10g hrstaff/tsmith
# zfs list -t all -r hrstaff
NAME                          USED  AVAIL  REFER  MOUNTPOINT
hrstaff                       150M  66.8G    32K  /hrstaff
hrstaff/tsmith                150M  9.85G   150M  /hrstaff/tsmith
hrstaff/tsmith@yesterday         0      -   150M  -
# zfs snapshot hrstaff/tsmith@today
# zfs list -t all -r hrstaff
hrstaff                       150M  66.8G    32K  /hrstaff
hrstaff/tsmith                150M  9.90G   100M  /hrstaff/tsmith
hrstaff/tsmith@yesterday     50.0M      -   150M  -
hrstaff/tsmith@today             0      -   100M  -
```

Setting a User Quota on a ZFS File System

To set a user quota on a file system, use `zfs set` followed by `userquota@<`*name*`>=`, the space amount, and the file system name.

```
# zfs create finance/tax
# zfs set userquota@rsmart=10g finance/tax
```

To display the user quota setting for a file system, use `zfs get` followed by `userquota@<`*name*`>` and the file system name.

```
# zfs get userquota@rsmart finance/tax
NAME                PROPERTY             VALUE     SOURCE
finance/tax         userquota@rsmart     10g       local
```

Setting a Group Quota on ZFS File System

To set a group quota on a file system, use `zfs set` followed by `groupquota@<`*name*`>`=, the space amount, and the file system name.

```
# zfs create finance/ar
# zfs set groupquota@ar=20GB finance/ar
```

To display the group quota setting for a file system, use `zfs get` followed by `groupquota@<group>` and the file system name.

```
# zfs get groupquota@ar finance/ar
NAME                PROPERTY            VALUE     SOURCE
finance/ar          groupquota@ar       20G       local
```

Setting Default User and Group Quotas

- Default user or group quota is applied automatically for anyone who does not have a specific quota defined.
- Default user and group quotas are not inheritable to descendant file systems.
- Example
  - You can set a default user or group quota on a file system by:

```
# zfs set defaultuserquota=30gb students/labstaff/admindata
# zfs set defaultgroupquota=120g students/math
```

Displaying User and Group Space Usage

To display general user space usage, use `zfs userspace` followed by the file system name.

```
# zfs userspace finance/tax
TYPE          NAME      USED      QUOTA
POSIX User    root      227M      none
POSIX User    rsmart    455M      10g
```

To display general group space usage, use `zfs groupspace` followed by the file system name.

```
# zfs groupspace finance/ar
TYPE          NAME      USED      QUOTA
POSIX Group   root      217M      none
POSIX Group   ar        217M      20G
```

Identifying User and Group Space Usage

To identify individual user space usage, use `zfs userused@<name>` followed by the file system name.

```
# zfs get userused@rsmart finance/tax
NAME                PROPERTY            VALUE     SOURCE
finance/tax         userused@rsmart     455M      local
```

To identify group space usage, use `zfs groupused@<name>` followed by the file system name.

```
# zfs get groupused@ar finance/ar
NAME                PROPERTY            VALUE     SOURCE
finance/ar          groupused@ar        217M      local
```

Removing User and Group Quotas

To remove a user quota, use `zfs set userquota@<name>=none` followed by the file system name.

```
# zfs set userquota@rsmart=none finance/tax
```

To remove a group quota, use `zfs set groupquota@<name>=none` followed by the file system name.

```
# zfs set groupquota@ar=none finance/ar
```

Identifying Reservation Restrictions

- A ZFS *reservation* is an allocation of space from the pool that is guaranteed to be available to a dataset.
- Space cannot be reserved for a dataset if that space is not currently available in the pool.
- The total amount of all outstanding unconsumed reservations cannot exceed the amount of unused space in the pool.
- A dataset can use more space than it has reserved in the following situation:
  - Unreserved space is available in the pool.
  - Its current usage is below its quota.
- A dataset cannot consume space that is reserved for another dataset.

Setting Space Reservations on Datasets and Snapshots

To set a space reservation on a dataset and snapshot, use `zfs set` followed by `reservation`, the space amount, and the file system name.

```
# zfs set reservation=20g finance/ap
# zfs list
NAME                  USED      AVAIL     REFER     MOUNTPOINT
finance               20.0G     13.2G     19K       /finance
finance/ap            10g       33.2G     18K       /finance/ap
```

Setting a Space Reservation on a Dataset

To set a space reservation on a specific dataset, use `zfs set` followed by `refreservation=`, the space amount, and the file system name.

```
# zfs set refreservation=10g finance/ap
# zfs list
NAME              USED      AVAIL     REFER     MOUNTPOINT
finance           10.0G     23.2G     19K       /finance
finance/ap        10g       33.2G     18K       /finance/ap
```

- The space consumed by descendants, snapshots, and clones is not included.
- The setting counts against the parent dataset's quotas and reservation.

Displaying Reservation Values

To see the values of both reservations, use `zfs get` followed by `reservation`, `refreservation`, and the file system name.

```
# zfs get reservation,refreservation finance/ap
NAME            PROPERTY         VALUE     SOURCE
finance/ap      reservation      20G       local
finance/ap      refreservation   10g       local
```

Note: If `refreservation` is set, a snapshot is allowed only if enough free pool space exists outside this reservation to accommodate the current number of referenced bytes in the dataset.

Quiz

To set the amount of disk space that a dataset can consume, which of the following properties should you use?

a. refreservation
b. reservation
c. quota
d. refquota

Practice 4-3 Overview: Configuring ZFS Properties

This practice covers the configuration of:

- Quota and reservation properties
- The `share` property
- ZFS compression

Agenda

- Planning for data storage configuration and backup
- Managing data redundancy with mirrored storage pools
- Administering data backup and restore by using ZFS snapshots
- Managing data storage space with ZFS file system properties
- Troubleshooting ZFS failures

Troubleshooting ZFS Failures

- Identifying problems in ZFS
- Repairing:
  - A damaged ZFS configuration
  - A missing device
  - A damaged device
  - Damaged data

Identifying Problems in ZFS

This section covers the following topics:

- Overview of ZFS troubleshooting
- Basic recovery process
- Configuring `syslog` for FMD messages
- Determining problems in a ZFS storage pool
- Interpreting `zpool status` output

Troubleshooting in ZFS: Overview

The `zpool status` command is central to ZFS troubleshooting. This command does the following:

- Analyzes various failures in the system
- Identifies the most severe problem
- Presents:
  - A suggested action
  - A link to a knowledge article for more information
  - Only a single problem

Basic Recovery Process

1. Identify errors through the Fault Management Daemon (FMD) messages displayed in the system console or in `/var/adm/messages`.
2. Find further repair instructions in `zpool status -x`.
3. Repair the failures:
   - Replace the faulted or missing device and bring it online.
   - Restore the faulted configuration or corrupted data from a backup.
4. Verify the recovery by using `zpool status -x`.
5. Back up the restored configuration, if applicable.

Configuring syslog for FMD Messages

1. Create a new file named /var/adm/messages.fmd for Fault Management Daemon to log the device-related messages.
2. Back up the current /etc/syslog.conf file.
3. Edit the /etc/syslog.conf file by entering a new line below the existing line as follows:

```
*.err;kern.debug;daemon.notice;mail.crit          /var/adm/messages
daemon.err                                        /var/adm/messages.fmd
```

4. Restart the syslog service for the new configuration to take effect by using svcadm restart system-log.

Determining Problems in a ZFS Storage Pool

- Use `zpool status -x` to determine if a known problem exists.
- If no bad pools exist, the “all pools are healthy” status is returned.

```
# zpool status -x
all pools are healthy
```

- Without the `-x` flag, the status of all pools (regardless of health) is displayed.

Interpreting `zpool status` Output

Header section

```
# zpool status hrpool
  pool: hrpool
 state: DEGRADED
status: One or more devices has been taken offline by the
        administrator. Sufficient replicas exist for the pool to
        continue functioning in a degraded state.
action: Online the device using 'zpool online' or replace the
        device with 'zpool replace'.
 scrub: none requested
config:
        NAME            STATE       READ   WRITE  CKSUM
        hrpool          DEGRADED       0       0       0
          mirror-0      DEGRADED       0       0       0
            c1t0d0      ONLINE         0       0       0
            c1t1d0      OFFLINE        0       0       0
errors: No known data errors
```

Interpreting `zpool status` Output

Configuration (`config`) field: first section

```
# zpool status hrpool
  pool: hrpool
 state: DEGRADED
status: One or more devices has been taken offline by the
        administrator. Sufficient replicas exist for the pool to
        continue functioning in a degraded state.
action: Online the device using 'zpool online' or replace the
        device with 'zpool replace'.
 scrub: none requested
config:

        NAME            STATE     READ   WRITE  CKSUM
        hrpool          DEGRADED     0       0       0
          mirror-0      DEGRADED     0       0       0
            c1t0d0      ONLINE       0       0       0
            c1t1d0      OFFLINE      0       0       0

errors: No known data errors
```

Interpreting `zpool status` Output

Configuration (`config`) field: second section

```
# zpool status hrpool
  pool: hrpool
 state: DEGRADED
status: One or more devices has been taken offline by the
        administrator. Sufficient replicas exist for the pool to
        continue functioning in a degraded state.
action: Online the device using 'zpool online' or replace the
        device with 'zpool replace'.
 scrub: none requested
config:
        NAME          STATE     READ   WRITE  CKSUM
        hrpool        DEGRADED     0       0      0
          mirror-0    DEGRADED     0       0      0
            c1t0d0    ONLINE       0       0      0
            c1t1d0    OFFLINE      0       0      0

errors: No known data errors
```

Determining Problems in a ZFS Storage Pool

ZFS displays `syslog` messages for the following:

- Device state transition
- Data corruption
- Pool failures and device failures

Repairing a Damaged ZFS Configuration

- ZFS maintains a cache of active pools and their configuration on the root file system in `/etc/zfs/zpool.cache`.
- To recover the configuration, you can:
  - Export the pool (if it is visible at all)
  - Reimport it

Repairing a Missing Device

If a device cannot be opened, `UNAVAIL` is displayed in the `zpool status` output.

```
# zpool status hrpool
  pool: hrpool
state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://www.sun.com/msg/ZFS-8000-2Q
scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        hrpool        DEGRADED     0     0     0
          mirror-0  DEGRADED     0     0     0
            c1t0d0  ONLINE       0     0     0
            c1t1d0  UNAVAIL      0     0     0  cannot open

errors: No known data errors
```

Repairing a Missing Device

- An `UNAVAIL` status means that:
  - The device could not be opened when the pool was first accessed
  - The device has since become unavailable
- If the device causes a top-level virtual device to be unavailable, nothing in the pool can be accessed.
- To restore normal operation, reattach the device to the system.

Reattaching a Device

Device Type	Action
Network-attached drive	Restore connectivity.
USB or other removable media	Reattach to the system.
Local disk	Determine if it is a disk problem or a controller problem.

Repairing a Missing Device

- ZFS might not automatically detect device availability if:
  - The pool was degraded
  - The device was replaced while the system was up
- Use `zpool online` to notify ZFS that the device is now available and ready to be reopened.

```
# zpool online hrpool c0t1d0
```

Repairing a Damaged Device

This section covers the following topics:

- Determining the cause of device failure
- Clearing transient errors
- Replacing a device in a ZFS storage pool
- Viewing resilvering status

Determining the Cause of Device Failure

Possible causes of device failure:

- Bit rot
- Misdirected reads or writes
- Administrator error
- Temporary outage
- Bad or flaky hardware
- Offlined device

Determining the Cause of Device Failure

Use `zpool status -v` to examine the error counts.

```
# zpool status -v
  pool: hrpool
 state: UNAVAIL
status: One or more devices are faulted in response to IO failures.
action: Make sure the affected devices are connected, then run 'zpool
        clear'.
   see: http://www.sun.com/msg/ZFS-8000-HC
 scrub: scrub completed after 0h0m with 0 errors on Tue Oct  4 13:08:42 2011
config:

        NAME        STATE     READ WRITE CKSUM
        hrpool      UNAVAIL      0     0     0  insufficient replicas
          c1t0d0    ONLINE       0     0     0
          c1t1d0    UNAVAIL      4     1     0  cannot open

errors: Permanent errors have been detected in the following files:

/hrpool/data/aaa
/hrpool/data/bbb
/hrpool/data/ccc
```

Determining the Cause of Device Failure

Check the system log:

- If there are a large number of SCSI or Fibre Channel driver messages, the system has serious hardware problems.
- If no `syslog` messages are generated, the damage is likely to be transient.

Clearing Transient Errors

- To clear the error counters for RAID-Z or mirrored devices and to clear errors associated with the device, use `zpool clear` *`poolname devicename`*.

```
# zpool clear hrpool c1t0d0
```

- To clear errors associated with the virtual devices in the pool and to clear data error counts associated with the pool, use `zpool clear` *`poolname`*.

```
# zpool clear hrpool
```

Replacing a Device in a ZFS Storage Pool

- For a device to be replaced, the device must be part of a replicated configuration.
- The disk is part of a replicated configuration. Therefore, sufficient replicas from which to retrieve good data must exist.
- A device cannot be safely replaced if:
  - The loss of a device causes the pool to become faulted
  - The device contains too many data errors in an unreplicated configuration

Replacing a Device in a ZFS Storage Pool

- To replace a device with a new device in the same location, use `zpool replace` *`poolname devicename`*.

```
# zpool replace hrpool c1t0d0
```

- To replace a damaged device with a different device, use `zpool replace` *`poolname devicename devicename`*.

```
# zpool replace hrpool c1t0d0 c2t0d0
```

Viewing Resilvering Status

Resilvering:

- Is the process of moving data from one device to another
- Is monitored by using `zpool status`
- Resilvers only the minimum amount of necessary data
- Is interruptible and safe

Scrubbing

- Examines all data to discover silent errors due to hardware faults or disk failure
- Supports automatic repair of damage discovered during the scrub
- Is monitored by using `zpool status`
- Cannot be run if:
  - Another scrub is already in progress
  - A resilver is in progress
- Is begun by using *`zpool scrub pool`*

Repairing Damaged Data

This section covers the following topics:

- Overview of data corruption
- Identifying the type of data corruption
- Repairing a corrupted file or directory
- Repairing damage to a ZFS storage pool

Data Corruption: Overview

- Data corruption can occur if:
  - The pool is not replicated
  - Corruption occurred while the pool was degraded
  - An unlikely series of events caused the corruption of multiple copies of a piece of data
- Two basic types of data can be corrupted:
  - Pool metadata
  - Object data

Identifying the Type of Data Corruption

Use `zpool status -v` *`poolname`* to identify the type of data corruption.

Example of object data corruption:

```
# zpool status -v hrpool
  pool: hrpool
 state: ONLINE
status: One or more devices has experienced an error resulting in data
        corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore
        the entire pool from backup.
   see: http://www.sun.com/msg/ZFS-8000-8A
<output omitted>
errors: Permanent errors have been detected in the following files:
/hrpool/data/abc
/hrpool/data/def.txt
/hrpool/data/ghi.txt
```

Identifying the Type of Data Corruption

Example of pool metadata corruption:

```
# zpool status -v sales
pool: sales
   id: 1422736890544688191
 state: FAULTED
status: The pool metadata is corrupted.
action: The pool cannot be imported due to damaged devices or data.
   see: http://www.sun.com/msg/ZFS-8000-72
config:
        sales FAULTED corrupted data
        c1t1d0 ONLINE
```

Repairing a Corrupted File or Directory

- The system may still be able to function.
- Any damage is effectively unrecoverable.
- No good copies of the data exist anywhere on the system.
- If the data is valuable, restore the affected data from backup.
- If the damage is in a file data block, remove the file.

Repairing Damage to a ZFS Storage Pool

If you cannot open or import a pool because of damage to the pool metadata, you must perform one of the following actions:

- Attempt to recover the pool by using `zpool clear -F` *`poolname`* or `zpool import -F` *`poolname`*.
  - An attempt is made to roll back to an operational state.
  - To review a damaged pool and see recommended recovery steps, use `zpool status`.
- Restore the pool and all its data from a backup copy.
  - Save the pool configuration as displayed in `zpool status`.
  - Destroy the pool by using `zpool destroy -f` *`poolname`*.
  - Keep a file of the dataset layout and local property settings.
  - Reconstruct the complete pool configuration.
  - Populate the data by using the backup / restore strategy.

Practice 4-4 Overview: Troubleshooting ZFS Failures

This practice covers the troubleshooting of:

- ZFS device issues
- ZFS data errors in a mirror pool

Summary

In this lesson, you should have learned how to:

- Plan for data storage configuration and backup
- Manage data redundancy with mirrored storage pools
- Back up and restore data by using ZFS snapshots
- Manage data storage space with ZFS file system properties
- Troubleshoot ZFS failures