Oracle Solaris 11 ZFS Administration

Student Guide

D75915GC10
Edition 1.0
August 2012
D78775

Authors

Anies Rahman

Tammy Shannon

Technical Contributors and Reviewers

Lori Alt

Harry Burks

Susan Chang

John Hathaway

Kristi Herd

Dominic Kay

Rosemary Martinak

Kristi McNeill

Lynn Rohr

Stephanie Scheffler

Cindy Swearingen

Editors

Daniel Milne

Richard Wallis

Anwesha Ray

Graphic Designer

Seema M Bopaiah

Publishers

Syed Imtiaz Ali

Veena Narasimhan

Contents

Preface

1 Introduction
Overview 1-2
Course Goals 1-3
Course Agenda: Day 1 1-4
Course Agenda: Day 2 1-6
Course Agenda: Day 3 1-7
Course Agenda: Day 4 1-8
Introductions 1-9
Your Learning Center 1-10
Your Practice Environment 1-11

2 Oracle Storage-Related Technologies and the Role of ZFS
Objectives 2-2
Lesson Agenda 2-3
Current Oracle Storage-Related Technologies 2-4
Role of ZFS in Data Management and Storage 2-5
Sun ZFS Storage Appliance 2-6
Sun ZFS Storage Appliance: Architecture 2-7
Sun ZFS Storage Appliance: Capabilities 2-10
Sun ZFS Storage Appliance: Browser User Interface 2-11
Sun ZFS Storage Appliance: Command-Line Interface 2-12
Sun ZFS Storage Appliance: Benefits 2-13
Sun ZFS Storage Appliance: Usage 2-14
Sun ZFS Storage Appliance and ZFS 2-15
Quiz 2-16
Additional Resources for the Sun ZFS Storage Appliance 2-17
Lesson Agenda 2-18
Pillar Axiom Technologies: Overview 2-19
Pillar Axiom Architecture: Overview 2-20
Pillar Axiom Hardware: Overview 2-21
Pillar Axiom Software: Overview 2-22
Pillar Axiom Storage System: Infrastructure 2-23
Pillar Axiom Storage Domains and Volumes 2-24

Pillar Axiom Technology: Benefits 2-25
Pillar Axiom Technology: Usage 2-26
Pillar Axiom Technology and ZFS 2-27
Quiz 2-28
Additional Resources for Pillar Axiom 2-29
Lesson Agenda 2-30
Cloud Computing 2-31
Cloud Architecture 2-32
Cloud Service Models 2-33
Cloud Service Models: Summary 2-35
Cloud Deployment Models 2-36
Cloud Computing: Benefits 2-37
Who Uses the Cloud? 2-38
Cloud Computing and ZFS 2-39
Quiz 2-40
Lesson Agenda 2-41
COMSTAR and iSCSI 2-42
COMSTAR 2-43
COMSTAR: Requirements 2-44
Configuring COMSTAR 2-45
COMSTAR: Benefits 2-46
COMSTAR: Usage 2-47
COMSTAR and ZFS 2-48
Quiz 2-49
Additional Resources for COMSTAR 2-51
Current Oracle Storage-Related Technologies 2-52
Summary 2-53

3 Planning for Data Management
Objectives 3-2
Lesson Agenda 3-3
Storage Requirements for the Business Application: Overview 3-4
Business Applications 3-5
Storage Requirements for the Outstanding Property Management Company 3-7
Storage Requirements for Fantastic Products LLC 3-9
Quiz 3-11
Lesson Agenda 3-12
Root Pool Management: Requirements 3-13
ZFS Root Pool Storage 3-14
Managing Swap and Dump Devices 3-15
Managing Swap and Dump Devices: Best Practice 3-16

Swap and Dump Storage: Requirements 3-17
Using an Alternate Disk for Booting 3-18
Alternate ZFS Root File System for Booting 3-19
Quiz 3-20
Lesson Agenda 3-21
ZFS Hierarchy: Requirements 3-22
Data Sharing in the Non-Global Zone 3-23
ZFS Data Encryption: Requirements 3-24
ZFS Data Migration: Requirements 3-25
ZFS Component Upgrades: Requirements 3-26
Quiz 3-27
Lesson Agenda 3-28
ZFS Properties: Requirements 3-29
Quiz 3-30
Lesson Agenda 3-31
Data Protection Requirements Using ZFS ACLs 3-32
Quiz 3-33
Lesson Agenda 3-34
ZFS Delegated Administration Model: Requirements 3-35
Quiz 3-36
Lesson Agenda 3-37
Business Data: Backup and Recovery Requirements 3-38
Quiz 3-39
Practice 3-1 Overview: Determining the Existing Configuration 3-40
Summary 3-41

4 Managing the ZFS Root Pool
Objectives 4-2
Planning Workflow: Orientation 4-3
Lesson Agenda 4-4
Implementing the Plan to Manage the ZFS Root Pool 4-5
Lesson Agenda 4-6
Managing the ZFS Swap and Dump Devices 4-7
Inspecting the Swap and Dump Devices 4-8
Adjusting the Size of the ZFS Swap Device 4-9
Adjusting the Size of the ZFS Swap Device: Example 4-11
Adjusting the Size of the ZFS Dump Device 4-13
Adjusting the Size of the ZFS Dump Device: Example 4-14
Adding a New ZFS Swap Volume 4-15
Adding a New ZFS Swap Volume: Example 4-16
Removing a ZFS Swap Volume 4-17

Removing a ZFS Swap Volume: Example 4-18
Removing a ZFS Dump Device 4-19
Removing a ZFS Dump Device: Example 4-20
Troubleshooting ZFS Dump Device Issues 4-21
Quiz 4-23
Practice 4-1 Overview: Managing the ZFS Swap and Dump Devices 4-27
Lesson Agenda 4-28
Booting from an Alternate ZFS Root File System 4-29
Inspecting the Current Boot Environment 4-30
Creating a Mirrored ZFS Root Pool 4-31
Booting from an Alternate Disk in a Mirrored ZFS Root Pool 4-32
Booting from an Alternate ZFS Root File System on an x86-Based System 4-33
Booting from an Alternate ZFS Root File System on a SPARC-Based System 4-34
Booting in a ZFS Root Environment for Recovery Purposes 4-35
Quiz 4-37
Practice 4-2 Overview: Booting from an Alternate ZFS Root File System 4-39
Summary 4-40

5 Configuring and Managing the Oracle Solaris ZFS Components
Objectives 5-2
Planning Workflow: Orientation 5-3
Lesson Agenda 5-4
Implementing the Plan to Manage the ZFS Hierarchy 5-5
Lesson Agenda 5-6
Configuring the ZFS Hierarchy for a Business Application 5-7
Reviewing the ZFS Hierarchy Planning Requirements 5-8
Creating a Mirrored Storage Pool 5-10
Creating a Mirrored Storage Pool: Example 5-11
Creating a RAID-Z Storage Pool 5-12
Creating a RAID-Z Storage Pool: Example 5-13
Creating the ZFS File Systems 5-14
Populating the ZFS File Systems with Application Data 5-15
Quiz 5-16
Practice 5-1 Overview: Configuring the ZFS Storage Hierarchy 5-18
Lesson Agenda 5-19
Sharing ZFS File Systems in a Non-Global Zone 5-20
Creating a Non-Global Zone 5-21
Creating a Non-Global Zone: Example 5-23
Delegating a ZFS Dataset to a Non-Global Zone 5-25
Delegating a ZFS Dataset to a Non-Global Zone: Example 5-26
Adding a ZFS File System to a Non-Global Zone 5-27

Adding a ZFS File System to a Non-Global Zone: Example 5-29
Quiz 5-30
Practice 5-2 Overview: Sharing ZFS File Systems in a Non-Global Zone 5-32
Lesson Agenda 5-33
Encrypting ZFS Data 5-34
Securing ZFS Data with ZFS Encryption 5-35
Oracle Solaris 11 Encryption Methods 5-36
Wrapping Keys 5-37
Encrypting and Decrypting a ZFS File 5-39
Encrypting and Decrypting a ZFS File: Example 5-40
Encrypting a ZFS Storage Pool 5-41
Encrypting a ZFS File System 5-42
Quiz 5-44
Practice 5-3 Overview: Encrypting ZFS Data 5-46
Lesson Agenda 5-47
Migrating ZFS Data 5-48
Migrating a ZFS Storage Pool 5-49
Migrating a ZFS Storage Pool: Example 5-51
Migrating File Systems by Using Shadow Migration 5-52
Migrating a Remote ZFS File System: Example 5-54
Sharing ZFS File Systems with SMB 5-55
Behavior of sharesmb Property 5-57
Quiz 5-58
Practice 5-4 Overview: Migrating ZFS Data 5-60
Lesson Agenda 5-61
Upgrading ZFS Components 5-62
Upgrading ZFS Storage Pools 5-63
Upgrading ZFS File Systems 5-64
Quiz 5-65
Practice 5-5 Overview: Upgrading ZFS Components 5-66
Summary 5-67

6 Configuring ZFS Properties
Objectives 6-2
Planning Workflow: Orientation 6-3
Lesson Agenda 6-4
Implementing the Plan to Configure ZFS Properties 6-5
Lesson Agenda 6-6
Configuring ZFS Storage Pool Properties 6-7
Reviewing the listsnapshots Property Requirements 6-8
listsnapshots Property: Overview 6-9

Setting the listsnapshots Property 6-11
Setting the listsnapshots Property: Example 6-12
Reviewing the autoexpand Property Requirements 6-13
autoexpand Property: Overview 6-14
Setting the autoexpand Property 6-15
Setting the autoexpand Property: Example 6-16
Setting the autoreplace Property 6-17
autoreplace Property: Overview 6-18
Setting the autoreplace Property 6-19
Setting the autoreplace Property: Example 6-20
Reviewing the bootfs Property Requirements 6-21
bootfs Property: Overview 6-22
Setting the bootfs Property 6-23
Setting the bootfs Property: Example 6-24
Quiz 6-26
Practice 6-1 Overview: Configuring ZFS Pool Properties 6-27
Lesson Agenda 6-28
Configuring ZFS File System Properties 6-29
Reviewing the Quota and Reservation Property Requirements 6-30
Quota Properties: Overview 6-31
Setting the quota Property 6-32
Setting the quota Property: Example 6-33
Setting the User Quota Property 6-34
Setting the Group Quota Property 6-36
Identifying User and Group Space Usage 6-37
Displaying User and Group Space Usage 6-39
Setting the User and Group Properties: Example 6-40
Removing Quotas 6-41
ZFS reservation Property: Overview 6-42
Setting the reservation Property 6-43
Setting the reservation Property: Example 6-44
Reviewing the compression Property Requirements 6-45
ZFS compression Property: Overview 6-46
Setting the ZFS compression Property 6-48
Reviewing the File Change Ownership Property Requirements 6-49
ZFS File Change Ownership Property: Overview 6-50
Setting the ZFS File Change Ownership Property 6-51
Working with the ZFS File Change Ownership Property: Example 6-52
Quiz 6-53
Practice 6-2 Overview: Configuring ZFS File System Properties 6-57
Lesson Agenda 6-58

Managing ZFS Properties in a Non-Global Zone 6-59
zoned Property: Overview 6-60
Working with the zoned Property 6-61
checksum Property: Overview 6-62
Verifying the checksum Property Setting 6-63
Verifying the checksum Property Setting: Example 6-64
Quiz 6-65
Practice 6-3 Overview: Configuring ZFS Properties in a Non-Global Zone 6-67
Summary 6-68

7 Protecting Data by Using ZFS ACLs
Objectives 7-2
Planning Workflow: Orientation 7-3
Lesson Agenda 7-4
Implementing the Plan for Data Protection by Using ZFS ACLs 7-5
New Solaris ACL Model: Overview 7-6
ACL Formats 7-7
Trivial File ACL in Verbose and Compact Format: Examples 7-9
Non-Trivial Directory ACL in Verbose and Compact Format: Examples 7-11
ACLs and umask Settings 7-12
ACL Entry Types 7-13
ACL Access Privileges 7-14
ACL Inheritance: Overview 7-16
ACL Inheritance Flags 7-17
ZFS ACL Properties 7-18
Primary Rules of ACL Access on a ZFS File 7-20
Quiz 7-21
Lesson Agenda 7-25
Configuring and Managing ZFS ACLs 7-26
Adding an ACL Entry 7-27
Inserting an ACL Entry by Index ID 7-28
Replacing an ACL Entry by Index ID 7-29
Replacing an Entire ACL Entry 7-30
Removing an ACL Entry 7-31
Removing an ACL Entry by Index ID 7-32
Removing All ACL Entries from a File 7-33
Setting the ACL Inheritance file_inherit Flag 7-34
Setting the ACL Inheritance dir_inherit Flag 7-35
ACL Inheritance and ZFS ACL Properties 7-36
Setting the ZFS ACL aclinherit Property: Example 7-37
Quiz 7-38

Reviewing the ACL Requirements 7-40
Configure ACLs for Directories and Files: Example 7-41
Managing ZFS ACLs: Example 7-42
Configuring ACLs in Compact Format: Example 7-45
Practices 7-1 and 7-2 Overview: Configuring ACLs for Directories and Files and Managing ZFS ACLs 7-46
Summary 7-47

8 Using ZFS Delegated Administration
Objectives 8-2
Planning Workflow: Orientation 8-3
Lesson Agenda 8-4
Implementing the Plan for ZFS Delegated Administration 8-5
Reviewing the Plan for ZFS Delegated Administration 8-6
ZFS Rights Profile: Overview 8-7
ZFS Delegated Administration Model: Overview 8-8
ZFS Delegated Permissions (Sampling) 8-9
ZFS Delegated Properties 8-10
Quiz 8-11
Lesson Agenda 8-14
Configuring the ZFS Delegated Administration Model 8-15
Configuring a ZFS Rights Profile 8-16
Configuring a ZFS Rights Profile: Example 8-17
Enabling and Disabling the delegation Property 8-18
Delegating ZFS Permissions 8-19
Delegating a ZFS Permission Set 8-20
Delegating a ZFS Permission Set: Example 8-21
Removing ZFS Delegated Permissions 8-22
Removing ZFS Delegated Permission Sets 8-23
Quiz 8-24
Practices 8-1 and 8-2 Overview: Configuring a ZFS Rights Profile and Delegating ZFS Permissions 8-27
Summary 8-28

9 Backing Up and Recovering ZFS Data
Objectives 9-2
Planning Workflow: Orientation 9-3
Lesson Agenda 9-4
Implementing a Plan for Backing Up and Restoring ZFS Data 9-5
Reviewing the Plan for Backing Up and Recovering ZFS Data 9-6
Solutions for Backing Up and Recovering ZFS Data 9-7

ZFS Snapshot Streams: Overview 9-9
Receiving a ZFS Snapshot Stream 9-11
Advantages and Disadvantages of Using ZFS Snapshot Streams for Backup and Recovery 9-12
File System Backup and Recovery with Snapshots 9-14
Automating ZFS Snapshots: Overview 9-16
Clone Creation and Promotion: Overview 9-18
Quiz 9-20
Lesson Agenda 9-22
Backing Up ZFS Data by Using Snapshots 9-23
Using Snapshots for Backup and Recovery 9-24
Using Snapshots for Backup and Recovery: Example 9-25
Differentiating Between Two Snapshots 9-26
ZFS Snapshot Difference Identifiers 9-27
Creating and Promoting a Clone 9-28
Creating and Promoting a Clone: Example 9-29
Enabling and Scheduling Automatic Snapshots by Using the Command Line 9-30
Modifying Automatic Snapshots Scheduling by Using the Command Line 9-31
Deleting Unwanted Automatic Snapshots by Using the Command Line 9-32
Enabling and Scheduling Automatic Snapshots by Using the GUI 9-33
Expanded View of the Time Slider Manager 9-35
Enabling and Scheduling Automatic Snapshots by Using the GUI 9-36
Browsing and Recovering Automatic Snapshots by Using the GUI 9-37
Deleting Unwanted Automatic Snapshots by Using the GUI 9-43
Deleting Unwanted Automatic Snapshots by Using the GUI: Example 9-44
Enabling and Scheduling Automatic Snapshots by Using the Command Line: Example 9-45
Quiz 9-47
Practice 9-1 Overview: Backing Up ZFS Data by Using Snapshots 9-49
Lesson Agenda 9-50
Backing Up and Restoring ZFS Data Remotely 9-51
Sending and Receiving a ZFS Snapshot Stream 9-52
Sending and Receiving a ZFS Snapshot Stream: Example 9-53
Applying ZFS Properties to a ZFS Snapshot Stream: Overview 9-54
Options for Applying ZFS Properties to ZFS Snapshot Streams 9-55
Using a ZFS Snapshot Recovery Archive 9-56
Using a ZFS Snapshot Recovery Archive: Example 9-58
Quiz 9-60
Practice 9-2 Overview: Backing Up ZFS Data Remotely 9-62
Summary 9-63

10 ZFS Data Management Challenge

Objectives 10-2
Planning Workflow: Orientation 10-3
Reviewing the Requirements for the ZFS Data Management Challenge 10-4
Practice 10-1 Overview: ZFS Data Management Challenge 10-5
Summary 10-6

Preface

Profile

Before You Begin This Course

Before you begin this course, you should be able to perform advanced Oracle Solaris 11 system administration tasks.

How This Course Is Organized

Oracle Solaris 11 ZFS Administration is an instructor-led course featuring lectures and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills that are introduced.

Related Publications

Oracle Publications

Title	Part Number
N/A	N/A

Additional Publications

- System release bulletins
- Installation and user's guides
- *read.me* files
- International Oracle User's Group (IOUG) articles
- *Oracle Magazine*

Typographic Conventions

The following two lists explain Oracle University typographical conventions for words that appear within regular text or within code samples.

1. Typographic Conventions for Words Within Regular Text

Convention	Object or Term	Example
Courier New	User input; commands; column, table, and schema names; functions; PL/SQL objects; paths	Use the `SELECT` command to view information stored in the `LAST_NAME` column of the `EMPLOYEES` table. Enter `300`. Log in as `scott`
Initial cap	Triggers; user interface object names, such as button names	Assign a When-Validate-Item trigger to the ORD block. Click the Cancel button.
Italic	Titles of courses and manuals; emphasized words or phrases; placeholders or variables	For more information on the subject see *Oracle SQL Reference Manual* Do *not* save changes to the database. Enter *`hostname`*, where *`hostname`* is the host on which the password is to be changed.
Quotation marks	Lesson or module titles referenced within a course	This subject is covered in Lesson 3, “Working with Objects.”

2. Typographic Conventions for Words Within Code Samples

Convention	Object or Term	Example
Uppercase	Commands, functions	`SELECT employee_id` `FROM   employees;`
Lowercase, italic	Syntax variables	`CREATE ROLE` *`role`*`;`

Initial cap	Forms triggers	`Form module: ORD`
		`Trigger level: S_ITEM.QUANTITY item`
		`Trigger name: When-Validate-Item`
		`. . .`

Lowercase	Column names, table names, filenames, PL/SQL objects	`. . .`
		`OG_ACTIVATE_LAYER`
		` (OG_GET_LAYER ('prod_pie_layer'))`
		`. . .`
		
		`SELECT last_name`
		`FROM   employees;`

Bold	Text that must be entered by a user	`CREATE USER` **`scott`** `IDENTIFIED BY` **`tiger`**`;`

1 Introduction

Overview

- Course goals
- Course agenda
- Introductions
- Your learning center
- Your practice environment

Welcome to the *Oracle Solaris 11 ZFS Administration* course. This specialized course is designed to teach additional skills to experienced system administrators who are able to perform advanced Oracle Solaris 11 system administration tasks.

Course Goals

The goals of this course are to:

- Enable you to perform advanced ZFS administration tasks in Oracle Solaris 11 successfully and efficiently
- Increase your knowledge and skill in the following areas:
  - Oracle Solaris 11 ZFS components management
  - ZFS root pool management
  - Data protection using ZFS access control lists
  - ZFS delegated administration
  - ZFS data backup and recovery
- Build your ZFS administration skills with numerous and meaningful practice opportunities

Course Agenda: Day 1

- Lesson 1: Introduction
- Lesson 2: Oracle Storage-Related Technologies and the Role of ZFS
  - ZFS Storage Appliance
  - Pillar Axiom Storage Technology
  - Cloud Technology
  - COMSTAR technology
- Lesson 3: Planning for Data Management

Oracle Solaris 11 ZFS Administration consists of four days of lecture and practices.

On this first day, Lesson 2 introduces you to four of Oracle's current storage-related technologies and shows how ZFS relates to each of them. Although the focus of this course is managing data in Oracle Solaris 11 by using ZFS, you should be aware of Oracle's latest offerings in the data storage space. By being more familiar with each of these technologies and the role ZFS plays, you will be better equipped to determine which offerings your company can use to meet your specific data management and storage requirements.

In Lesson 3, we discuss how to plan for data management for a business application. The plan touches on all the subsequent topic areas in the course, such as ZFS root pool management, ZFS component configuration, ZFS properties, ZFS ACLS, ZFS delegated administration, and data backup and recovery.

Course Agenda: Day 1

- Lesson 4: Managing the ZFS Root Pool
  - Implementing the Plan to Manage the ZFS Root Pool
  - Managing the ZFS Swap and Dump Devices
  - Booting From a ZFS Root File System

In Lesson 4, you implement the plan to manage all aspects of the ZFS root pool facilities, including the swap and dump devices. You learn how to boot from a ZFS root file system and from an alternate disk.

Course Agenda: Day 2

- Lesson 5: Configuring and Managing the Oracle Solaris ZFS Components
  - Implementing the Plan to Manage the ZFS Hierarchy
  - Configuring the ZFS Hierarchy for a Business Application
  - Sharing ZFS File Systems in a Non-Global Zone
  - Encrypting ZFS File Systems
  - Migrating ZFS Data
  - Upgrading ZFS Components
- Lesson 6: Configuring ZFS Properties
  - Implementing the Plan to Configure ZFS Properties
  - Configuring ZFS Storage Pool Properties
  - Configure ZFS File System Properties
  - Managing ZFS Properties Within a Non-Global Zone

ORACLE

In Lesson 5, you configure critical components of the ZFS hierarchy, such as ZFS pools and file systems that you use to store the data for a business application. You learn how to share file systems in a non-global zone as well as how to encrypt file systems. You also learn about shadow migration, which helps you migrate data from other file systems (such as UFS and CIFS) to ZFS data. The lesson concludes with a discussion about upgrading ZFS pools and file systems.

In Lesson 6, you configure ZFS storage pool, file system, and non-global zone properties according to the data management plan.

Course Agenda: Day 3

- Lesson 7: Protecting Data by Using ZFS ACLs
  – Implementing the Plan for Data Protection by Using ZFS ACLs
  – Configuring and Managing ZFS ACLs
- Lesson 8: Using ZFS Delegated Administration
  – Implementing the Plan for ZFS Delegated Administration
  – Configuring the ZFS Delegated Administration Model

In Lesson 7, you learn how to protect the business data in ZFS files by using access control lists (ACLs). You learn how to grant specific file permissions to a user. We also discuss the inheritance function, which saves resources.

In Lesson 8, we discuss ZFS delegated administration. You learn how to grant and revoke the ZFS administration privileges of a user. You also learn how to create a ZFS rights profile, which can be customized for a set of privileges.

Course Agenda: Day 4

- Lesson 9: Backing Up and Recovering ZFS Data
  - Implementing a Plan for Backing Up and Restoring ZFS Data
  - Backing Up ZFS Data by Using Snapshots
  - Backing Up and Restoring ZFS Data Remotely
- Lesson 10: ZFS Data Management Challenge
  - Scenario-based exercise
  - No step-by-step instructions or solutions

Note: Class starts at 9 AM and ends at 5 PM each day. There are several short breaks throughout the day, with an hour for lunch.

In Lesson 9, you are introduced to multiple ZFS backup and restore facilities. We discuss ZFS snapshots as a backup technique and the snapshot rollback utility as a means of restoring backups. You also learn how to use the `zfs send` and `receive` commands to store the application data locally or remotely. Because storing the data remotely is useful for disaster recovery purposes, you are also shown how to create an archive of a ZFS pool and how to recover the pool from the archive.

Lesson 10 consists of a challenge that is designed to test what you have learned about data management using ZFS. In a scenario that is similar to the scenario you use throughout the course, you complete a comprehensive practice without the aid of step-by-step instructions or documented solutions.

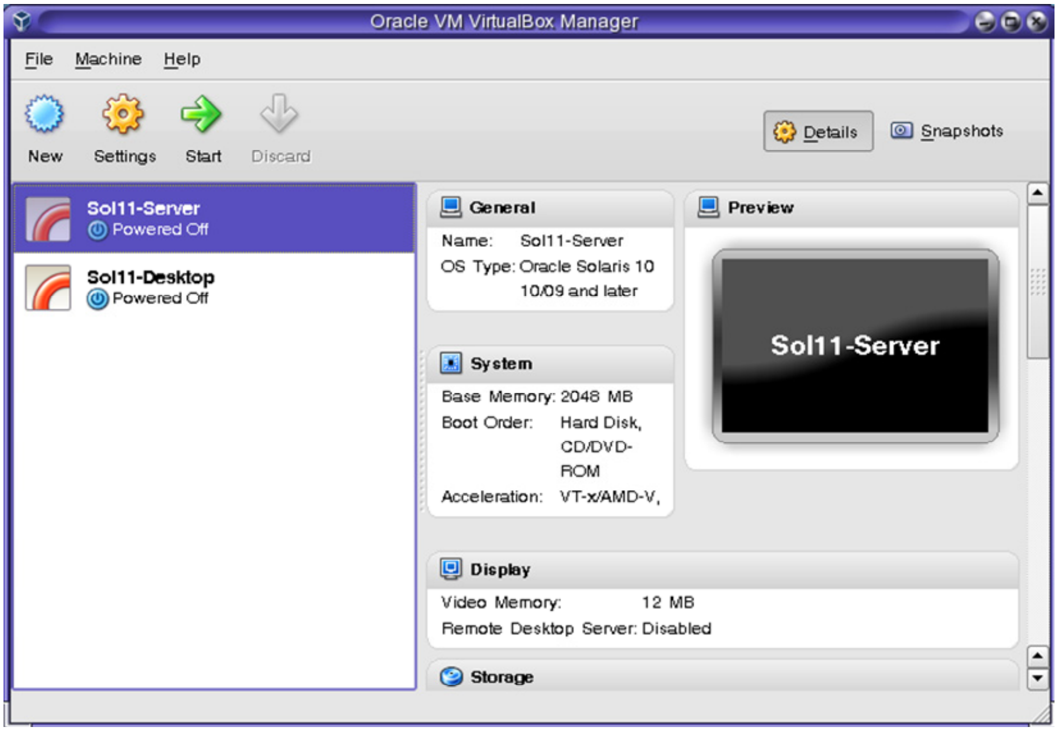
Introductions

- Name
- Company affiliation
- Title, function, and job responsibility
- Experience related to the topics presented in this course
- Reasons for enrolling in this course
- Expectations for this course

Your Learning Center

- Logistics
  - Restrooms
  - Break rooms and designated smoking areas
- Cafeterias and restaurants in the area
- Emergency evacuation procedures
- Instructor contact information
- Cell phone usage
- Online course attendance confirmation form

Your Practice Environment



ORACLE

The practice environment that we use in this course is based on the Oracle VM VirtualBox virtualization software (an example of which is shown in the slide). VirtualBox is a cross-platform virtualization application that extends the capabilities of your existing computer so that it can run multiple operating systems (inside multiple virtual machines) simultaneously.

Open your *Activity Guide* to "Practices for Lesson 1: Introduction." Your instructor will walk you through the material, and you will have a chance to familiarize yourself with the practice environment configuration and setup.

2

Oracle Storage-Related Technologies and the Role of ZFS

Objectives

After completing this lesson, you should be able to describe:

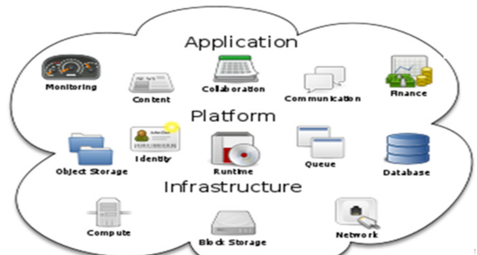
- Sun ZFS Storage Appliance
- Pillar Axiom technology
- Cloud technology
- COMSTAR technology

Lesson Agenda

- Sun ZFS Storage Appliance
- Pillar Axiom Technology
- Cloud Technology
- COMSTAR Technology

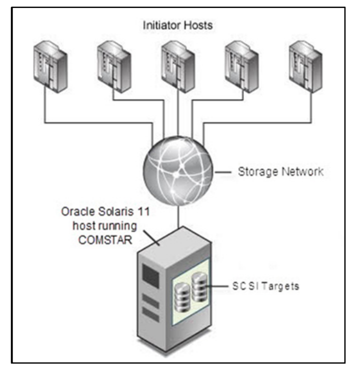
Current Oracle Storage-Related Technologies

Sun ZFS Storage Appliance



Cloud Computing

Pillar Axiom



COMSTAR

As we mentioned in the introduction to this course, you should be aware of Oracle's latest offerings in the data storage space and the role that Oracle Solaris 11—and specifically ZFS—plays in this area. Being familiar with each of these technologies will enable you to determine which option is better suited to the data management and storage needs of your particular business.

In this lesson, you are introduced to Oracle's latest generation of network attached storage (NAS) and storage area network (SAN) storage products:

- ZFS Storage Appliance for NAS storage
- Pillar Axiom storage systems for SAN and NAS storage

These products have been engineered to enable Oracle software to run faster and more efficiently.

In addition, we discuss cloud computing and how you can use ZFS to manage applications in a cloud infrastructure. We conclude the lesson with a look at COMSTAR (Common Multiprotocol SCSI Target), a framework that enables you to make storage devices on a system available to Linux, Mac OS, and Windows client systems as if they were local storage devices.

Role of ZFS in Data Management and Storage

- ZFS can manage a single system with a limited number of disks.
  - Example: In our training environment, ZFS manages between 10 and 12 disks.
- ZFS is powerful enough to manage business data stored on large storage devices.
  - Example: In the ZFS Storage Appliance environment, ZFS manages data and storage with specific utilities.
- ZFS can manage storage networks.
  - Example: In NAS and SAN environments, ZFS controls data and storage.

Before we begin our look at each of the storage-related technologies, let's spend just a moment positioning ZFS in the context of data management and storage. ZFS is extremely versatile. You can use ZFS to manage a single system with a limited number of disks, as we have in our training environment.

On the other hand, ZFS is powerful enough that you can use it to manage business data stored on large storage devices. For example, as you will see shortly, ZFS is used to manage data and storage in the ZFS Storage Appliance environment.

You can also use ZFS to manage data and storage in NAS and SAN environments.

In short, ZFS can play several roles in different types of environments, depending on what you need it to do.

Sun ZFS Storage Appliance

What is a ZFS Storage Appliance?

Oracle's Sun ZFS Storage Appliance delivers enterprise-class NAS capabilities with leading Oracle integration, simplicity, efficiency, and performance.

The Sun ZFS Storage Appliance 7000 family of products provides robust application and data storage for Oracle's Exadata, SPARC SuperCluster, and Exalogic Elastic Cloud and offers immediate benefits for customers using NAS for enterprise applications, virtualization, cloud, storage consolidation, and data protection.

These products support a rich set of utilities to manage the data stored on these systems while simultaneously providing high-speed data storage and the power and benefits of the ZFS file system.

Sun ZFS Storage Appliance: Architecture

- Sun ZFS Storage Appliance 7000 family of products
  - Sun ZFS Storage 7120 (small enterprises)
  - Sun ZFS Storage 7320 (mid-range storage enterprises; entry-level cluster option)
  - Sun ZFS Storage 7420 (large enterprises; cluster option)
- Analytics
  - A system for dynamically monitoring the storage performance at a glance using graphics
- ZFS hybrid storage pool
  - Low-power, high-capacity disks and flash-based devices managed as a single data hierarchy

ORACLE

The Sun ZFS Storage Appliance is available in three configurations, including dual-cluster configurations that offer maximum availability. Each configuration comes bundled with the same software, including data protocols, compression, and DTrace Analytics software, for system troubleshooting and performance optimization.

The Sun ZFS Storage Appliance has three current platforms: 7120, 7320, and 7420. The Sun ZFS Storage 7120 is ideal for small enterprises. The Sun ZFS Storage 7320 is designed for enterprises with mid-range storage needs and provides a high-availability, entry-level cluster option. The Sun ZFS Storage 7420 is ideal for data-intensive business applications and for virtualized environments that require multiple data services and heterogeneous file sharing. It is available in single or cluster configurations.

Note: Platforms such as 7110, 7210, 7310, and 7410 are considered legacy.

The systems provide a comprehensive and intuitive analytics environment to help you isolate and resolve issues. You can generate real-time graphs of storage usage and system performance. You can also save these graphs for future review. An advanced version of DTrace is also provided on the appliance for server analytics.

Another key feature of the appliance is its intelligent hybrid storage pool that can be composed of flash-memory devices to accelerate reads and writes. The hybrid storage pool can be simultaneously configured on high-capacity, solid-state disks and DRAM (dynamic random access memory). In the hybrid storage pool, all these devices are managed as a single hierarchy.

Sun ZFS Storage Appliance: Architecture

- Data services
  - RAID-Z, mirrored and striped configuration
  - Data deduplication
  - Built-in data compression
- Availability
  - Predictive self-healing
  - Diagnosis of system hardware failure
  - Checksums of all application data and metadata
  - Link aggregations and IP multipathing
  - Lights-out management for remote power control and console access

The Sun ZFS Storage Appliance leverages ZFS capabilities and power. You can use ZFS utilities, such as data compression and deduplication, to save on storage costs.

Predictive self-healing is provided in the redundant configuration, such as in mirrored data and RAID-Z configurations. Self-healing is accomplished though the checksums maintained at the data-block level. Automatic diagnosis of system hardware is performed as part of the fault-management software. When a hardware item such as a CPU is diagnosed to be out of commission, it is taken offline and its workload is spread over the remaining CPUs.

Lights-out management is provided so that system administrators can access and manage systems remotely.

Sun ZFS Storage Appliance: Capabilities

Protocols	Devices	Software
SMB	Solid State Disks (SSD)	Oracle Solaris ZFS File System
NFS	Flash-Based	Storage Management Software (BUI, CLI)
HTTP/HTTPS	NAS	
WebDAV	Local Disk	
iSCSI		
F C		
SRP		
iSER		
FTP/SFTP		

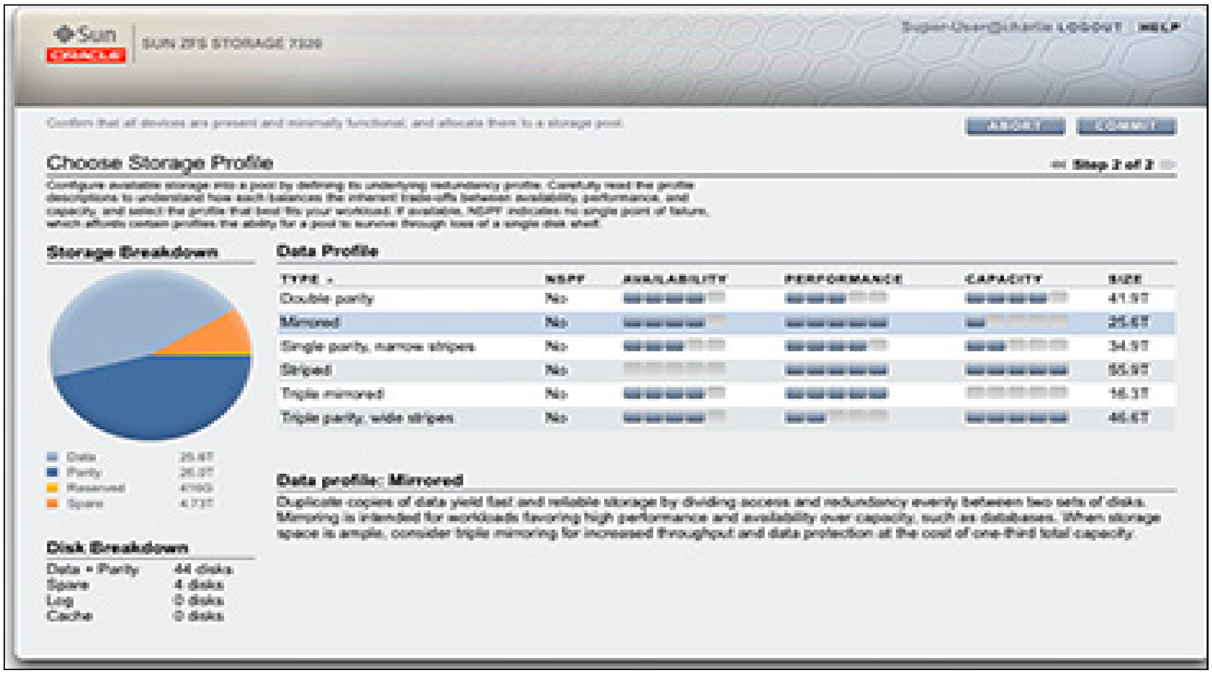
The table shown in the slide displays various protocols, devices, and software that are used by the ZFS Storage Appliance.

The appliance architecture supports multiple file systems by using specific protocols, such as SMB (Server Message Block) and NFS. These protocols (and their associated file systems) can be specified by using the `share` properties. WebDAV (Web-based Distributed Authoring and Versioning) is an extension of the HTTP protocol. FC refers to Fiber Channel. SRP (SCSI RDMA Protocol) is used with the RDMA technology. iSER refers to iSCSI Extensions for RDMA. WebDAV is a file-level protocol; FC, SRP, and iSER are block-level protocols.

You can store application data on different types of high-speed devices, such as solid-state disks (SSDs) and flash memory. The storage can be expanded as required. These devices are an integral part of the appliance and offer fast I/O (input/output)—as well as large amounts of storage—to business applications.

The appliance uses the Oracle Solaris ZFS file system and its utilities for efficient storage management. Additionally, the appliance offers a rich set of storage management software that consists of the browser user interface (BUI) and the command-line interface (CLI). You can use either interface to configure and monitor data storage.

Sun ZFS Storage Appliance: Browser User Interface



The Sun ZFS Storage Appliance software includes a very powerful and easy-to-use BUI that makes system administration significantly easier (an example of which is presented in the slide). You can create storage profiles, configure storage pools, and allocate disks and the pool architecture. You can also create the initial configuration for the appliance, which involves configuring items such as network, system date, and locale.

Sun ZFS Storage Appliance: Command-Line Interface

```
% ssh root@dory
Password:
Last login: Mon Oct 13 15:43:05 2009 from
   kiowa.sf.fishpo
dory:>
```

```
dory:configuration> show
Children:
net => Configure networking
services => Configure services
version => Display system version
users => Configure administrative users
roles => Configure administrative roles
preferences => Configure user preferences
alerts => Configure alerts
storage => Configure Storage
```


The Sun ZFS Storage Appliance software also provides a CLI for system configuration. To use the CLI to configure system settings such as networking, you log in using the `ssh` facility, as shown in the example in the slide.

After you have successfully logged in, you can invoke different levels of facilities by using the subcommand `configure`. For example, you can use the subcommand `configure user preferences`. To see the complete hierarchy of available commands, enter `configuration` and then `show`.

Sun ZFS Storage Appliance: Benefits

- High input/output speed
- Large amounts of potential storage
- Custom data management software (BUI/CLI)
- Hybrid ZFS pools
- Interoperability with other file systems (CIFS)
- ZFS file system utilities

Sun ZFS Storage Appliance: Usage

- Video on demand
  - Requires highly available and scalable storage
- Mission-critical applications
  - Data protection, self-healing, and hardware diagnosis
- Bulk data storage
  - Large amount of storage
- Storage for virtualized environment and cloud
  - Storage expansion on demand

Sun ZFS Storage Appliance and ZFS

- The appliance uses a customized version of Oracle Solaris.
- The appliance's data management software uses ZFS and other utilities.
- The appliance is ideal for businesses that deal with large amounts of data.
- ZFS, which is a file and storage management system, is better suited for built-in disk and large storage-like clusters.
- The Sun ZFS Storage Appliance Simulator can be used in the Oracle VirtualBox environment.

The Sun ZFS Storage Appliance uses a customized version of Oracle Solaris. The appliance's data management software uses ZFS as well as other utilities. If you know the ZFS commands and features, you will be able to manage your data on the Sun ZFS Storage Appliance.

The ZFS Storage Appliance is ideal for businesses that deal with large amounts of data and have to manage storage farms and clusters. ZFS, on the other hand, is a file and storage management system. It can be used with built-in disks and large storage-like clusters.

A recommended way of exploring the functionality and operations of the ZFS Storage Appliance is by using the Sun ZFS Storage Appliance Simulator, which is available for Oracle VM VirtualBox. You can download the simulator free of charge and install it in a current version of VirtualBox. A link to the Sun ZFS Storage Appliance Simulator website is provided in the slide titled "Additional Resources for the Sun ZFS Storage Appliance."

Quiz

You can configure Sun ZFS Storage Appliance interoperability with the Windows OS by using a share property called __________.

a. `SIM`

b. `SMB`

c. `BSM`

Answer: b

Additional Resources for the Sun ZFS Storage Appliance

- Sun ZFS Storage Appliance overview:
  http://www.oracle.com/us/products/servers-storage/storage/nas/overview/index.html
- Sun ZFS Storage Appliance resources:
  http://www.oracle.com/us/products/servers-storage/storage/nas/resources/index.html
- Sun ZFS Storage 7000 Appliance administration training:
  http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=609&p_org_id=1001&lang=US&get_params=dc:D69263GC10,p_preview:N
- Sun ZFS Storage Appliance documentation:
  http://www.oracle.com/technetwork/documentation/oracle-unified-ss-193371.html?ssSourceSiteId=ocomen
- Sun ZFS Storage Appliance Simulator:
  http://www.oracle.com/webapps/dialogue/ns/dlgwelcome.jsp?p_ext=Y&p_dlg_id=10521841&src=7299332&Act=45

ORACLE

Lesson Agenda

- Sun ZFS Storage Appliance
- Pillar Axiom Technology
- Cloud Technology
- COMSTAR Technology

Pillar Axiom Technologies: Overview

- **Multi-protocol storage**: Provides a unified SAN (storage area network) and NAS (network attached storage) storage solution
- **Modular architecture**: Enables independent capacity and performance scaling
- **Advanced quality-of-service (QoS)**: Provides dynamic CPU, cache, and capacity management
- **Distributed RAID**: Supports linear scaling, performance under fault, and fast drive rebuild

Oracle’s next-generation SAN solution is the Pillar Axiom storage system. Although promoted as a SAN solution, Pillar Axiom is in fact a unified SAN and NAS storage offering. This storage solution delivers fast, dependable performance for databases and applications in mission-critical environments with unique quality-of-service (QoS) capabilities.

The Pillar Axiom Storage System’s modular architecture eliminates the need to change storage platforms as your needs increase, enabling you to start small and then independently scale both capacity and performance. With this offering, you can consolidate SAN-based storage by leveraging patented QoS capabilities that prioritize data access across the system and ensure that all applications get the service levels they require when they require them.

The Pillar Axiom’s storage pool architecture with distributed RAID enables the best performance and capacity scalability, the best performance under fault, and the fastest drive rebuild times in the industry.

Pillar Axiom Architecture: Overview

- The Pillar Axiom family of products
  – Pillar Axiom 300 storage system
  – Pillar Axiom 500 storage system
  – Pillar Axiom 600 storage system
- Quality of service (QoS)
  – Data prioritization based on importance of application
- Storage domains
  – Subset of a virtual storage pool containing grouping of physical bricks. A *brick* is defined as a storage enclosure.
- Storage volume expansion
  – File systems and LUNs can be expanded to meet growing business needs.

ORACLE

The Pillar Axiom product family consists of three storage systems: Pillar Axiom 300, 500, and 600. The Pillar Axiom 600 storage system is the most current version of the platform.

You can use QoS to prioritize data based on the nature of your application. You can also set performance levels and RAID characteristics by using QoS.

The storage domains are created to perform maintenance, security, and other operations. Multiple bricks can be chained together.

Pillar Axiom Hardware: Overview

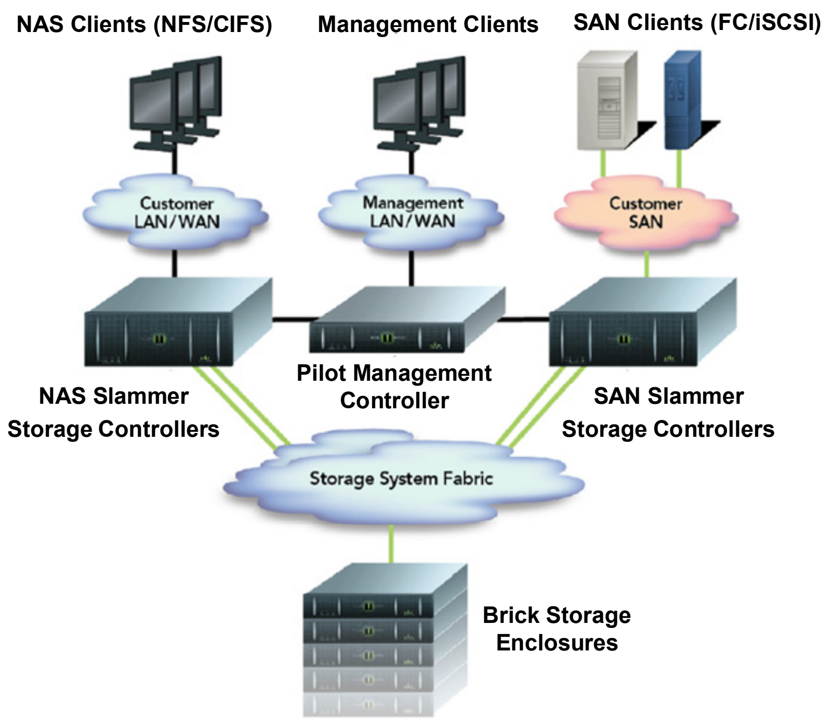
Modular system

- Pilot management controller
  - Out-of-band management system that directs and manages all system activity
- Slammer storage controllers
  - Interface to host storage network
  - Processing every I/O request
- Brick storage enclosures
  - Contain SSD, SATA, or FC bricks (disk drives)
- Storage system fabric (SSF)
  - Carries all data traffic among Slammer storage controllers and Brick storage enclosures

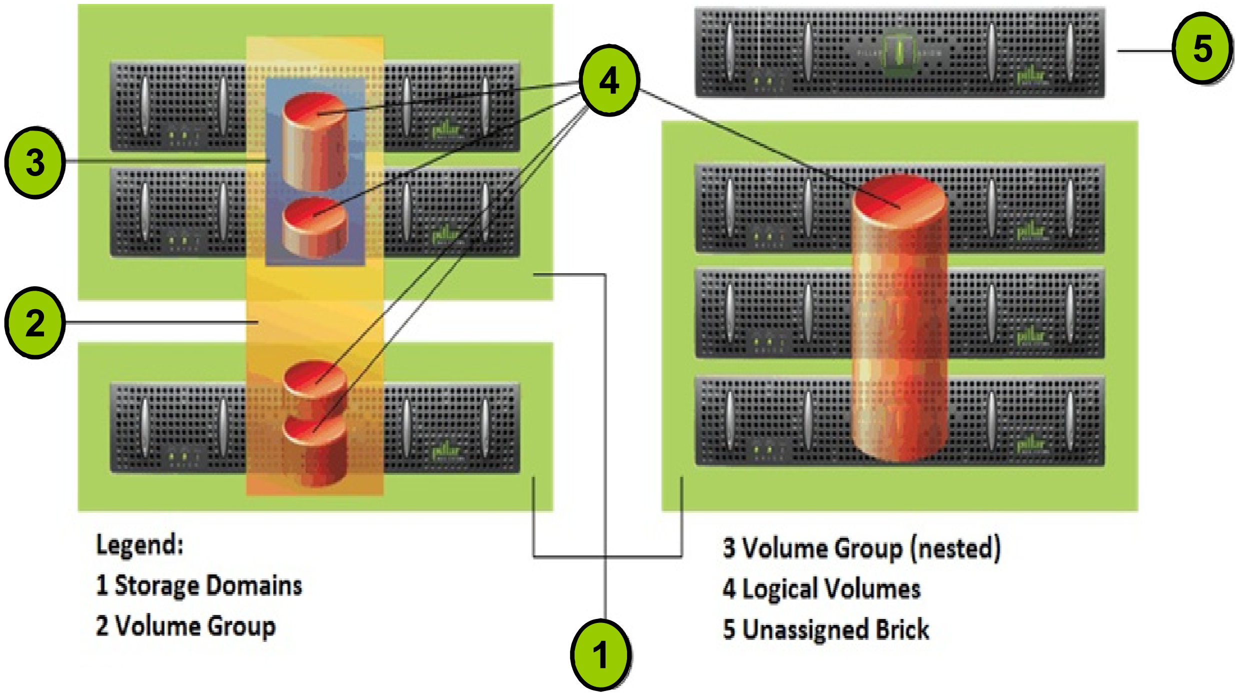
Pillar Axiom Software: Overview

- Pilot Axiom storage services manager
  - GUI interface to configure and monitor the Pillar Axiom storage system
- Policy-based provisioning and storage management
  - Administrator-defined policies to provision storage requirements
- Quality of service (QoS) attributes
  - Administrator-defined policies that determine how to store the data in the storage pool (queuing, caching priorities, RAID type)
- Storage domains
  - Administrator-managed logical volumes in a specific collection of bricks

Pillar Axiom Storage System: Infrastructure



Pillar Axiom Storage Domains and Volumes



The Pillar Axiom storage domains and volumes are identified in the slide. The total storage is divided at the physical hardware and logical level for organized data access and manageability.

- Item 1 identifies the physical storage domain which may contain multiple disk clusters.
- Item 2 demonstrates how a data volume group is configured to use the storage domains.
- Item 3 shows that the major volume group is divided into smaller groups.
- Item 4 illustrates different type of logical volumes.
- Item 5 identifies an assigned brick that is standby and can be configured in a storage domain as needed.

Pillar Axiom Technology: Benefits

- Convenient management of storage resources as needed by the business
- Consistent QoS tools to manage I/O contention
- Data migration support
- Distributed RAID technology

Pillar Axiom Technology: Usage

The Pillar Axiom technology is used when business applications:

- Store mission-critical data on a storage area network (SAN) or on network attached storage (NAS)
- Need to manage storage resources priorities
- Want to store redundant data
- Want to migrate data to other storage media

Pillar Axiom Technology and ZFS

Pillar Axiom storage technology

- At a lower (storage/hardware) level
- Unaware of a specific file system
- Compatible with Solaris 9 and Oracle Solaris 10
- Compatible with Solaris Sun Clusters

ZFS technology

- Used in Solaris 10 and 11
- Can be used in Pillar Axiom storage at a higher layer
- No direct relationship between ZFS and Pillar Axiom storage

This slide describes the differences between ZFS and the Pillar Axiom storage technology. Note that the Pillar Axiom storage technology is not currently compatible with Oracle Solaris 11.

Obviously, Pillar Axiom and ZFS are two very different technologies, and there is no direct relationship between the two. Pillar Axiom is a hardware technology and ZFS is a software technology.

For customers who want flexible storage hardware that can use any file system and is compatible with both Solaris 9 and 10 as well as Solaris Sun Clusters, Pillar Axiom is an excellent option.

Quiz

Which network storage technology is used by Pillar Axiom storage systems?

a. SAN
b. NAS
c. Both

Answer: c

Additional Resources for Pillar Axiom

- Pillar Axiom 600 storage system overview:
  http://www.oracle.com/us/products/servers-storage/storage/san/pillar/pillar-axiom-600/overview/index.html
- Pillar Axiom 600 storage system resources:
  http://www.oracle.com/us/products/servers-storage/storage/san/pillar/pillar-axiom-600/resources/index.html
- Pillar Axiom 600 SAN storage system administration training:
  http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=609&p_org_id=1001&lang=US&get_params=dc:D74247GC10,p_preview:N
- Pillar Axiom 600 storage system documentation:
  http://docs.oracle.com/cd/E26030_01/index.html

Lesson Agenda

- Sun ZFS Storage Appliance
- Pillar Axiom Technology
- Cloud Technology
- COMSTAR Technology

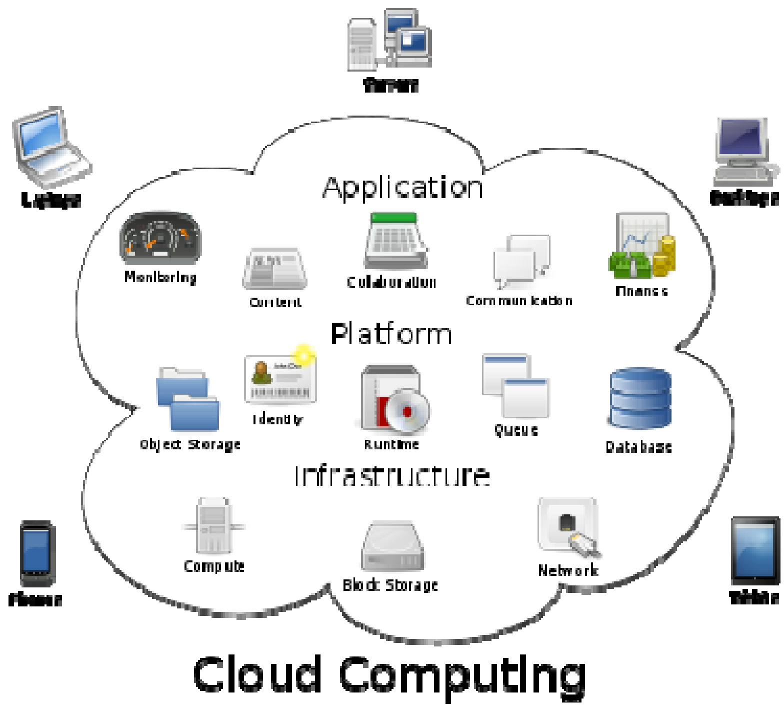
Cloud Computing

- Cloud computing is the Internet-based delivery of computing services.
- Shared resources, software, and information are provided to computers and other devices.
- Internet infrastructure is generally used as the network.
- Cloud computing offers computation, software applications, data access, data management, and storage resources.
- Details and location of the cloud infrastructure are transparent to users.
- End users access the cloud primarily by using a web browser.

Cloud computing is Internet-based computing in which shared resources, software, and information are provided to computers and other devices on demand (similar to the way that the electricity grid is delivered to homes and businesses).

Cloud computing is a natural evolution of the widespread adoption of virtualization, service-oriented architecture, and utility computing. Cloud computing describes a new delivery model for IT services based on the Internet, and it typically involves the provisioning of dynamically scalable and often virtualized resources. It is also considered to be the by-product and consequence of the ease-of-access to remote computing sites provided by the Internet.

Cloud Architecture



The cloud computing infrastructure offers consumers flexible services. Users access the cloud by using a desktop, laptop, or mobile device.

Based on the service model, the cloud can offer business application, operating system, and hardware management. You can combine cloud computing with your traditional IT Infrastructure to create the most suitable IT model.

Cloud Service Models

The cloud consists of three service models:

- Infrastructure as a service (IaaS)
  - Cloud providers offer basic hardware such as computers, storage, firewalls, load balancers, and networks on demand.
  - Cloud users install and manage operating systems (such as Solaris 11) and their business applications.
- Platform as a service (PaaS)
  - Cloud providers offer a computing platform that can include operating systems, programming environments, databases, and web servers.
  - Cloud users develop and execute their applications on the platform.

Cloud offerings are classified into three different service models. The models are created to suit a variety of organizations.

Infrastructure as a service (IaaS) provides the hardware needed to develop a product. Because hardware maintenance is not the consumer's responsibility, the consumer can focus on software development and operation.

Platform as a service (PaaS) goes one step further than IaaS to provide a comprehensive "computing platform." In addition to hardware, the computing facilities (such as programming, databases, and web servers) are included. Application developers can develop and run their applications without the cost and complexity of purchasing the underlying hardware and software layers.

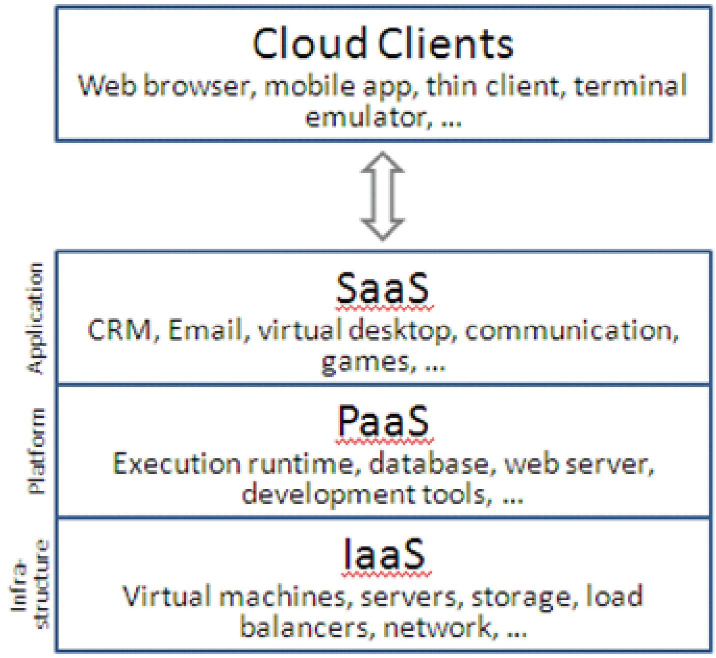
Cloud Service Models

- Software as a service (SaaS)
  - Cloud providers offer installation and operation of the application software as well as management of the cloud infrastructure.
  - Cloud users view their applications from the cloud clients such as desktop computers, laptops, and smartphones.

Software as a service (SaaS) offers a complete package to the organizations who would like to contract out the full suite of application installation, operation and maintenance. The consumer is able to view the business application but depends on the cloud provider for maintenance and management of the application.

This model can be used by organizations that are in the business of only developing products. After a product is developed, an organization can hand if off to a cloud provider to manage.

Cloud Service Models: Summary



Cloud Deployment Models

Common cloud deployment models include:

- Public cloud
  - Cloud providers offer software applications, storage, and other resources. Cloud users connect to the public cloud through the Internet.
- Private cloud
  - An internal or corporate cloud that provides hosted services to a limited number of people behind a firewall
- Hybrid cloud
  - A combined public and private cloud

The different cloud deployment models offer a variety of choices to the consumer.

- **Public cloud:** A public cloud provider can be in the business of selling storage to the public. In this case, an individual consumer can save critical data to the cloud storage with a drag of the mouse. The data may be songs or tax information (whatever a consumer needs to save).
- **Private cloud:** Organizations that have high security as their prime objective can create their own internal private cloud. This cloud model may be suitable for organizations working with highly classified information.
- **Hybrid cloud:** Some organizations, such as auto manufacturers, may want to use the hybrid cloud model. Using this model, they can configure their intranet (behind the firewall) for the company and its employees. For their vendors, an extranet can be created in the public cloud. Security is assumed in both cases.

Cloud Computing: Benefits

- Flexible service models are available to suit a variety of business needs and requirements.
- A corporation can save significant time to market a product.
- Platform and other resources are available on demand.
- Cloud users do not care where the cloud infrastructure is.
- Cloud users pay for resource usage. They do not need to plan, order, and maintain the equipment.

Who Uses the Cloud?

- Corporations that have a need to save and manage data
- Businesses that want to save on short and long term storage maintenance costs
- Corporations that are planning to develop a product quickly
- Government departments that need to store large amounts of data
- Individuals who want to save data, audio, video, and music away from their computer

ORACLE

Cloud Computing and ZFS

- The Oracle Solaris 11 ZFS file system manages the application data stored on the cloud hardware.
- The business application can be managed by the cloud provider or the cloud user.
- ZFS compression and deduplication can be used to save storage resources in the cloud.
- ZFS utilities can be used to manage application data efficiently.

Quiz

If you want a cloud technology provider to manage your business application and the complete cloud infrastructure for you, which service model would you use?

a. IaaS
b. SaaS
c. PaaS

Answer: b

Lesson Agenda

- Sun ZFS Storage Appliance
- Pillar Axiom Technology
- Cloud Technology
- COMSTAR Technology

COMSTAR and iSCSI

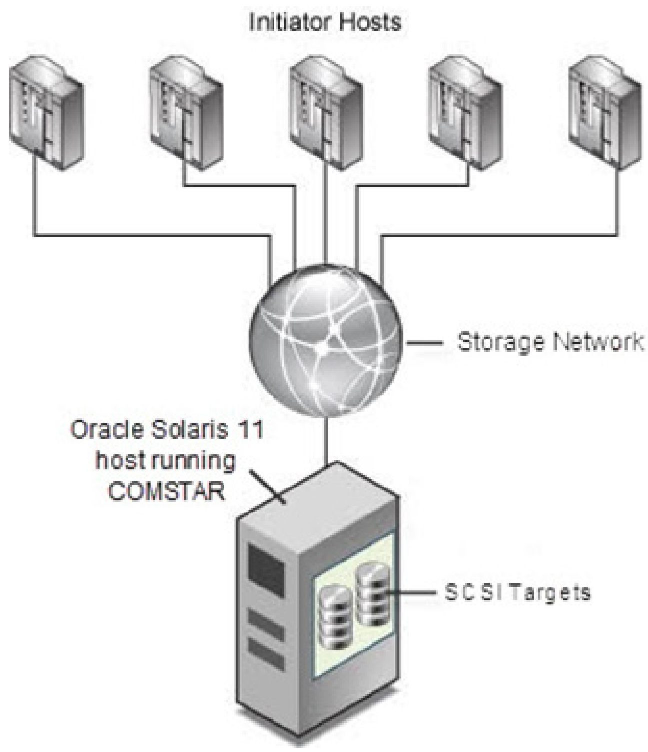
- Software-based framework that links data storage systems
- Supports iSCSI protocol and target devices
- COMSTAR manages the iSCSI target devices,
- COMSTAR uses a SCSI Target Mode Framework (STMF) to manage target devices.

COMSTAR (Common Multiprotocol SCSI Target) is a software framework that provides support for the iSCSI protocol. iSCSI (Internet small computer system interface) is an IP-based storage networking standard for linking data storage subsystems. By carrying SCSI commands over IP networks, the iSCSI protocol enables you to mount disk devices from across the network onto your local system. On your local system, you can use the devices like block devices.

COMSTAR enables you to convert any Oracle Solaris 11 host into a SCSI target device that can be accessed over a storage network by initiator hosts by using SCSI Target Mode Framework (STMF) to manage target storage devices. STMF provides the following components:

- **Port providers (or plug-ins):** Implement protocols, such as Fibre Channel (FC) and iSCSI
- **Logical unit providers:** Emulate various SCSI devices, such as disk and tape devices
- **Management library (`libstmf`):** Provides the COMSTAR management interface

COMSTAR



Multiple hosts can connect to the SCSI targets on the local host running COMSTAR with Oracle Solaris 11 as the operating system.

COMSTAR: Requirements

- Solaris storage software and devices
- `storage-server` software package on the system providing storage devices
- Functional NIC (network interface card)

To use COMSTAR, you must meet the requirements listed in the slide.

The Solaris storage software is available as part of the operating system. The devices are either locally attached or network attached.

The `storage-server` package is not installed as part of the Oracle Solaris 11 operating system installation. You install this package from the local IPS repository.

Configuring COMSTAR

1. Install the `storage-server` software package.
2. Create an iSCSI LUN.
   a. Enable the `stmf` service.
   b. Identify a disk volume to serve as the SCSI target.
   c. Run the `stmfadm` utility to create a LUN.
   d. Make the LUN viewable to the initiators.
3. Create the iSCSI target.
   a. Enable the target service.
   b. Run the `itadm` utility to create an iSCSI target.
4. Configure an iSCSI initiator.
   a. Enable the initiator service.
   b. Configure the target device discovery method.
   c. Reconfigure the `/dev` namespace to recognize the iSCSI disk.
5. Access the iSCSI disk.
   a. Use the `format` utility to identify the iSCSI LUN information.
   b. Create a ZFS file system on the iSCSI LUN.

1. **Install the storage-server software package:** This package contains all the software required to configure SCSI targets. The storage-server software package is installed on the system that provides the storage devices.
2. **Create an iSCSI LUN:** This task is performed on the system that provides the disk volumes. The disk volume provided by the server is referred to as the *target*. When the LUN is associated with an iSCSI target, it can be accessed by an iSCSI initiator.
3. **Create the iSCSI target:** This task is performed on the system that provides the disk volumes.
4. **Configure an iSCSI initiator:** This task is performed on the initiator client host.
5. **Access the iSCSI disk:** This task is performed on the initiator client host.

COMSTAR: Benefits

- The iSCSI protocol runs across existing Ethernet networks:
  - Any supported network interface card (NIC), Ethernet hub, or Ethernet switch can be used.
  - One IP port can handle multiple iSCSI target devices.
  - Existing infrastructure and management tools for IP networks can be used.
- Existing Fibre Channel devices can be connected to clients without the cost of Fibre Channel host bus adaptors (HBAs).
- Systems with dedicated arrays can export replicated storage with ZFS or UFS file systems.
- There is no upper limit to the maximum number of configured iSCSI target devices
- You can connect to Fibre Channel or SAN environments with the appropriate hardware.

COMSTAR: Usage

- COMSTAR storage is part of the cloud technology.
- The location of the storage is transparent to the consumer.
- Organizations can store data on rented storage.

COMSTAR and ZFS

- A ZFS file system can be created on an iSCSI LUN.
- The iSCSI LUN configuration is part of the COMSTAR configuration.
- The iSCSI LUN resides on the iSCSI target.

This slide explains how ZFS and COMSTAR can be used together.

Note: You use ZFS to create the storage pool as well as the ZFS volume (ZVOL) that will be used as an iSCSI LUN on the target system. You first create the ZFS volume to be used as an iSCSI LUN, and then you create a LUN for the ZFS volume.

Quiz

What does COMSTAR stand for?

a. Community Star
b. Common Multiprotocol SCSI Target
c. Common Multiprotocol Platform

Answer: b

Quiz

COMSTAR manages iSCSI target devices.

a. True
b. False

Answer: a

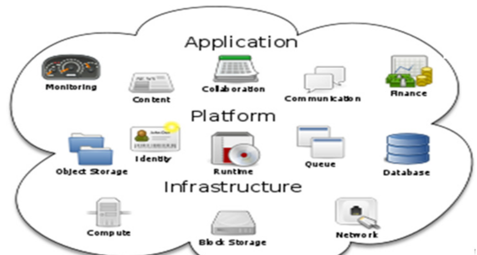
Additional Resources for COMSTAR

- COMSTAR overview: http://java.sun.com/developer/technicalArticles/openstorage/COMSTAR/index.html
- COMSTAR training in *Transition to Oracle Solaris 11*:
- http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getCourseDesc?dc=D73488GC10&p_org_id=18&lang=US
- Configuring a COMSTAR iSCSI target practice: http://apex.oracle.com/pls/apex/f?p=44785:24:0:::24:P24_CONTENT_ID,P24_PREV_PAGE:5914,1
- COMSTAR configuration documentation: http://docs.oracle.com/cd/E23824_01/html/821-1459/fmvcd.html#scrolltoc

ORACLE

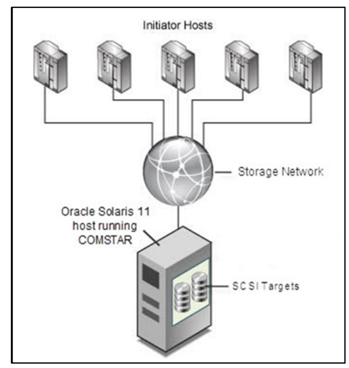
Current Oracle Storage-Related Technologies

Sun ZFS Storage Appliance



Cloud Computing

Pillar Axiom



COMSTAR

You should now have a better understanding of the most current Oracle data management and storage technology offerings and the role that ZFS plays in the data management and storage spaces. Although we focus in this course on managing data with ZFS, you now know that this is only one of several options that Oracle offers to address your business application needs.

Summary

In this lesson, you should have learned how to describe:

- Sun ZFS Storage Appliance
- Pillar Axiom technology
- Cloud technology
- COMSTAR technology

In this lesson, you were introduced to the Sun ZFS Storage Appliance, Pillar Axiom, cloud, and COMSTAR technologies. In addition, you learned about the architecture, benefits, and use of each technology and about each technology's relationship to ZFS.

3

Planning for Data Management

Objectives

After completing this lesson you should be able to identify the requirements for:

- The business application's data storage
- ZFS root pool management
- Configuration and management of the Oracle Solaris ZFS components
- ZFS properties configuration
- Using ZFS access control lists to protect business data
- The ZFS delegated administration model
- Business data backup and recovery

Lesson Agenda

- Data Storage Requirements for the Business Application
- Root Pool Management Requirements
- Oracle Solaris ZFS Components Configuration and Management Requirements
- ZFS Properties Configuration Requirements
- Requirements to Protect Business Data by Using ZFS Access Control Lists
- ZFS Delegated Administration Model Requirements
- Business Data Backup and Recovery Requirements

In this lesson, you use a data management plan that outlines all of the requirements you must meet to satisfy the needs of the business.

Storage Requirements for the Business Application: Overview

- Meet with the business application analyst.
- Understand the business application architecture.
- Discuss the data redundancy requirements.
- Determine the data storage requirements for the application and subsystems.
- Discuss the application's memory usage.

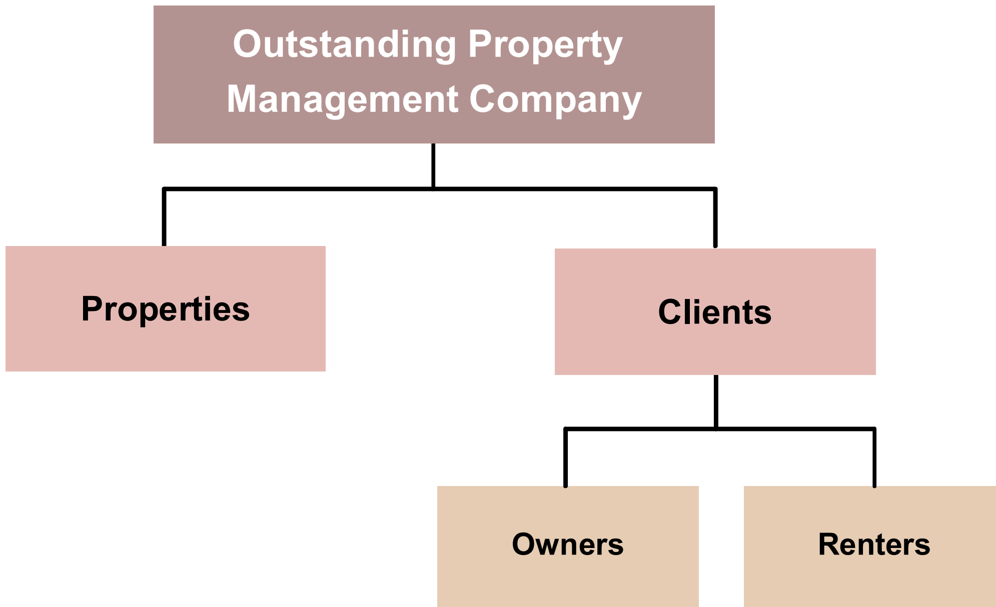
To gather data storage requirements, you meet with the business application analyst, who has captured the requirements in a data management plan. The business analyst works with the business application and is therefore familiar with the application itself, its architecture, and the requirements. Understanding the application architecture is crucial for a data administrator. Knowing the architecture can help you understand the process flow, required processes, reports generated, and the application's subsystems.

For smooth application operation, determining the data redundancy requirements is important because you, as a data administrator, can determine the type of ZFS storage pools you must configure. In case of data loss, if you can't afford the wait time to recover the data, you may consider using a mirrored storage or RAID-Z storage type configuration for the ZFS pools.

How much data is going to be stored by the application and the subsystems? This information can help you create the application data pools with the appropriate amount of storage. Based on the nature of the data in the subsystems, the business analyst is also able to explain if every subsystem requires a separate pool.

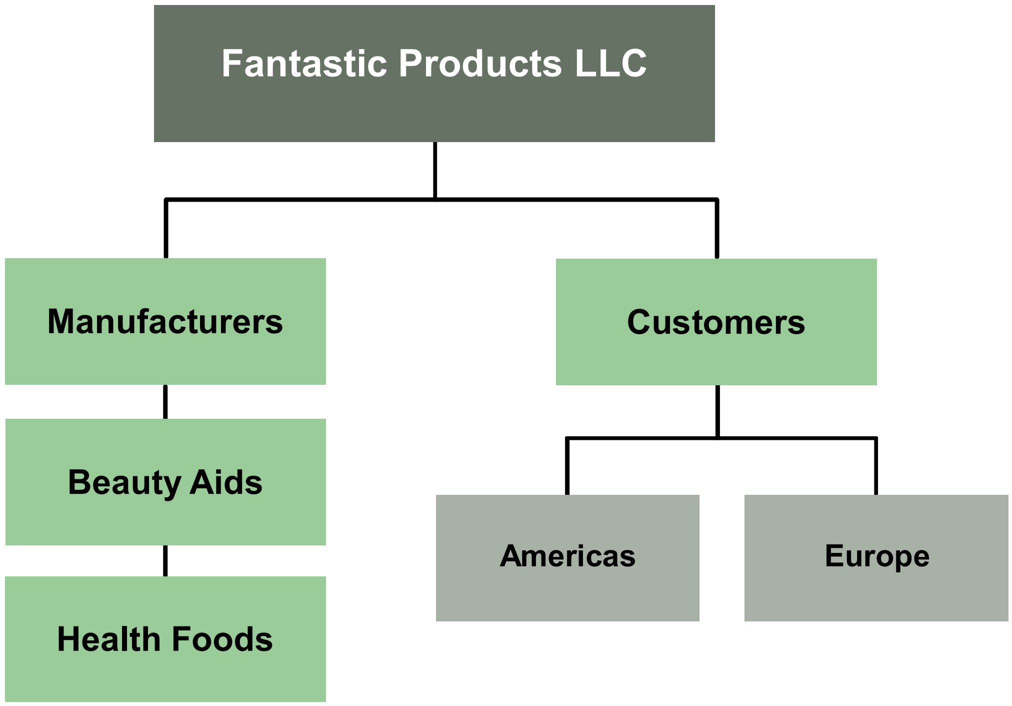
By determining the memory usage of the application, you can determine the swap space requirements. Normally the swap space is allocated in the root pool; however, you will need to verify that the amount of space allocated to the root pool is sufficient to meet the needs of the business application.

Business Applications



In the practice for Lesson 1, you were presented with an overview of the project assignment that you complete as part of this course. As you recall, you are working for the Data Mapping and Storage Solutions (DMSS) company, which is in the business of managing data and storage for businesses on a contract basis. For training purposes, we assume DMSS has the management contract with two fictitious businesses. The first business is called Outstanding Property Management Company. This company is in the business of managing properties on behalf of property owners, as illustrated in the graphic in the slide. As part of your project assignment responsibilities, you are tasked with supporting the application for this company. For this application, we consider both renters and owners to be clients.

Business Applications



The second fictitious company that you support as part of your project assignment is Fantastic Products LLC. This company stores and manages manufacturer, product (beauty aids and health foods), and customer data, as illustrated in the graphic in the slide. The customers are located in North and South America and in Europe.

Storage Requirements for the Outstanding Property Management Company

Storage Entity Name	Data Storage Type	Data Type	Number of Required Disks (1 Disk = 1 GB)
`opmpool`	ZFS Mirrored Storage Pool	Property Management Application Data	4
`props`	File System	Properties Data	Dynamic; as needed
`clients`	File System	OPM Clients Data	Dynamic; as needed
`renters`	File System	OPM Renters Data	Dynamic; as needed
`owners`	File System	OPM Property Owners Data	Dynamic; as needed
File System Backup	Snapshots	All OPM Data	Not required initially

You are ready to begin gathering storage requirements for each application. You begin with the requirements for the Outstanding Property Management Company (OPM). According to the plan that the business application analyst has created, you must create a ZFS pool called `opmpool` for the OPM application. This pool is a mirrored storage pool.

During the analysis phase of the project, the mirrored and RAID-Z configurations were considered and the mirrored configuration was selected over RAID-Z for performance reasons. The OPM application analyst needed an online copy of the data available all the time to be able to recover the data as quickly as possible. The mirrored configuration provides this facility.

The storage in the training environment is scaled down. The OPM storage needs are determined to be 2 GB. Because each disk has 1 GB of storage, this pool requires four disks. This pool contains four file systems:

- **`props`:** A file system to store property information
- **`clients`:** A file system to store the company's clients
- **`renters`:** A sub file system in the `clients` file system; stores renter information
- **`owners`:** A sub file system in the `clients` file system; stores information about property owners

In addition, there is a file system backup that contains backup snapshots for each file system.

All the file systems require storage that can grow dynamically based on future growth.

Pool storage usage should be monitored. The best practice is to add more storage after storage usage reaches 80%.

Storage Requirements for Fantastic Products LLC

Storage Entity Name	Data Storage Type	Data Type	Number of Required Disks (1 Disk = 1 GB)
`fplpool`	ZFS RAID-Z Storage Pool	Fantastic Products Application Data	3
`custfs`	File System	Customers Data	Dynamic; as needed
`mfrfs`	File System	Product Manufacturers Data	Dynamic; as needed
`bafs`	File System	Beauty Aids Product Data	Dynamic; as needed
`hpfs`	File System	Health Products Data	Dynamic; as needed

For Fantastic Products LLC (FPL), the plan is to create a ZFS pool called `fplpool`. There is a business requirement for the FPL application to maximize disk space and utilization. Mirrored and RAID-Z storage pool configurations were considered, but the RAID-Z configuration was selected over the mirror configuration because the FPL needed the capability to do random searches. The RAID-Z technology provides faster random reads than the mirrored technology. The secondary factor was that it provides data redundancy.

The storage in the training environment is scaled down. The FPL storage requirements are determined to be 2 GB. Because each disk has 1 GB of storage, the FPL application's RAID-Z pool requires three disks (one disk being used by the parity information).

The plan is for the pool to contain four file systems:

- **`custfs`**: Stores customer information
- **`mfrfs`**: Stores the company's manufacturers
- **`bafs`**: Stores information about beauty aids
- **`hpfs`**: Stores information about health products

All the file systems require storage that can grow dynamically to accommodate future growth.

For this application, the business analyst anticipates that the volume of transactions will increase in the near future. Therefore, the plan is to increase the memory in the root pool to allow this growth.

The pool storage usage should be monitored. The best practice is to add more storage after storage usage reaches 80%.

Quiz

Why did the Fantastic Products LLC application choose the RAID-Z storage pool architecture instead of a mirrored storage pool? (Choose all that apply.)

a. ZFS mirror pool technology is too complicated.

b. RAID-Z technology is cost-effective.

c. The application is random-search intensive.

Answer: b, c

Lesson Agenda

- Data Storage Requirements for the Business Application
- Root Pool Management Requirements
- Oracle Solaris ZFS Components Configuration and Management Requirements
- ZFS Properties Configuration Requirements
- Requirements to Protect Business Data by Using ZFS Access Control Lists
- ZFS Delegated Administration Model Requirements
- Business Data Backup and Recovery Requirements

Root Pool Management: Requirements

System Level	Outstanding Property Management Company	Fantastic Products LLC
Inspect swap and dump devices.		
Increase swap and dump to 1.5 GB each. Add a 1 GB independently manageable swap volume.		Requirement for current project Requirement for the new project
Inspect the current boot environment.		
Boot from an alternate disk in a mirrored ZFS root pool.		
Boot from a backup ZFS root file system.		
Boot an unbootable system for system recovery.		

Now you consider the root pool management requirements that the business application analyst has identified in the data management plan. The requirements are listed in the slide. As you can see in the table, these requirements are at the system level and can affect either of the OPM and FPL applications. At this time, only the FPL application has a requirement to increase the swap and dump storage allocations. As a ZFS storage administrator, you keep in close contact with the business application analysts to address any system performance issues.

ZFS Root Pool Storage

- Oracle Solaris 11 system software storage
- Image Packaging System storage
- ZFS pool metadata storage

Note: Storage requirements are estimated to be 32 GB.

The ZFS root pool storage requirements depend on the space used by the Oracle Solaris 11 operating system, the Image Packaging System (IPS) repository, and future requirements for alternate boot environments.

Note: Although the IPS repository can be set up on a separate server, in the training environment it is installed on the Sol11-Server virtual machine.

Based on each of the factors in the slide, the business application analyst has determined that you need a 32 GB disk to meet the root pool storage requirements.

Note: The disk size requirements for ZFS root pool storage presented in this course are constrained by the practice environment and are not intended to reflect reality.

Managing Swap and Dump Devices

- Swap space extends the total virtual memory that is available.
- Monitor memory paging.
- Adjust swap space for the application.
- Allocate sufficient storage to the dump device for the application vulnerability.
- System swap space should default to ¼ of the physical memory.
- System dump device storage should default to half the physical memory.

To ensure that the swap and dump space has been sufficiently allocated, we check the application memory requirements. If the transaction processing rate is high, higher swap space may be beneficial. The dump device is used when the system or application has terminated abnormally and creates a system dump. If the application or the system crashes often, it may require more dump space.

The swap and dump storage guidelines in the slide are recommended as a starting point in the absence of any indication from the business applications. They should be reviewed over time to ensure that they are sufficient for the applications using them.

Note: In the training environment, each of the swap and dump storage allocations is approximately half of the available physical memory.

Managing Swap and Dump Devices: Best Practice

System Type	Swap Volume Size	Dump Volume Size
System with about 4 GB of physical memory	1 GB	2 GB
Mid-range server with about 8 GB of physical memory	2 GB	4 GB
High-end server with about 16 GB to 128 GB of physical memory	4 GB	8 GB to 64 GB
High-end server with more than 128 GB of physical memory	¼ of physical memory size	½ of physical memory size

The swap and dump storage allocation should be customized to the requirements of a business application. However, in the absence of any requirements or historical data, the slide presents a chart with recommended storage allocations based on the type of system.

Swap and Dump Storage: Requirements

- The FPL application requires permanently increasing the swap space to 1.5 GB.
- The FPL application's new project requires an attachable 1 GB swap volume.
- The FPL application's new project requires increasing the dump space to 1.5 GB.

Your business application analyst has determined that the default swap and dump storage for the OPM application is sufficient.

However, for the FPL application, it was estimated that the transaction rate would increase quickly; therefore, there is a need for more swap memory. The plan is to increase the swap from 1 GB to 1.5 GB.

Although we plan to increase the swap space for the FPL application, that swap space is shared with other applications. For a new project, the FPL application requires a separate, manageable swap volume that can be added to the total swap space as needed when the transaction volume is high. Therefore, the plan requires that you create a 1 GB swap volume that can be added to the system memory as required by the FPL application.

Because this application is brand new and is considered slightly unstable, there is a recommendation to increase the dump device storage. In accordance with the plan, when the time comes, you will increase the dump storage from 1 GB to 1.5 GB.

Using an Alternate Disk for Booting

- Oracle Solaris 11 installation creates one boot disk.
- UFS boots from a single root file system.
- ZFS boots from a particular ZFS pool.
- You must specify a boot device and a root file system.
- ZFS root pool can have multiple bootable root file systems.
- Disk corruption creates issues.
- Best practice: Make a mirrored root pool by adding an alternate disk.

During operating system installation, Oracle Solaris 11 creates one root pool on one main disk, which becomes the only boot disk. In Oracle Solaris 10 and earlier releases using the UFS file system, only a single root file system is provided. However, in Oracle Solaris 11 using ZFS, you can have multiple root file systems, which are bootable.

While using ZFS as the root file system, the system boots from a specific ZFS pool; however, you can also create an alternate root pool. While working with multiple bootable file systems, you can designate a particular bootable file system to be used in the next system boot.

If this disk becomes corrupted, it can affect the entire operating system and any applications running on it.

For this reason, the requirement is for you to create and test the alternate boot disk. If the main boot disk becomes corrupted, it will affect both the business applications and any non-global zones because all these components belong to the global zone.

You can add a disk in the root pool to create a mirrored root pool. Both the disks in the pool contain the same data, which means that you now have an online backup of the root pool.

Configuring a mirrored root pool is a best practice because it reduces down time caused by to hardware failures.

Alternate ZFS Root File System for Booting

- Oracle Solaris 11 installation creates one boot environment (BE) called `solaris`.
- Best practice: Create another BE as backup.
- Requirement: Create a backup BE called `solaris2`.

ORACLE

During operating system installation, Oracle Solaris 11 creates one boot partition called the *boot environment* (BE). By default, this BE is called `solaris` and contains the ZFS root file system. As a precautionary measure during operating system installation, it is a best practice to create another BE as a backup. In the case of data corruption, this alternate BE can be used to recover the data. When you create another BE, Solaris places a boot record on it so that the system can be booted from this alternate BE.

In the data management plan, the business application analyst has specified a requirement for you create a backup BE called `solaris2`.

Quiz

What is the purpose of creating an alternate boot environment? (Select two answers.)

a. To create a backup of the whole system, including business applications
b. To have an alternate boot environment that is bootable
c. To have another boot environment for installing packages

Answer: b, c

Lesson Agenda

- Data Storage Requirements for the Business Application
- Root Pool Management Requirements
- Oracle Solaris ZFS Components Configuration and Management Requirements
- ZFS Properties Configuration Requirements
- Requirements to Protect Business Data by Using ZFS Access Control Lists
- ZFS Delegated Administration Model Requirements
- Business Data Backup and Recovery Requirements

ZFS Hierarchy: Requirements

Outstanding Property Management Company		Fantastic Products LLC	
Storage pool type and name	Type: Mirrored Name: `opmpool`	**Storage pool type and name**	Type: RAID-Z Name: `fplpool`
Number of disks	4 disks at 1 GB each		3 disks at 1 GB each
File systems	`props`	**File systems**	`custfs`
	`clients`		`mfrfs`
	`renters`		`bafs`
	`owners`		`hpfs`

As part of the data storage requirements, the business application analyst has delivered the ZFS hierarchy requirements, which are summarized in the table in the slide.

Data Sharing in the Non-Global Zone

Create a non-global zone called `fpzone` to meet the following requirements:

- The FPL manufacturers file system (`mfrfs`) must be delegated to the zone.
- The FPL manufacturers file system is automatically shared with other users.
- The Zone Administrator (vs. Global Zone Administrator) is responsible for managing the `mfrfs` file system.
- The customer file system (`custfs`) is available to any user for viewing purposes only.
- No other application exists in the zone.
- The zone shares the global zone network interface.

As part of the ZFS hierarchy requirements, the business analyst has determined that the manufacturer and the customer data in the FPL application should be shared with any system user. However, the manufacturer data should be managed only from the zone. A zone is created for data-sharing purposes.

To meet these requirements, you must configure the `mfrfs` and `custfs` file systems in a non-global zone called `fpzone`. When you create the zone, you specify that it uses the same network interface as the global zone.

ZFS Data Encryption: Requirements

Outstanding Property Management Company		Fantastic Products LLC	
Encryption requirements	File system: `investordata` in the `investorpool`	**Encryption requirements**	File: `/export/home/encrypted`
			Pool: `securepool`

The business analysts for both applications have also identified several data encryption requirements, which are presented in the table in the slide. These requirements cover encryption at the file, file system, and pool levels. The analysts have determined the level of encryption required (file, file system, or pool) based on the specific needs of the business.

For example, the Outstanding Property Management Company wants to protect data about potential investors. To support this requirement, the business analyst asks you to create an encrypted file system called `investordata` in the `investorpool`.

Fantastic Products LLC has a requirement to save the application passwords in a central place (`/export/home/encrypted` file in the root pool). For security reasons, this file must be encrypted. The Fantastic Products LLC application also maintains a confidential file system called `newproducts` in a pool called `securepool`. To ensure that all the product-related data in the pool is encrypted automatically, the business analyst wants you to encrypt at the pool level.

ZFS Data Migration: Requirements

Fantastic Products LLC: Data Migration Requirements	
Storage pool	Migrate `mfrs` storage pool to remote system.
File systems	Migrate Asian manufacturer data (ZFS file system `mfrs/asia`) to `fplpool/asia` (remote).
	Migrate manufacturer UFS data (called `ufsdata`) to `fplpool/ufsdata` (remote).
	Migrate manufacturer UFS data (ZFS file system `fplpool/ufsdata`) to `mfrs/ufsdata` (local).

The Fantastic Products LLC application analyst has also identified several data migration requirements for the application. As summarized in the table in the slide, these requirements include migrating a storage pool between systems and using the shadow migration feature to migrate local and remote ZFS and UFS file systems to a target ZFS file system.

ZFS Component Upgrades: Requirements

Outstanding Property Management Company		Fantastic Products LLC	
Upgrade requirements	Pools: Upgrade as required to most current ZFS release.	**Upgrade requirements**	Pools: Upgrade as required to most current ZFS release.
	File systems: Upgrade as required to most current ZFS release.		File systems: Upgrade as required to most current ZFS release.

To ensure that both the Outstanding Property Management Company and Fantastic Products LLC applications are using the most current ZFS release, the business analysts have specified a requirement in the data management plan to upgrade all pools and file systems. This requirement ensures that both applications are able to take advantage of the latest pool and file system features.

Quiz

Including the global zone, how many zones will you be working with?

a. 1
b. 2
c. 3

Answer: b

Lesson Agenda

- Data Storage Requirements for the Business Application
- Root Pool Management Requirements
- Oracle Solaris ZFS Components Configuration and Management Requirements
- ZFS Properties Configuration Requirements
- Requirements to Protect Business Data by Using ZFS Access Control Lists
- ZFS Delegated Administration Model Requirements
- Business Data Backup and Recovery Requirements

ZFS Properties: Requirements

Outstanding Property Management Company		Fantastic Products LLC	
ZFS properties requirements		**ZFS properties requirements**	
Pool	Snapshot display Storage auto expand Device auto replace	**Pool**	Boot file system (`bootfs`)
File systems	Storage quota User storage quota Group storage quota Storage reservation	**File systems**	Compression File change ownership
		Non-global zone	Zoned Checksum

The next part of the data management plan specifies the ZFS properties required on both the OPM and FPL application pools and the file systems. Because the plan states there is a requirement to create backup snapshots for the Outstanding Property Management application data, you enable the snapshot display property. Your business analyst expressed concern about using too much storage for this application, so to address this concern, you will use the storage auto expand property, which will help us expand the storage as needed.

For the OPM application, the business analyst expressed concern about disk failure because OPM data is so critical. To address this concern, you will enable the device auto replace property. Because there was a requirement to share the manufacturer information, you will enable the NFS and CIFS file sharing properties. Based on the requirement to manage the storage efficiently, you will create storage quotas and make storage reservations for several of the application's users.

The FPL application uses an alternate boot environment. You observe the behavior of the `bootfs` property while the system transitions from one BE (`solaris`) to the FPL BE (`fplsolaris`). There is also a requirement for you to use the compression property on the file systems to save storage costs as well as to enable the file change ownership property.

In addition, you will configure the `zoned` property and verify the setting of the `checksum` property on the shared file system `mfrfs` in the `fpzone` non-global zone.

Quiz

Which business application plans to use the ZFS compression property?

a. Outstanding Property Management Company

b. Fantastic Products LLC

Answer: b

Lesson Agenda

- Data Storage Requirements for the Business Application
- Root Pool Management Requirements
- Oracle Solaris ZFS Components Configuration and Management Requirements
- ZFS Properties Configuration Requirements
- Requirements to Protect Business Data by Using ZFS Access Control Lists
- ZFS Delegated Administration Model Requirements
- Business Data Backup and Recovery Requirements

Data Protection Requirements Using ZFS ACLs

Fantastic Products LLC	
ZFS ACLs	
Configure ACLs for Directories and Files in Verbose Format	Create a group called `fplgroup` and assign the users Polly Anna (`panna`) and Jerry Moose (`jmoose`) to the this group. Set permissions for the members of `fplgroup` to read, write, and execute on the the Beauty Aids Product Line (`baprodline`) and Health Products Product Line (`hpprodline`) file systems in the `mfrs` pool.

Manage the ACLs	Set permissions on the Beauty Aid product line data as follows:
	• Jerry Moose (`jmoose`): `baline1`directory: Grant write data and add subdirectory permissions.
	• Polly Anna (`panna`):
	  • `baline2` directory: Grant read and write; set `file_inherit`.
	    • `nails1`: Read only
	    • `skinbeauty`: Read, write, and execute (set `aclmode` property y to `passthrough`)

Configure ACLs in Compact Format	• John Holt (`jholt`): `hairpins`: Read and execute • Polly Anna (`panna`): `hairpins`: Read, write, and execute

Your business analyst determined that the FPL application's Beauty Aids Product Line (`baprodline`) and Health Products Product Line (`hpprodline`) data must be protected for privacy purposes. The additional level of data protection that the business wants is provided by ZFS access control lists (ACLs). You use the ZFS ACLs to restrict user access to the data on an “as-needed” basis.

The table in this slide identifies each ZFS ACL requirement and specifies which users and groups need which permissions based on their required level of access to the data.

Quiz

Which FPL application data requires ZFS ACL protection?

a. Manufacturer data

b. Owners' data

c. Beauty Aids product data

Answer: c

Lesson Agenda

- Data Storage Requirements for the Business Application
- Root Pool Management Requirements
- Oracle Solaris ZFS Components Configuration and Management Requirements
- ZFS Properties Configuration Requirements
- Requirements to Protect Business Data by Using ZFS Access Control Lists
- ZFS Delegated Administration Model Requirements
- Business Data Backup and Recovery Requirements

ZFS Delegated Administration Model: Requirements

Outstanding Property Management Company		Fantastic Products LLC	
ZFS Delegated Administration		**ZFS Delegated Administration**	
ZFS Rights Profiles	Create and assign the ZFS administration role (`zfsadmin`) to Jerry Moose (`jmoose`).	**ZFS Rights Profiles**	N/A
ZFS Permissions	N/A	**ZFS Permissions**	Grant ZFS delegated permissions to Polly Anna (`panna`) to include `destroy` on her home directory. Assign Polly Anna (`panna`), John Holt (`jholt`), and Jerry Moose (`jmoose`) to a group (`fplgroup`). Create a permission set (`adminset`) for the `mfrs` pool's file system and attach the permission set to `fplgroup`.

After considering the teams using each application, the business application analysts have determined that certain users of both the OPL and FPL applications should be given specific ZFS administration privileges. The users Polly Anna and Jerry Moose were selected for administering the ZFS file systems, snapshots, and clones. You are responsible for configuring a rights profile and assigning it to these users.

The OPM application needs to maintain growing information about the vacant and potentially new properties to be considered for prospecting. This task is assigned to Jerry Moose. As the ZFS storage administrator, you create a ZFS Administration role and assign it to Jerry Moose so he can administer the file systems in his home directory (file system) and maintain the new property information.

One application user must maintain information about all the major and minor business applications. This task is assigned to Polly Anna.

The FPL application must keep a record of all FPL users. This list includes the users of any sub-application or any software package used in the FPL application. Three users (Polly Anna, John Holt, and Jerry Moose) are given this responsibility. As part of this assignment, the three users need many permissions, including creating and destroying a file system. This will help them organize unique data in a different file system.

Quiz

Which users have been selected to administer the ZFS components? (Select one group of names.)

a. John Holt, Super Student, Polly Anna
b. Polly Anna, John Holt, Jerry Moose
c. Jerry Moose, Super Student, Polly Anna

Answer: b

Lesson Agenda

- Data Storage Requirements for the Business Application
- Root Pool Management Requirements
- Oracle Solaris ZFS Components Configuration and Management Requirements
- ZFS Properties Configuration Requirements
- Requirements to Protect Business Data by Using ZFS Access Control Lists
- ZFS Delegated Administration Model Requirements
- Business Data Backup and Recovery Requirements

Business Data:
Backup and Recovery Requirements

Outstanding Property Management Company		Fantastic Products LLC	
ZFS Data Backup		**ZFS Data Backup**	
ZFS Data Backup Using Snapshots	Take a monthly backup called `may2012` of the `opmpool/clients` file system.. Provide the differences between the `may2012` and `june2012` snapshots of `opmpool/clients/renters` to the application analyst. Create a clone of the `renters@june2012` snapshot and promote it	**ZFS Data Backup Using Snapshots**	Set up automatic snapshots on `mfrs/fplusers` every 10 minutes.
ZFS Data Remote Backup	Back up the `renters@june2012` snapshot to a remote system. Create a ZFS recovery archive for `admpool` and recover the data.	**ZFS Data Remote Backup**	Create a ZFS recovery archive for `admpool` and recover the data. **Note:** This is a shared requirement.

The FPL application analyst expects frequent updates to the FPL users file system (`fplusers`). Therefore, you must ensure that `fplusers` is backed up every 10 minutes. The application analyst is concerned only about keeping an updated copy of the user data. The objective is to keep a monthly, weekly, and every-10-minute backup so that, at any given time, there are only three updated copies saved. You, as a ZFS storage administrator, are responsible for configuring the automatic snapshots, which are efficient and resource saving.

The OPM application analyst wants you to take monthly backups of the OPM application's `clients` file system, which includes owner and renter information. In addition, the OPM application analyst wants you to back up the renters information on a remote system. By saving the data offsite, you can recover the data in case of a local disaster. In addition, you are asked to create a recovery archive of OPM data to provide a means to recover the data in case the original data becomes corrupted.

Quiz

Which application requires the user to back up the file system every 10 minutes?

a. Outstanding Property Management Company
b. Fantastic Products LLC

Answer: b

Practice 3-1 Overview: Determining the Existing Configuration

This practice covers the following topics:

- Determining the disk for the root pool
- Determining the available storage capacity
- Inspecting the existing delegated administration model

You have completed your data management requirements gathering and are now ready to begin implementing the plan. Your first task is to determine the existing configuration, which you do as part of this practice. The practice should take about 15 minutes to complete.

Summary

In this lesson, you should have learned how to identify the requirements for:

- The business application's data storage
- ZFS root pool management
- Configuration and management of the Oracle Solaris ZFS components
- ZFS properties configuration
- Using ZFS access control lists to protect business data
- The ZFS delegated administration model
- Business data backup and recovery

4

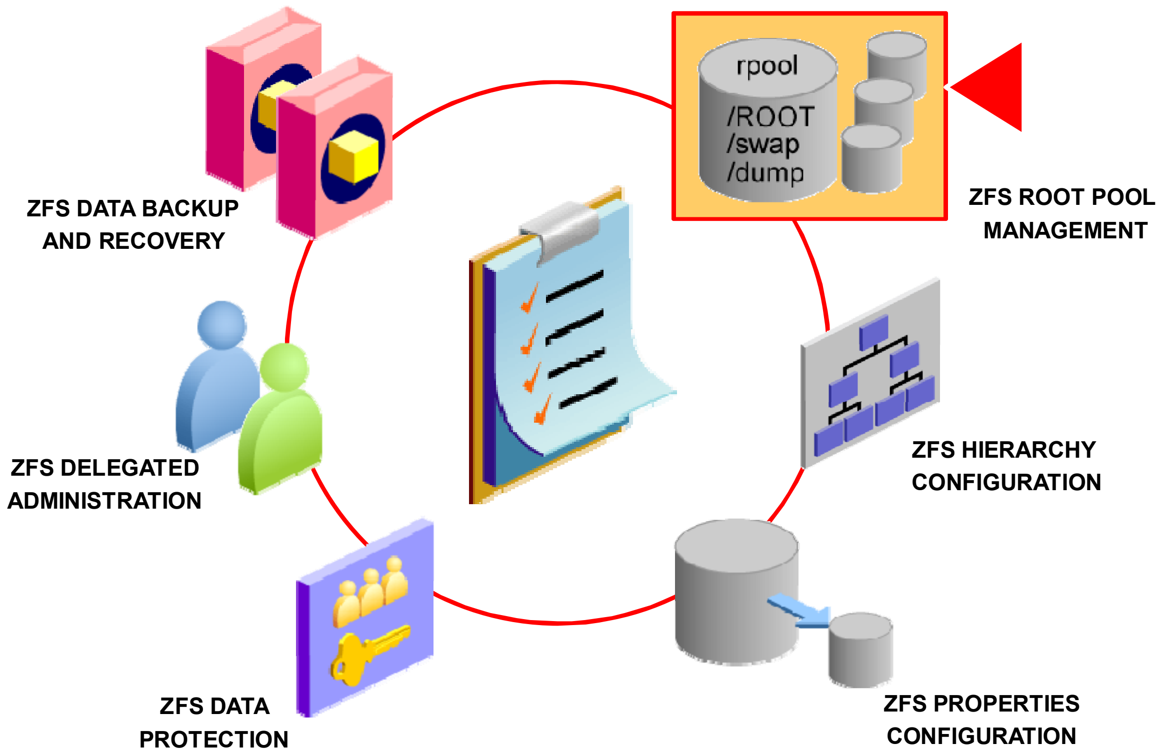
Managing the ZFS Root Pool

Objectives

After completing this lesson, you should be able to:

- Implement a plan to manage the ZFS root pool
- Manage the ZFS swap and dump devices
- Troubleshoot ZFS dump device issues
- Boot from an alternate ZFS root file system

Planning Workflow: Orientation



This course presents each of the planning tasks in the context of a workflow. We use the graphic shown in the slide at the beginning of each lesson to remind ourselves where we are in the workflow. We discuss why certain tasks precede or follow other tasks and the importance of each task as it pertains to the data management plan.

As indicated in the graphic, we start with how to manage the ZFS root pool, followed by how to configure and manage the ZFS pools and file systems. The next task in our data management plan is to configure the ZFS properties for our ZFS components. We then look at how to protect our ZFS data. We then focus on using ZFS delegated administration. After we know how we are going to protect and administer the ZFS data, we look at how to back it up. We finish our planning tasks with archiving and recovering ZFS data.

In this lesson, we discuss the importance of the ZFS root pool and the preventive steps you can take to ensure that you can recover the root pool if the boot data becomes corrupted.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Root Pool
- Managing the ZFS Swap and Dump Devices
- Booting from an Alternate ZFS Root File System

Implementing the Plan to Manage the ZFS Root Pool

System Level	Outstanding Property Management Company	Fantastic Products LLC
Inspect swap and dump devices.		
• Increase swap and dump to 1.5 GB each. • Add a 1 GB independently manageable swap volume.		• Requirement for current project • Requirement for the new project
Inspect the current boot environment.		
Boot from an alternate disk in a mirrored ZFS root pool.		
Boot from a backup ZFS root file system.		
Boot an unbootable system for system recovery.		

ORACLE

It's now time to implement the plan for managing the ZFS root pool. The ZFS root pool system-level requirements, as well as the swap and dump requirements for the Fantastic Products LLC application, are presented in the slide. In the sections that follow, you learn how to complete each of the ZFS root pool tasks you have been assigned.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Root Pool
- Managing the ZFS Swap and Dump Devices
- Booting from an Alternate ZFS Root File System

Managing the ZFS Swap and Dump Devices

This section covers the following topics:

- Inspecting the swap and dump devices
- Adjusting the sizes of the ZFS swap and dump devices
- Adding a new ZFS swap volume
- Removing a ZFS swap volume
- Removing a ZFS dump device
- Troubleshooting ZFS dump device issues

Inspecting the Swap and Dump Devices

```
# zfs list -r rpool
NAME                      USED  AVAIL  REFER  MOUNTPOINT
rpool                    10.2G  21.0G    39K  /rpool
rpool/ROOT               2.14G  21.0G    31K  legacy
rpool/ROOT/solaris       2.14G  21.0G  1.58G  /
rpool/ROOT/solaris/var    513M  21.0G   369M  /var
rpool/dump               1.03G  21.1G  1.00G  -
…
…
rpool/swap               1.03G  21.1G  1.00G  -
```

`/dev/zvol/dsk/rpool/dump`

`/dev/zvol/dsk/rpool/swap`

The sizes of the swap and dump volumes are dependent on the amount of physical memory on the system.

- Swap device size is generally 1/4 of physical memory.
- Dump device size is approximately half the size of physical memory.

ORACLE

As the Oracle Solaris 11 operating system is being installed on a system, separate ZFS volumes in the ZFS root pool are automatically created for a swap area and a dump device as part of the ZFS root file system installation process. These volumes are identified as devices in the `/dev/zvol/{dsk,rdsk}/pool` directory.

Note: A ZFS volume is a dataset that represents a block device.

The sizes of the swap volume and the dump volume are dependent on the amount of physical memory on the system. The swap device size is generally 1/4 of the physical memory, and the dump device size is approximately half the size of the physical memory. These sizes, however, can vary as illustrated in the slide example, where the swap size (1 GB) is half of the physical RAM.

Note: The default swap size is generally between 512 MB and 2 GB.

Note: You cannot use the same ZFS volume for both the swap area and a dump device.

Adjusting the Size of the ZFS Swap Device

To reset the size of the swap device:

1. Use `zfs get all rpool/swap` to review the following swap volume storage allocation properties:
   - `swap used`
   - `swap referenced`
2. Use `swap -l` to display the swap memory allocation.
3. Use `zfs get volsize rpool/swap` to determine the current size of the swap volume.
4. Use `zfs set volsize=`*`size`*` rpool/swap` to modify the swap memory allocation.
5. Use `zfs get volsize rpool/swap` to verify the change.
6. Use `swap -a /dev/zvol/dsk/rpool/swap` to allocate the swap space.

ORACLE

If business needs require it, you can adjust both the swap volume size and the dump volume size after installation. You might recall from the "Planning for Data Management" lesson that the Fantastic Products LLC application needs additional swap to address its anticipated increase in transaction rates (from 1 GB to 1.5 GB) and additional dump storage to address the potential instability of this new application (from 1 GB to 1.5 GB).

To complete this task, you need to adjust the sizes of both the swap and dump devices. The steps for adjusting the size of the swap device are shown in the slide.

Notes for step 1: You want to review the swap properties that indicate the amount of storage allocated to the swap device. For a definition of each property, see the `zfs(1M)` man page.

Notes for step 2: The `swap -l` command displays swap device details such as the device name. To see the total swap space usage and availability, you can use the `swap -s` command. Example output from this command is as follows:

```
# swap -s
total: 110532k bytes allocated + 26216k reserved = 136748k used,
2100812k available
```

The output of this command shows the following:

- **`allocated`:** Total amount of swap space in bytes currently allocated for use
- **`reserved`:** Total amount of swap space in bytes not currently allocated but claimed by memory mappings for possible future use
- **`used`:** Total amount of swap space in bytes that is either allocated or reserved
- **`available`:** Total amount of swap space in bytes that is currently available for future allocation and reservation

These numbers include swap space from all configured swap areas as listed by the `-l` option as well as swap space in the form of physical memory.

You can use the amount of available and used swap space (in the `swap -s` output) as a way to monitor swap space usage over time. If a system's performance is good, use `swap -s` to determine how much swap space is available. When the performance of a system slows down, check the amount of available swap space to determine if it has decreased. Then you can identify what changes to the system might have caused swap space usage to increase. When using the `swap –s` command, keep in mind that the amount of physical memory available for swap usage changes dynamically as the kernel and user processes lock down and release physical memory.

Adjusting the Size of the ZFS Swap Device: Example

```
# zfs get all rpool/swap
NAME        PROPERTY              VALUE                  SOURCE
<output omitted>
rpool/swap  used                  1.03G                  -
<output omitted>
rpool/swap  referenced            1.00G                  -
<output omitted>
# swap -l
swapfile                     dev     swaplo   blocks     free
/dev/zvol/dsk/rpool/swap    124,2         8  2097144  2097144
# zfs get volsize rpool/swap
NAME        PROPERTY  VALUE    SOURCE
rpool/swap  volsize   1G       local
# zfs set volsize=1.5G rpool/swap
# zfs get volsize rpool/swap
NAME        PROPERTY  VALUE    SOURCE
rpool/swap  volsize   1.50G    local
# swap -a /dev/zvol/dsk/rpool/swap
#
```


In this example, we adjust the size of the swap device. We first use `zfs get all rpool/swap` to review the swap volume used and the referenced properties that affect storage allocation.

We then use `swap -l` to display the swap memory allocation.

Note: The `swap -l` command lists the status of all the swap areas. The status includes the following:

- Path name for the swap area
- Device major and minor numbers (`dev`)
- Swaplow value (`swaplo`), which represents the starting address of the memory, which is also on the memory page boundary (8 × 512 = 4096 bytes). In this example, the memory page size is 4 KB (4096).
- Swaplen value (`blocks`), which represents the number of 512-byte blocks of total memory that are available
- Number of free 512-byte blocks (`free`)

This list does not include swap space in the form of physical memory because this space is not associated with a particular swap area. For more information about swaplow and swaplen, see the man pages.

We are now ready to adjust the memory allocation of the swap volume from 1 GB to 1.5 GB. We first run the `zfs get volsize rpool/swap` command to verify that the current size of the swap volume is 1 GB. And it is. We then use the `zfs set volsize=1.5G rpool/swap` command to adjust the size of the volume.

We use the `zfs get volsize rpool/swap` command to verify the change. In the command output, you can see that the `volsize` property is now set to 1.5 GB.

The final step is to allocate the swap space by using the `swap –a /dev/zvol/dsk/rpool/swap` command.

Adjusting the Size of the ZFS Dump Device

To reset the size of the dump device:

1. Use `zfs get all rpool/dump` to review the following swap volume storage allocation properties:
   - `dump used`
   - `dump referenced`
2. Use `dumpadm` to list the attributes of the dump device.
3. Use `zfs get volsize rpool/dump` to determine the size of the current dump volume.
4. Use `zfs set volsize=`*`size`* `rpool/dump` to modify the storage of the dump device.
5. Use `zfs get volsize rpool/dump` to verify the change.

The steps for adjusting the size of the swap device are shown in the slide.

Note for step 1: As you did with the swap device, you want to review the dump properties that indicate the amount of storage allocated to the dump device. For a definition of each property, see the `zfs(1M)` man page.

Note for step 2: Confirm that the dedicated dump device is pointing to `rpool/dump`, which is the dump volume defined in the root pool.

Adjusting the Size of the ZFS Dump Device: Example

```
# zfs get all rpool/dump
NAME         PROPERTY              VALUE                 SOURCE
<output omitted>
rpool/dump   used                  1.03G                 -
<output omitted>
rpool/dump   referenced            1.00G                 -
<output omitted>
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash
  Savecore enabled: yes
   Save compressed: on
# zfs get volsize rpool/dump
NAME         PROPERTY  VALUE    SOURCE
rpool/dump   volsize   1G       local
# zfs set volsize=1.5G rpool/dump
# zfs get volsize rpool/dump
NAME         PROPERTY  VALUE    SOURCE
rpool/dump   volsize   1.50G    local
```


In this example, we adjust the size of the dump device. We first use `zfs get all rpool/dump` to review the dump volume `used` and `referenced` properties that affect storage allocation. The volume appears to take up about 1 GB of storage.

We then use `dumpadm` to list the attributes of the dump device. Note the directory location of the device: `/dev/zvol/dsk/rpool/dump`.

We are now ready to adjust the storage allocation of the dump device volume from 1 GB to 1.5 GB. We first run the `zfs get volsize rpool/dump` command to verify that the current size of the volume is 1 GB. And it is. We then use the `zfs set volsize=1.5G rpool/dump` command to adjust the size of the volume.

The final step is to use the `zfs get volsize rpool/dump` command to verify the change. In the command output, you can see that the `volsize` property is now set to 1.5 GB.

Note: If you disable and remove the dump device, you will need to re-create it and then enable it with the `dumpadm` command.

Adding a New ZFS Swap Volume

To add a new swap volume:

1. Use `zfs create -V 1G rpool/`*`swapname`* to create a new swap volume.
2. Use `zfs list -r /rpool` to verify the change.
3. Use `swap -a /dev/zvol/dsk/rpool/`*`swapname`* to activate the new swap area.
4. Use `swap -l` to confirm the swap area availability.
5. Use the `vi` editor to add an entry in the `/etc/vfstab` file for the new swap area.

If you need more swap space and the swap device is busy, you can add another swap volume. On an active system, for example, you might consider adding a second swap volume to increase overall swap size. The steps for adding a new swap volume are listed in the slide.

Note: Using a swap file on a ZFS file system is not currently supported. You must use a volume.

Note for step 1: The `-V` option specifies that a ZFS volume is being created.

Note for step 5: Swap areas are listed in the `/etc/vfstab` file. A swap device entry in the `/etc/vfstab` file contains the full path name of the swap volume path name. The entry you make as part of adding a new swap volume makes the new swap allocation permanent for your system.

Adding a New ZFS Swap Volume: Example

```
# zfs create -V 1G rpool/swap2
# zfs list -r rpool
NAME                         USED  AVAIL  REFER  MOUNTPOINT
rpool                       10.2G  21.0G    39K  /rpool
<output omitted>
…
…
rpool/swap                  1.55G  19.6G  1.50G  -
rpool/swap2                 1.03G  19.5G  1.00G  -
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
swapfile                      dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap    124,2         8  2097144  2097144
/dev/zvol/dsk/rpool/swap2   124,3         8  2097144  2097144
# vi /etc/vfstab
#device          device           mount            FS       fsck     mount    mount
#to mount        to fsck          point            type     pass     at boot  options
#
/devices         -                /devices         devfs    -        no       -
<output omitted>
fd               -                /dev/fd          fd       -        no       -
swap             -                /tmp             tmpfs    -        yes      -
/dev/zvol/dsk/rpool/swap          -                -                 swap     -        no

/dev/zvol/dsk/rpool/swap2         -                -                 swap     -        no
```


In this example, we add a new swap volume called `swap2`. We first create the new swap area by using the `zfs create -V 1G rpool/swap2` command. We then verify the change by using the `zfs list -r /rpool` command. We can see that `swap2` is now part of the root pool. The next step is to use the `swap -a /dev/zvol/dsk/rpool/swap2` command to activate the new swap area. We use `swap -l` to confirm the swap area availability. We can see that `swap2` is available. The final step is to use the `vi` editor to add an entry in the `/etc/vfstab` file for the new swap area.

Notice that the mount point for swap is `/tmp` and the file system type is `tmpfs`. The TMPFS file system stores files and their associated information in memory (in the `/tmp` directory) rather than on disk, which speeds access to those files. This feature results in a major performance enhancement for applications such as compilers and DBMS products that use `/tmp` heavily. The TMPFS file system allocates space in the `/tmp` directory from the system's swap resources. This feature means that as you use up space in the `/tmp` directory, you are also using up swap space. As a result, if your applications use the `/tmp` directory heavily and you do not monitor swap space usage, your system could run out of swap space.

Removing a ZFS Swap Volume

To remove a swap volume:

1. Use `swap -l` to determine if the swap device is in use.
2. Use `swap -d /dev/zvol/dsk/rpool/`*`swapname`* to remove the swap area.
3. Use the `vi` editor to delete the entry for the swap volume in the `/etc/vfstab` file.
4. Use `swap -l` to verify that the swap volume is no longer available.

If your system's swap space requirements change, you might need to remove a swap volume before you can increase or decrease space allocated for swap. The steps for removing a swap volume are listed in the slide.

Note for step 1: If the swap device is not being used, the number of blocks in the `blocks` field will equal the number of blocks in the `free` field.

Note for step 2: The `-d` option deletes the specified swap area.

Removing a ZFS Swap Volume: Example

```
# swap -l
swapfile                      dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap    124,2         8  2097144  2097144
/dev/zvol/dsk/rpool/swap2   124,3         8  2097144  2097144
# swap -d /dev/zvol/dsk/rpool/swap2
# vi /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
/devices        -               /devices        devfs   -       no      -
<output omitted>
fd              -               /dev/fd         fd      -       no      -
swap            -               /tmp            tmpfs   -       yes     -
/dev/zvol/dsk/rpool/swap        -               -               swap    -       no
     -
# swap -l
swapfile                      dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap    124,2         8  2097144  2097144
```


In this example, we remove a swap volume called `swap2`.

We first use the `swap -l` command to verify that the swap area is not in use. We can tell that the swap device is not in use because the numbers of blocks displayed in the `blocks` and `free` fields are equal to one another.

The next step is to use the `swap -d /dev/zvol/dsk/rpool/swap2` command to remove the swap area. The final step is to use the `vi` editor to delete the swap volume entry in the `/etc/vfstab` file. We use `swap -l` to confirm that the swap volume is no longer available; it is not.

Removing a ZFS Dump Device

To remove the default dump device:

1. Use `zfs create -V *value* *pool*/dump` to create a dump volume in a pool.
2. Use `dumpadm -d /dev/zvol/dsk/*pool*/dump` to allocate the volume to the system dump facilities.
3. Use `zfs destroy rpool/dump` to remove the default dump device.
4. Use the `zfs list rpool/dump` to verify that the volume has been deleted.

The ZFS dump device usually requires no administration because it is set up automatically at installation time. However, if you choose to remove the default dump device and create a dump device in a non-root pool, there are a few things you need to keep in mind. For example, there is no command that allows you to directly delete, remove, or destroy a dump device. Instead, you have to deallocate the device by switching to another device, and then you can destroy the ZFS `rpool/dump` volume. The steps for removing the ZFS default dump device are presented in the slide.

Note: If you choose at some point to re-create the dump device in the root pool, you can do so by using the steps listed here and switching *pool*/dump and `rpool/dump`.

Removing a ZFS Dump Device: Example

```
# zfs create -V .5G admpool/dump
# dumpadm -d /dev/zvol/dsk/admpool/dump
      Dump content: kernel pages
       Dump device: /dev/zvol/dsk/admpool/dump (dedicated)
Savecore directory: /var/crash
  Savecore enabled: yes
   Save compressed: on
# zfs destroy rpool/dump
# zfs list rpool/dump
cannot open 'rpool/dump': dataset does not exist
```


In this example, we remove the default dump device.

Suppose that we have decided to create a dump volume in a pool called `admpool`. We first create the dump volume in `admpool`. We then allocate the `admpool/dump` volume to the system dump facilities. Our final step is to delete the default dump volume and verify that it has been removed from the system; it has been removed.

Troubleshooting ZFS Dump Device Issues

- If you are experiencing problems in capturing a system crash dump:
  - Check the size of the dump device; it may be too small. Correct the problem by increasing the size of the dump device (`zfs set volsize=`*`size`*` rpool/dump`).
  - Verify that the dump device is enabled. Correct the problem by enabling the dump device (`dumpadm -d /dev/zvol/dsk/rpool/dump`).
- If you cannot add a dump device to a pool, ensure that you are trying to add the dump device to the root pool.
- If the default dump device was removed and you have created a new dump device manually, check that the dump device is enabled.

Let's take just a minute to discuss how to troubleshoot a few common issues associated with dump devices. If you are having problems capturing a system crash dump, the reason might be that the dump device is too small to capture the existing crash dump. For example, a system with 128 GB or greater memory will need a larger dump device than the dump device that is created by default. If the dump device is too small to capture an existing crash dump, you should receive a message indicating that it is too small.

To correct the problem, increase the size of the dump device by using the `zfs set volsize=size rpool/dump` command.

Note: Resizing a large dump device can be a time-consuming process.

Another reason you might be having problems capturing a system crash dump is if the dump device is disabled. If the default dump device has been removed and a new dump device has been manually created but never enabled, the dump device will not work properly. To correct this problem, enable the device by using the `dumpadm -d /dev/zvol/dsk/rpool/dump` command as in the following example:

```
# dumpadm -d /dev/zvol/dsk/rpool/dump
    Dump content: kernel pages
     Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash
  Savecore enabled: yes
   Save compressed: on
```

Note: The `-d` option is used to modify a dump configuration

If you are having trouble adding a dump device to a pool, it might be because you are trying to add the dump device to a pool with multiple top-level devices. You should always add the dump device to the root pool.

Note: A top-level device error message is displayed only when you are using three plain top-level devices. Otherwise, if you are trying to add the device to a RAID-Z or mirrored storage pool, the message is different. If you add a dump device to an application pool with only one device, the system accepts it.

For more information about managing ZFS swap and dump devices, see “Managing Your ZFS Swap and Dump Devices” in the *Oracle Solaris Administration: ZFS File Systems* guide.

Quiz

The size of the swap volume is dependent on the amount of physical memory on the system.

a. True

b. False

Answer: a

Quiz

Which statement is true about ZFS swap and dump devices?

a. During installation, no swap or dump devices are created in the ZFS root pool. They must be created manually.
b. During installation, by default, one swap device and one dump device are created in the ZFS root pool.
c. During installation, by default, two swap devices and one dump device are created in the ZFS root pool.
d. During installation, by default, one swap device is created in the ZFS root pool. The dump device must be created manually.

Answer: b

Quiz

You receive the following error message:

```
dumpadm: dump device /dev/zvol/dsk/rpool/dump is too small to
hold a system dump
dump size 36255432704 bytes, device size 34359738368 bytes
```

What action should you take to correct the issue?

a. Rename the dump device.
b. Increase the size of the dump device.
c. Change the dump device to a slice.
d. Verify that the dump device has been enabled. If it has not, enable it.

Answer: b

Quiz

You receive the following error message:

```
dump is not supported on device '/dev/zvol/dsk/datapool/dump':
'datapool' has multiple top level vdevs
```

What is the issue?

a. The size of the dump device is too small.
b. The dump device needs to be enabled.
c. Another dump device has already been configured.
d. A dump device cannot be added to a pool with multiple virtual top-level devices.

Answer: d

Practice 4-1 Overview: Managing the ZFS Swap and Dump Devices

This practice covers the following topics:

- Inspecting the swap and dump devices
- Adjusting the sizes of the swap and dump devices
- Adding a new ZFS swap volume
- Removing a ZFS swap volume
- Troubleshooting ZFS dump device issues

This practice should take about 45 minutes to complete.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Root Pool
- Managing the ZFS Swap and Dump Devices
- Booting from an Alternate ZFS Root File System

ORACLE

Booting from an Alternate ZFS Root File System

This section covers the following topics:

- Inspecting the current boot environment
- Creating a mirrored ZFS root pool
- Booting from an alternate disk in a mirrored ZFS root pool
- Booting from an alternate ZFS root file system on an x86-based system
- Booting from an alternate ZFS root file system on a SPARC-based system
- Booting in a ZFS root environment for recovery purposes

Now you are going to work with one of the most important functions in Oracle Solaris 11: the operating system booting function. As you know from the lesson titled "Planning for Data Management," the ZFS root pool contains the file system for booting the Solaris operating system. If the boot data becomes corrupted, the system cannot be booted without manual intervention. Therefore, it is critical that system administrators understand the boot mechanism and know alternative ways to boot the system if the boot data becomes corrupted.

Inspecting the Current Boot Environment

- Use `zpool status` to determine the disk used by the root pool.
- Use `format` to view the storage capacity of the disk.
- Use `more` to display the `menu.lst` file and identify the file system selected for booting by locating the pool's `bootfs` entry.

```
#
title Oracle Solaris 11 11/11
bootfs rpool/ROOT/solaris
```


Before making any modifications to the ZFS boot environment, you should familiarize yourself with the current boot environment. When inspecting the boot environment, the first thing you do is run the `zpool status` command to determine which disk is being used by the root pool. This is your boot disk. You also want to know the storage capacity of this disk. To view the storage capacity, use the `format` command.

Note: You can determine the amount of storage capacity you need by considering the size of the operating system (3 GB to 5 GB), the possible need to install IPS (approximately 6 GB in size), the number of BEs, and any extra packages.

Finally, you display the `menu.lst` file by using the `more` command. Locate the pool's `bootfs` entry to identify the file system that was selected by default during installation and the `beadm` activation process for booting the system. An example of the `bootfs` entry is shown in the slide, where you can see that the boot file system is the root pool file system.

Creating a Mirrored ZFS Root Pool

1. Use `zpool status rpool` to display your current root pool status.
2. Prepare a second disk for attachment to the root pool (if necessary).
3. Use `zpool attach rpool` *`device device`* to attach a second disk to configure a mirrored root pool.
4. Use `zpool status rpool` to view the root pool status to confirm that resilvering is complete.
5. Verify that you can boot successfully from the new disk.
6. Set up the system to boot automatically from the new disk.

You learned in the lesson titled "Planning for Data Management" that having a mirrored root pool is a best practice because it reduces the down time caused by hardware failures. You can create a mirrored ZFS root pool during installation or you can configure it afterward. The steps for creating a mirrored ZFS root pool are shown in the slide.

Note for step 2: For an x86-based system, you need to confirm that the disk has an `fdisk` partition, an SMI disk label, and a slice 0. For a SPARC-based system, you need to confirm that the disk has an SMI (VTOC) disk label and a slice 0.

Note for step 4: Resilvering is not complete until the resilvered message is displayed in the scan field of the `zpool status` command output.

Note for step 6: For an x86-based system, you set up the system to boot automatically from the new disk by reconfiguring the BIOS. For a SPARC-based system, you use either the `eeprom` command or the `setenv` command from the boot PROM.

Booting from an Alternate Disk in a Mirrored ZFS Root Pool

On an x86-based system:

1. Configure the alternate disk in the appropriate BIOS menu.
2. Confirm the active boot device after the system has rebooted by using `prtconf -v|sed -n '/bootpath/,/value/p'`.

On a SPARC-based system:

1. Enter the alternate disk at the `ok` prompt by using `boot`, followed by the full path name.
2. Confirm the active boot device after the system has rebooted by using `prtconf -vp | grep bootpath`.

After configuring a mirrored ZFS root pool, you have the option of booting from either disk in the configuration. The steps for completing this task on x86-based and SPARC-based systems are shown in the slide.

Booting from an Alternate ZFS Root File System on an x86-Based System

1. Use `beadm list` to determine the default boot environment.
2. Use `beadm create` to create another boot environment. Verify the creation by using `beadm list`.
3. Verify that an entry for the new boot environment appears in the `menu.lst` file.
4. Use `beadm activate` to activate the new boot environment. Confirm the activation by using `beadm list`.
5. Issue `init 6` to restart the system.

As you learned in the lesson titled "Planning for Data Management," there is another preventive measure that you can take to protect against a corrupt boot file system: creating an alternate bootable file system or alternate BE. You can edit the `menu.lst` file to specify an alternate BE in the pool. The `menu.lst` file can contain entries for multiple root file systems in the pool.

The steps for creating an alternate ZFS root file system BE differ slightly between x86-based and SPARC-based systems. The steps for the x86-based system are shown in this slide. The steps for a SPARC-based system are shown in the next slide.

Note for step 3: An example of the entry in the `menu.lst` file for the new boot environment follows. In the example, a new boot environment called `solaris2` has been created in the ZFS root pool.

```
title solaris2

bootfs rpool/ROOT/solaris2

kernel$ /platform/i86pc/kernel/amd64/unix -B $ZFS-
BOOTFS,console=graphics

module$ /platform/i86pc/amd64/boot_archive
```

Note for step 4: The newly activated boot environment is activated on the next system restart.

Booting from an Alternate ZFS Root File System on a SPARC-Based System

1. Use `beadm list` to determine the default boot environment.
2. Use `beadm create` to create another boot environment. Verify the creation by using `beadm list`.
3. Verify that an entry for the new boot environment appears in the `menu.lst` file.
4. Use `beadm activate` to activate the new boot environment. Confirm the activation by using `beadm list`.
5. Issue `init 0` to switch to the boot prom layer.
6. Use `boot -L` to select the new boot environment.
7. Use `boot -Z rpool/ROOT/`*`newBE`* to boot the new boot environment.

The steps for creating an alternate boot environment on a SPARC-based system (as shown in steps 1–4 in the slide) are the same as those used on an x86-based system. The differences begin with rebooting the system in step 5.

Booting in a ZFS Root Environment for Recovery Purposes

1. Select the appropriate boot method.
2. Select the boot recovery problem.
   a. Resolve a bad root shell by booting the system to single-user mode and correcting the shell entry in the `/etc/passwd` file.
   b. Resolve a problem with a `menu.lst` boot entry.
   c. Resolve an unknown root password that prevents you from logging in to the system.
3. Set the root password by booting to single-user mode and setting the password.

ORACLE

Now let's look at what actions you can take to recover the ZFS root environment if your system will not boot. If your system will not boot, there might be a problem with the `menu.lst` file or root password. You can attempt to resolve the issue by using the steps shown in this slide.

Note for step 1: The boot method options are as follows:

- **x86: Live Media:** Boot from the installation media and use a GNOME terminal for the recovery procedure.
- **SPARC: Text installation:** Boot from the installation media or from the network, and select option 3 Shell from the text installation screen.
- **x86: Text installation:** From the GRUB menu, select the Text Installer and command line boot entry, and then select the option 3 Shell from the text installation screen.
- **SPARC: Automated installation:** Use the `ok` **`boot net:dhcp`** command to boot directly from an installation menu that allows you to exit to a shell.
- **x86: Automated installation:** Booting from an installation server on the network requires a PXE boot. Select the Text Installer and command line entry from the GRUB menu, and then select the option 3 Shell from the text installation screen.

Note for step 2a: To resolve this problem on an x86 system, edit the selected boot entry and add the `-s` option. On a SPARC system, shut down the system and then boot to single-user mode. After you log in as `root`, edit the `/etc/passwd` file and fix the root shell entry.

Note for step 2b: To resolve this problem, boot from media or the network by using one of the boot methods listed in step 1, import the root pool, and then fix the `menu.lst` entry. The next action is to update the boot archive by using either the `bootadm update-archive` or the `bootadm update-archive –R /a` command. The `-R/a` option updates an alternate boot archive. You run the `update-archive` command to ensure that all the device and necessary file pointers are updated. You should make a practice of updating the boot archive any time you make changes to the system. The final action for this recovery step is to confirm that the system boots successfully.

Note: The `bootadm` command manages the boot archive and, with x86 boot environments, the GRUB (GRand Unified Bootloader) menu. OpenBoot PROM (OBP)–based machines, such as SPARC systems, do not use GRUB and have no boot menu that is manageable by `bootadm`.

Note for step 2c: To resolve this problem, boot from media or the network by using one of the boot methods listed in step 1, import the root pool, and then mount the boot environment to remove the root password entry. This process is identical on x86 and SPARC platforms. To set the root password, complete step 3.

Note for step 3: This step assumes that you have removed an unknown root password in step 2c. To resolve this problem on an x86 system, edit the selected boot entry and add the `-s` option. On a SPARC system, boot the system to single-user mode, log in as root, and then set the root password.

Some problems might require you to replace a disk in the root pool. You learn how to replace a disk in the lesson on configuring ZFS properties.

Note: If you replace a root pool disk by using the `zpool replace` command, you must install the boot information on the newly replaced disk by using the `installgrub` command on an x86-based system and the `installboot` command on a SPARC-based system. Examples of each command:

- `x86#` **`installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c0t1d0s0`**
- `sparc#` **`installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdsk/c0t1d0s0`**

Quiz

Which entry in the `menu.lst` file identifies the file system that is selected by default during installation for booting the system?

a. `title`

b. `bootfs`

c. `kernel$`

d. `module$`

Answer: b

Quiz

When you create a new boot environment, a `bootfs` entry is automatically made in the `menu.lst` file for the new BE.

a. True
b. False

Answer: a

Practice 4-2 Overview: Booting from an Alternate ZFS Root File System

This practice covers the following topics:

- Inspecting the current boot environment
- Booting from an alternate disk in a mirrored ZFS root pool
- Booting from an alternate ZFS root file system on an X86-based system
- Booting from an alternate ZFS root file system on a SPARC-based system
- Booting in a ZFS root environment for recovery purposes

This practice should take about 65 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan to manage the ZFS root pool
- Manage the ZFS swap and dump devices
- Troubleshoot ZFS dump device issues
- Boot from an alternate ZFS root file system

5

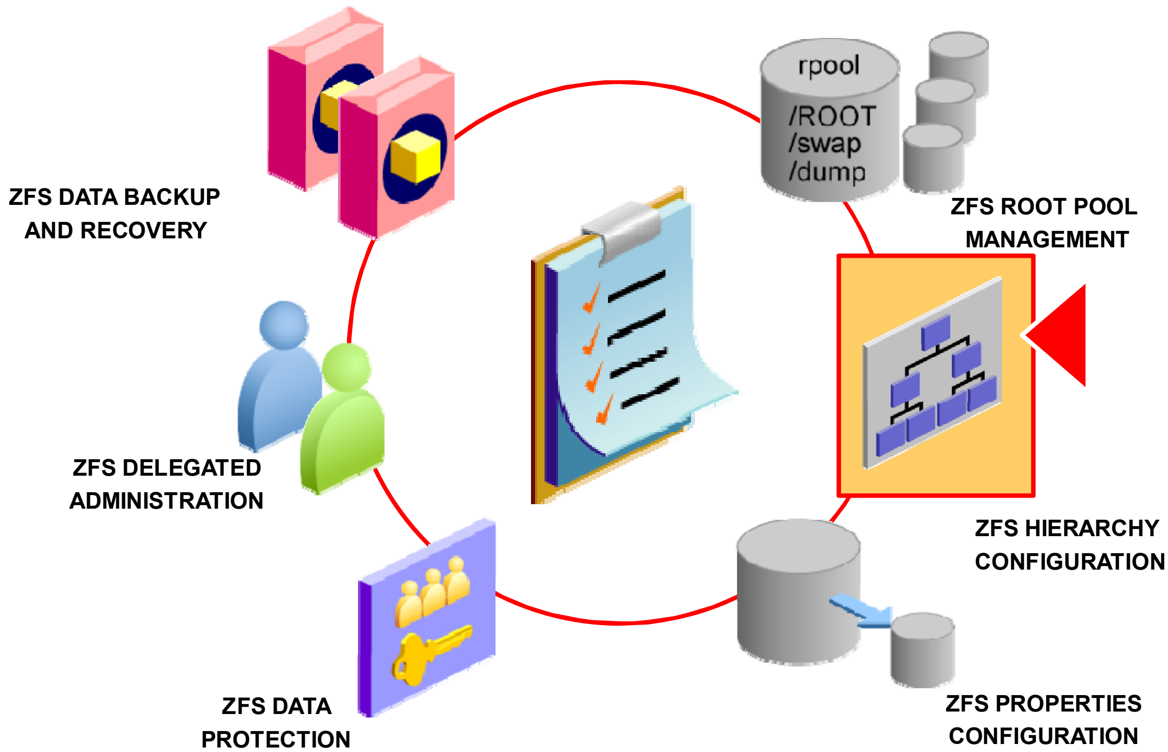
Configuring and Managing the Oracle Solaris ZFS Components

Objectives

After completing this lesson, you should be able to:

- Implement a plan to configure and manage the ZFS hierarchy
- Configure ZFS pools for the business application data
- Create ZFS file systems for the business application data
- Share ZFS file systems in a non-global zone
- Encrypt ZFS data
- Migrate ZFS data
- Upgrade ZFS components

Planning Workflow: Orientation



You have completed the ZFS root pool management tasks according to the data management plan. You are now ready to proceed to the next set of tasks, which require you to configure the ZFS components to support the pool and file system requirements identified in the plan.

In this lesson, you are shown how to build a scalable, flexible, and secure environment in which to house application data according to the needs and requirements of the business. You are also shown how to migrate and update ZFS pools and file systems.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Hierarchy
- Configuring the ZFS Hierarchy for a Business Application
- Sharing ZFS File Systems in a Non-Global Zone
- Encrypting ZFS Data
- Migrating ZFS Data
- Upgrading ZFS Components

Implementing the Plan to Manage the ZFS Hierarchy

Outstanding Property Management Company Requirements

Storage Entity Name	Data Storage Type	Data Type	Storage (Gigabytes)
opmpool	ZFS Mirrored Storage Pool	Property Management Application Data	4 GB
props	File System	Properties Data	Dynamic. As needed
clients	File System	OPM Clients Data	Dynamic. As needed
renters	File System	OPM Renters Data	Dynamic. As needed
owners	File System	OPM Property Owners Data	Dynamic. As needed
File System Backup	Snapshots	All OPM Data	Not required initially.

Fantastic Products LLC Requirements

Storage Entity Name	Data Storage Type	Data Type	Storage (Gigabytes)
fplpool	ZFS RAID-Z Storage Pool	Fantastic Products Application Data	3 GB
custfs	File System	Customers Data	Dynamic. As needed
mfrfs	File System	Product Manufacturers Data	Dynamic. As needed
bafs	File System	Beauty Aids Product Data	Dynamic. As needed
hpfs	File System	Health Products Data	Dynamic. As needed

In the lesson titled "Planning for Data Management," you were presented with the requirements for configuring ZFS pools and file systems for the business application data of both the Outstanding Property Management Company and Fantastic Products LLC. Now you implement the specifics of that plan.

In this lesson, you learn how to complete each of the ZFS hierarchy management tasks you have been assigned. These tasks include setting up the ZFS pools and file systems, populating the file systems with the application data, sharing the file systems in a non-global zone, enabling ZFS encryption on the file systems, migrating data, and upgrading the ZFS storage pools and file systems.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Hierarchy
- Configuring the ZFS Hierarchy for a Business Application
- Sharing ZFS File Systems in a Non-Global Zone
- Encrypting ZFS Data
- Migrating ZFS Data
- Upgrading ZFS Components

ORACLE

Configuring the ZFS Hierarchy for a Business Application

This section covers the following topics:

- Creating ZFS pools
- Creating ZFS file systems
- Populating the ZFS file systems with application data

Reviewing the ZFS Hierarchy Planning Requirements

Outstanding Property Management Company		Fantastic Products LLC	
Storage pool type and name	Type: Mirrored Name: `opmpool`	**Storage pool type and name**	Type: RAID-Z Name: `fplpool`
Number of disks	4 disks at 1 GB each		Three disks at 1 GB each
File systems	`props`	**File systems**	`custfs`
	`clients`		`mfrfs`
	`renters`		`bafs`
	`owners`		`hpfs`

According to the ZFS hierarchy requirements identified for the Outstanding Property Management Company during planning, you are tasked with creating a ZFS mirrored storage pool called `opmpool` that requires four disks. Because each disk has 1 GB of storage, you require four disks.

In addition, you need to create four file systems for this pool:

- **`props`:** Stores property information
- **`clients`:** Stores the company's clients
- **`renters`:** Sub file system in the `clients` file system that stores renter information
- **`owners`:** Sub file system in the `clients` file system that stores property owner information

You might recall from the lesson titled "Planning for Data Management" that there is also a requirement for backup snapshots for each file system. You are shown how to complete this task later in the course.

For Fantastic Products LLC, you are tasked with creating a RAID-Z storage pool called `fplpool` that requires three disks. Because each disk has 1 GB of storage, you need three disks.

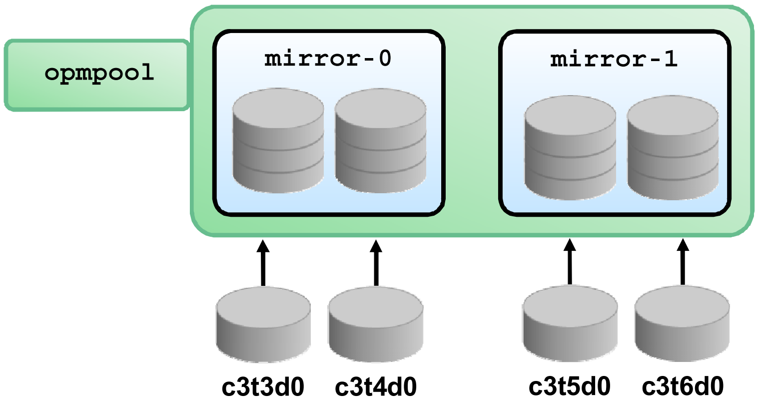
You need to create four file systems for this pool as well:

- **`custfs`**: Stores customer information
- **`mfrfs`**: Stores the company's manufacturers
- **`bafs`**: Stores beauty aids product information
- **`hpfs`**: Stores health products information

In this section, you are shown how to create each of the pool types (mirrored and RAID-Z). You are also shown how to create a file system and a sub file system. Finally, you are shown how to populate a file system with application data. During the practice that follows this section, you have the opportunity to create the pools and file systems, and you populate the file systems with data according to the requirements specified in the data management plan.

Creating a Mirrored Storage Pool

1. Use `format` to identify available disks for the new pool.
2. Use `zpool create` *`poolname`* `mirror` *`device device`* `mirror` *`device device`*.
3. Use `zpool list` to verify creation of the pool.
4. Use `zpool status` *`poolname`* to view the status of the pool.



The steps for creating a mirrored storage pool are presented in the slide.

Note for step 2: The number of devices in a mirrored storage pool can vary. Having two devices in each mirror, as shown in this step, is presented for example purposes only.

The graphic in the slide visually depicts the mirrored storage pool configuration for the Outstanding Property Management Company. There is a mirrored storage pool called `opmpool` with two mirrors: `mirror-0` and `mirror-1`. Each mirror contains two devices. Half of the data is on one device, and half of the data is on the other device.

Creating a Mirrored Storage Pool: Example

```
# zpool create opmpool mirror c3t3d0 c3t4d0 mirror c3t5d0 c3t6d0
# zpool list
NAME       SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
opmpool  1.97G   118K  1.97G   0%  1.00x  ONLINE  -
rpool    31.8G  10.1G  21.6G  31%  1.00x  ONLINE  -
# zpool status opmpool
  pool: opmpool
 state: ONLINE
  scan: none requested
config:

        NAME         STATE     READ WRITE CKSUM
        opmpool      ONLINE       0     0     0
          mirror-0   ONLINE       0     0     0
            c3t3d0   ONLINE       0     0     0
            c3t4d0   ONLINE       0     0     0
          mirror-1   ONLINE       0     0     0
            c3t5d0   ONLINE       0     0     0
            c3t6d0   ONLINE       0     0     0

errors: No known data errors
```


In this example, we create the mirrored storage pool called `opmpool` for the Outstanding Property Management Company.

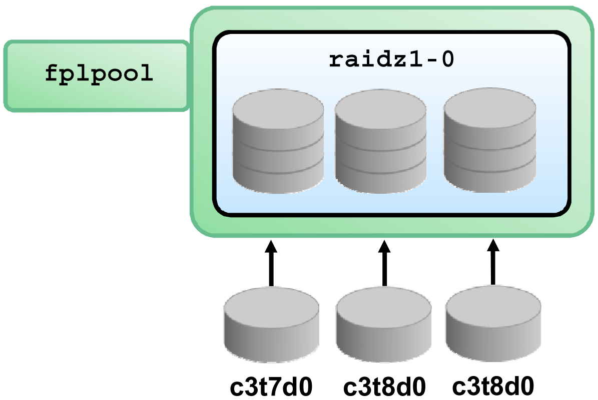
The first mirror contains devices `c3t3d0` and `c3t4d0`, and the second mirror contains devices `c3t5d0` and `c3t6d0`. Data is dynamically striped across both mirrors, with data being redundant between each disk in a mirror.

We then run the `zpool list` command to verify that the `opmpool` has been created.

We then check the status of the pool. The pool is in an `ONLINE` state, and the configuration is just as we specified: two mirrors (`mirror-0` and `mirror-1`) containing two devices each.

Creating a RAID-Z Storage Pool

1. Use `format` to identify available disks for the new pool.
2. Use `zpool create` *`poolname`* `raidz` *`device device device`*.
3. Use `zpool list` to verify creation of the pool.
4. Use `zpool status` *`poolname`* to view the status of the pool.



The steps for creating a RAID-Z pool are presented in the slide.

Note for step 2: The number of devices in a RAID-Z storage pool can vary. Having at least three devices in single-parity RAID-Z configuration, as shown in this step, is recommended. Single-parity RAID-Z (`raidz` or `raidz1`) is similar to RAID-5. In RAID-Z, ZFS uses variable-width RAID stripes so that all writes are full-stripe writes. If you have three disks in a single-parity RAID-Z configuration (as we do in our scenario), parity data occupies disk space equal to one of the three disks.

The graphic in the slide visually depicts the RAID-Z storage pool configuration for Fantastic Products LLC. There is a RAID-Z storage pool called `fplpool` with a single RAID-Z device: `raidz1-0`. The RAID-Z device consists of three disks.

Creating a RAID-Z Storage Pool: Example

```
# zpool create fplpool raidz c3t7d0 c3t8d0 c3t9d0
# zpool list
NAME       SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
fplpool   2.95G   176K  2.95G   0%  1.00x  ONLINE  -
opmpool   1008M  86.5K  1008M   0%  1.00x  ONLINE  -
rpool     31.8G  10.5G  21.3G  32%  1.00x  ONLINE  -
# zpool status fplpool
  pool: fplpool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        fplpool     ONLINE       0     0     0
          raidz1-0  ONLINE       0     0     0
            c3t7d0  ONLINE       0     0     0
            c3t8d0  ONLINE       0     0     0
            c3t9d0  ONLINE       0     0     0

errors: No known data errors
```

ORACLE

In this example, we create the RAID-Z storage pool called `fplpool` for Fantastic Products LLC. The pool contains three devices: `c3t7d0`, `c3t8d0`, and `c3t9d0`.

We then verify that the `fplpool` has been created by running the `zpool list` command. We then check the status of the pool. The pool is in an `ONLINE` state, and the configuration is just as we specified: one RAID-Z device (`raidz1-0)` that consists of three devices.

Note: At the `pool` level, you see all the available allocated storage. If you look at the ZFS `file system` level in the command and output that follows, you notice decreased available storage size due to the RAID-Z data parity calculations.

```
# zfs list /fplpool
NAME                        USED  AVAIL  REFER  MOUNTPOINT
fplpool                     117K  1.94G  34.6K  /fplpool
```

In the `zpool list` output in the slide, note that the total size of the pool is almost 3 GB, whereas in the `zfs list` output shown here the amount of available space is about 2 GB. In the `zfs list` output, approximately 1 GB of space is being used by the parity architecture.

Creating the ZFS File Systems

1. Use `zfs create -o mountpoint=/`*`path poolname/filesystemname`*.
2. Use `zfs list -r /`*`poolname`* to verify creation of the file system in a specified pool.

```
# zfs create -o mountpoint=/clients opmpool/clients
# zfs create -o mountpoint=/owners opmpool/clients/owners
# zfs list -r /opmpool
NAME                     USED  AVAIL  REFER  MOUNTPOINT
opmpool                  132K  1.94G    32K  /opmpool
opmpool/clients           31K  1.94G    31K  /clients
opmpool/clients/owners    31K  1.94G    31K  /owners
```


Now let's add the file systems to the ZFS storage pools. Because you use the same basic command to create all the file systems, the steps in the slide show how to create one file system.

Note for step 1: The `-o` option is used to specify a property and its associated value (`property=value`). In this case, you specify the `mountpoint` property and the value is `/path`, where `path` is the name of the file system in the ZFS namespace.

In the example, we first create the `clients` file system in the `opmpool`. We then create the `owners` file system as a sub file system to the `clients` file system according to the data management plan. We then verify creation of both file systems by running the `zfs list -r` command for the `opmpool` pool. Both the new file systems are listed.

Populating the ZFS File Systems with Application Data

1. Use `cd` to change to the directory in which you want to load the application data file.
2. Use `cp` to copy the file from its current location into the directory.
3. Use `ls` to verify that the file now resides in the directory.

```
# cd /owners
# cp /opt/ora/data/opm/ownersdallas dallas
# ls
dallas
```

ORACLE

Now that you have created the file systems, you can populate them with application data. The steps for copying application data into a file system are shown in the slide.

In the example, we change directories to the `owners` file system. We then copy the application data file called `dallas` from its current location in the `/opt/ora/data/opm/ownersdallas` to the `owners` file system. We then use the `ls` command to verify that the `dallas` file now resides in the `/owners` directory, and we see that the file is listed.

Quiz

You need to create a mirrored storage pool called `opmpool` that consists of two mirrors with four devices each. Which command do you use to create this pool?

a. `# zpool create opmpool mirror c3t3d0 c3t4d0 c3t5d0 c3t6d0 c3t7d0 c3t8d0 c3t9d0 c3t10d0`

b. `# zpool create opmpool mirror c3t3d0 c3t4d0 c3t5d0 c3t6d0 mirror c3t7d0 c3t8d0 c3t9d0 c3t10d0`

c. `# zpool create opmpool mirror c3t3d0 c3t4d0 mirror c3t5d0 c3t6d0 mirror c3t7d0 c3t8d0 mirror c3t9d0 c3t10d0`

Answer: b

Quiz

You need to create two file systems, `rpts2012` and `qrtrpts`, in `hrpool`. The `qrtrpts` is a sub file system of `rpts2012`. Which command(s) do you use to create these file systems?

a. `# zfs create -o mountpoint=/rpts2012 hrpool/rpts2012/qrtrpts`

b. `# zfs create -o mountpoint=/rpts2012 hrpool/rpts2012`

   `# zfs create -o mountpoint=/qrtrpts hrpool/qrtrpts`

c. `# zfs create -o mountpoint=/rpts2012 hrpool/rpts2012`

   `# zfs create -o mountpoint=/qrtrpts hrpool/rpts2012/qrtrpts`

Answer: c

Practice 5-1 Overview: Configuring the ZFS Storage Hierarchy

This practice covers the following topics:

- Creating ZFS pools
- Creating ZFS file systems
- Populating the ZFS file systems with application data

This practice should take about 25 minutes to complete.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Hierarchy
- Configuring the ZFS Hierarchy for a Business Application
- Sharing ZFS File Systems in a Non-Global Zone
- Encrypting ZFS Data
- Migrating ZFS Data
- Upgrading ZFS Components

Sharing ZFS File Systems in a Non-Global Zone

This section covers the following topics:

- Creating a non-global zone
- Delegating a ZFS dataset to a non-global zone
- Adding a ZFS file system to a non-global zone

Fantastic Products LLC		
File systems	`custfs`	Make shareable in the non-global zone (`fpzone`)
	`mfrfs`	Add as a dataset to the non-global zone (`fpzone`)
	`bafs`	
	`hpfs`	

Assume that you have created all the file systems according to the data management plan and that you have loaded all the application data into the appropriate directories. According to the plan, your next task is to set up a non-global zone called `fpzone` for Fantastic Products LLC. You might recall from the lesson titled "Planning for Data Management" that all the manufacturer data for this application is stored in the `mfrfs` file system and the plan calls for this file system to be designated as shareable in a non-global zone called `fpzone`.

In this section, you are shown how to create the non-global zone and configure it so that it uses the same network interface as the global zone. You are then shown how to delegate a ZFS dataset to the non-global zone. In this scenario, you use the `mfrfs` file system as the dataset. You are then shown how to add the `custfs` file system to the non-global zone and make it shareable to any users of the application.

Creating a Non-Global Zone

1. Use `zfs create -o mountpoint=/zones rpool/zones` to create the ZFS zones file system in the root pool.
2. Use `zfs list rpool/zones` to verify creation of the `zones` file system.
3. Use `zonecfg -z` *`zonename`* to create the non-global zone. Verify, commit, and exit the zone.
4. Use `zonecfg -z` *`zonename`* `info` to verify the non-global configuration.
5. Use `zoneadm list -cv` to display the status of the newly created zone.
6. Use `pkg publisher` to verify that the IPS repository publisher is set correctly and that it is online.

ORACLE

The steps for creating a non-global zone and creating the ZFS zones file system in the zone are presented in the slide.

Note for step 1: The `zones` file system contains all the individual zones' file systems. Typically, this file system is called `zones`, as shown in the slide.

Note for step 3: As you configure the zone, be sure to set `ip-type` to `shared`.

Note for step 5: The zone should be in the configured state, and the network should be shared with the global zone.

Note for step 6: During the zone installation process that occurs in step 9 in the next slide, the zone installation packages are automatically downloaded from the specified IPS publisher. If the publisher is not set up correctly or is not online, the zone installation process cannot be completed.

Creating a Non-Global Zone

7. Use `sysconfig create-profile -o /`*`pathname`*`/`*`zonename`*`.xml` to create the zone configuration profile.
8. Use `more /`*`pathname`*`/`*`zonename`*`.xml` to review the system configuration file you just created.
9. Use `zoneadm -z` *`zonename`* `install –c /`*`pathname`*`/`*`zonename`*`.xml` to install the zone using the system configuration profile.
10. Use `zoneadm list -iv` to display the status of the newly installed zone.
11. Use `zoneadm -z` *`zonename`* `boot` to boot the newly installed zone.

The steps for creating a non-global zone are continued in the slide from the previous slide.

Note for step 7: The system configuration profile specifies the default locale and time zone, the zone's root password, a naming service to use, and other aspects of the application environment. Most of the information is supplied by selecting from a list of choices presented by the system configuration tool (`sysconfig`). Typically, you can use the default options unless your system configuration requires otherwise. After you have supplied the required information for the zone, the configuration file is created

Note for step 9: The installation process automatically creates a ZFS file system (dataset) for the zone root path when the zone is installed. If the file system cannot be created, the zone is not installed. The installation process also verifies the specified publisher and downloads the zone installation packages from IPS. This process normally takes about three to five minutes per zone.

Note for step 10: The zone should be in the installed state.

Creating a Non-Global Zone: Example

```
# zfs create -o mountpoint=/zones rpool/zones
# zfs list rpool/zones
NAME          USED   AVAIL  REFER  MOUNTPOINT
rpool/zones   377M   20.4G    32K  /zones
# zonecfg -z fpzone
fpzone: No such zone configured
Use 'create' to begin configuring a new zone.

zonecfg:fpzone> create -b
zonecfg:fpzone> set zonepath=/zones/fpzone
zonecfg:fpzone> set brand=solaris
zonecfg:fpzone> set autoboot=true
zonecfg:fpzone> set ip-type=shared
zonecfg:fpzone> add net
zonecfg:fpzone:net> set physical=net0
zonecfg:fpzone:net> set address=192.168.0.200
zonecfg:fpzone:net> end
zonecfg:fpzone> verify
zonecfg:fpzone> commit
zonecfg:fpzone> exit
```


In this example, we create the non-global zone called `fpzone` for the Fantastic Products LLC. We first create the `zones` ZFS file system in the root pool and verify that it has been created. In the next step, we create and configure the `fpzone` zone. Note that we set `ip-type` to `shared` so that the non-global zone and the global zone share the same network. We verify, commit, and exit the zone configuration.

Creating a Non-Global Zone: Example

```
# zonecfg -z fpzone info
<output omitted>
# zoneadm list -cv
  ID NAME       STATUS     PATH                 BRAND     IP
   0 global     running    /                    solaris   shared
   - fpzone     configured /zones/fpzone        solaris   shared
# pkg publisher
PUBLISHER       TYPE     STATUS   URI
solaris         origin   online   http://s11-server.mydomain.com/
# sysconfig create-profile -o /opt/ora/data/fpzone.xml
<output omitted>
# zoneadm -z fpzone install -c /opt/ora/data/fpzone.xml
<output omitted>
# zoneadm list -iv
  ID NAME       STATUS     PATH                 BRAND     IP
   0 global     running    /                    solaris   shared
   - fpzone     installed /zones/fpzone         solaris   shared
# zoneadm -z fpzone boot
# zoneadm list -iv
  ID NAME              STATUS     PATH           BRAND     IP
   0 global            running    /              solaris   shared
   1 fpzone            running    /zones/fpzone  solaris   shared
```


Our next step is to verify the zone configuration. After we are satisfied that we have configured the zone correctly, we run the `zoneadm list -cv` command to verify that the newly created zone is in the configured state; we find that it is. We then verify that the IPS repository publisher is set correctly and online. In the example, we are using a local repository publisher called `solaris`, which resides at the URI shown in the slide. As you can see, the `solaris` publisher is online. In the next step, we create the system configuration file.

Note: In the practice for this task, you are given the configuration information you need to complete the system configuration profile.

After we complete the system configuration profile, we install the zone using the system configuration profile. Recall that during the installation process, a ZFS file system is automatically created for the zone path and the zone installation packages are downloaded from the specified IPS repository. When the installation process completes, we again check the status of the zone. At this point, the zone should be in the installed state, and it is. We then boot the zone. Our final step is to verify that the zone is running, and it is.

Delegating a ZFS Dataset to a Non-Global Zone

1. Use the `zonecfg` command's `add dataset` subcommand to delegate a file system to the non-global zone.
2. Use `zoneadm -z` *`zonename`* `reboot` to restart the zone.
3. Use `zoneadm list -iv` to display the status of the zone.
4. Use `zlogin` *`zonename`* to log in to the zone.
5. Use `zfs list -r` to display the zone's ZFS components.
6. Use `exit` to exit the zone.

Now that we have created a non-global zone, we can delegate a dataset to the zone. ZFS allows datasets to be delegated to a non-global zone, which gives complete control over the dataset and all its children to the zone administrator. The zone administrator can create and destroy file systems or clones in that dataset, as well as modify properties of the datasets. The zone administrator cannot affect datasets that have not been added to the zone, including exceeding any top-level quotas set on the delegated dataset.

The steps for delegating a ZFS file system to a non-global zone are presented in the slide.

Note for step 1: As part of the dataset configuration, you use the `set name=poolname/filesystem_name` command to make the file system visible in the zone (see the example in the next slide). After you have specified the dataset, verify and commit the changes and then exit the zone.

Note for step 2: The zone is rebooted for the modifications to take effect.

Note for step 3: The zone should be in the `running` state.

Note for step 5: The new file system should be displayed.

Delegating a ZFS Dataset to a Non-Global Zone: Example

```
# zonecfg -z fpzone
zonecfg:fpzone> add dataset
zonecfg:fpzone:dataset> set name=fplpool/mfrfs
zonecfg:fpzone:dataset> end
zonecfg:fpzone> verify
zonecfg:fpzone> commit
zonecfg:fpzone> exit
# zoneadm -z fpzone reboot
<output omitted>
# zoneadm list -iv
  ID NAME             STATUS     PATH           BRAND    IP
   0 global           running    /              solaris  shared
   4 fpzone           running    /zones/fpzone  solaris  shared
# zlogin fpzone
<output omitted>
# zfs list -r
NAME                         USED  AVAIL  REFER  MOUNTPOINT
mfrfs                       38.6K  1.94G  38.6K  /mfrfs
rpool                        377M  20.4G    31K  /rpool
<output omitted>
# exit
logout
```


In this example, we delegate the ZFS file system called `mfrfs` as a dataset to the non-global zone called `fpzone` for the Fantastic Products LLC. We first add the dataset to the zone configuration and then specify the dataset's name by using the pool name (`fplpool`) and the file system name (`mfrfs`). After we have added the dataset, we verify, commit, and exit the zone configuration.

In the next step, we restart the zone so that the configuration changes we just made will take effect. We then check the status of the zone to ensure that it is up and running, and it is.

We then log in to the zone and verify that the `mfrfs` file system is listed as one of the ZFS components. Our final step is to exit the zone.

Adding a ZFS File System to a Non-Global Zone

1. Use the `zonecfg` command's `add fs` subcommand to add a file system to the non-global zone for purposes of sharing.
2. Use `zoneadm -z` *`zonename`* `reboot` to restart the zone.
3. Use `zoneadm list -iv` to display the status of the zone.
4. Use `zlogin` *`zonename`* to log in to the zone.
5. Use `zfs list -r` to display the zone's ZFS components.
6. Use `exit` to exit the zone.

You have been shown how to create a non-global zone and delegate a ZFS dataset to it. Now we look at how to add a file system to a non-global zone and make the file shareable. You can add a ZFS file system as a generic file system when the goal is solely to share space with the global zone. The zone administrator can create and destroy files in the file system. However, the file system cannot be remounted in a different location, nor can the zone administrator change properties on the file system. The global zone administrator is responsible for setting and controlling properties of the file system.

The steps for adding a file system and making it shareable in a non-global zone are presented in the slide.

Note for step 1: As part of the file system configuration, you need to specify certain values about how and where the file system should be mounted. For example, you use the `set dir=/local/filesystem_name` command to specify the mount point directory location of the file system (see the example in the next slide). You use the `set special=poolname/filesystem_name` command to specify the file system that is to be shared. You also use the `set type=zfs` command to specify the type of file system, which in our scenario is a ZFS file type.

After you have specified the file system you want to add, you verify and commit the changes and then exit the zone.

Note for step 3: The zone should be in the `running` state.

Note for step 5: The new file system should be displayed.

Adding a ZFS File System to a Non-Global Zone: Example

```
# zonecfg -z fpzone
zonecfg:fpzone> add fs
zonecfg:fpzone:fs> set dir=/local/customers
zonecfg:fpzone:fs> set special=fplpool/custfs
zonecfg:fpzone:fs> set type=zfs
zonecfg:fpzone:fs> end
zonecfg:fpzone> verify
zonecfg:fpzone> commit
zonecfg:fpzone> exit
# zoneadm -z fpzone reboot
<output omitted>
# zoneadm list -iv
  ID NAME             STATUS     PATH           BRAND    IP
   0 global           running    /              solaris  shared
   5 fpzone           running    /zones/fpzone  solaris  shared
# zlogin fpzone
<output omitted>
# cd /local/customers
root@fpzone:/local/customers# ls
america   americas  europe   sandy
root@fpzone:/local/customers# exit
logout
```


In this example for Fantastic Products LLC, we add a ZFS file system called `custfs` to the non-global zone called `fpzone` for the purposes of sharing the file system with users of the FPL application data. We first add the file system to the zone configuration, and then we specify that the `custfs` file system should be mounted as `/local/customers`. We then identify the file system that we want to share by using the `set special=fplpool/custfs` command, and we specify the type of file system by using `set type=zfs`. After we have added the file system, we verify, commit, and exit the zone configuration.

In the next step, we restart the zone so that the configuration changes we just made will take effect. We then check the status of the zone to ensure that it is up and running, and it is.

We then log in to the zone and verify that users of the FPL application can access the `/local/customers` directory and the files in it. From the output in the example, we see that they can access the directory and its files.

Our final step is to exit the `/local/customers` directory.

Quiz

Which three statements are correct about delegating ZFS datasets to a non-global zone?

a. The zone administrator has complete control over the dataset and all its children.
b. The zone administrator can create and destroy file systems or clones in a delegated dataset but cannot modify properties of the datasets.
c. The `zonecfg` command's `add dataset` subcommand is used to delegate a file system to a non-global zone.
d. The zone administrator cannot affect datasets that have not been added to the zone.

Answer: a, c, d

Quiz

Which three statements are correct about adding a ZFS file system to a non-global zone for the purposes of sharing only?

a. The zone administrator can create and destroy files in the file system.

b. The file system cannot be remounted in a different location.

c. The zone administrator can change properties on the file system.

d. The `zonecfg` command's `add fs` subcommand is used to add a file system to a non-global zone.

Answer: a, b, d

Practice 5-2 Overview: Sharing ZFS File Systems in a Non-Global Zone

This practice covers the following topics:

- Creating a non-global zone
- Delegating a ZFS dataset to a non-global zone
- Adding a ZFS file system to a non-global zone

This practice should take about 30 minutes to complete.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Hierarchy
- Configuring the ZFS Hierarchy for a Business Application
- Sharing ZFS File Systems in a Non-Global Zone
- Encrypting ZFS Data
- Migrating ZFS Data
- Upgrading ZFS Components

Encrypting ZFS Data

This section covers the following topics:

- Securing ZFS data with ZFS encryption
- Encrypting a ZFS file
- Encrypting a ZFS storage pool
- Encrypting a ZFS file system

Outstanding Property Management Company		Fantastic Products LLC	
Encryption requirements	File system: `investordata` in the `investorpool`	**Encryption requirements**	File: `/export/home/encrypted`
			Pool: `securepool`

You might recall from the lesson titled "Planning for Data Management" that there are several data encryption requirements. There is a data encryption requirement for the Outstanding Property Management Company to protect data about potential investors. To support this requirement, you need to create an encrypted file system called `investordata` in the `investorpool`.

For Fantastic Products LLC, there is a requirement to save the application passwords in a central place (`/export/home/encrypted` file in the root pool), and this file must be encrypted for security reasons. The Fantastic Products LLC application also maintains a confidential file system called `newproducts` in a pool called `securepool`. According to the data management plan, the `securepool` must be encrypted. Setting the encryption at the pool level ensures that all product-related data in the pool is encrypted automatically.

In this topic, you are introduced to ZFS encryption and shown how to encrypt a ZFS file, storage pool, and file system. The level of encryption (file, pool, or file system) that you use is determined by the specific needs of your business.

Securing ZFS Data with ZFS Encryption

- ZFS encryption is integrated with the ZFS command set.
- An encryption policy can be set when a ZFS file system is created, but the policy cannot be changed.
- The encryption policy on a new file system is enabled by setting the encryption property to on: `encryption=on`.
- ZFS encryption is inheritable to descendent file systems.
- ZFS encryption uses the Oracle Solaris Cryptographic Framework, which gives it automatic access to:
  - Available hardware encryption acceleration
  - Optimized encryption algorithms software

ORACLE

Encrypted dataset support has been added to ZFS to protect against theft of physical storage and man-in-the-middle attacks on the SAN, and to provide dataset-level secured deletion. Data is encrypted at the dataset level, allowing a mix of encrypted and unencrypted datasets in the same ZFS storage pool. Encrypted root pools are not supported at this time.

Encryption is the process whereby data is encoded for privacy and a key is needed by the data owner to access the encoded data. The benefits of using ZFS encryption are as follows:

- ZFS encryption is integrated with the ZFS command set.
- You can set an encryption policy when a ZFS file system is created, but the policy cannot be changed. The default encryption policy is to prompt for a passphrase, which must be a minimum of eight characters in length.
- You enable the encryption policy on a newly created file system by setting the encryption property to on.
- ZFS encryption is inheritable to descendent file systems. Key management can be delegated through ZFS delegated administration.
- ZFS encryption uses the Oracle Solaris Cryptographic Framework, which gives it automatic access to any available hardware acceleration or optimized software implementations of the encryption algorithms.

Oracle Solaris 11 Encryption Methods

- The default data encryption algorithm is `aes-128-ccm`.
- Use `encrypt -l` to display the available encryption methods.

```
# encrypt -l
Algorithm       Keysize:  Min   Max (bits)
------------------------------------------
aes                        128   256
arcfour                      8  2048
des                         64    64
3des                       128   192
```

- Use `cryptoadm list` to display the mechanisms that are available for each encryption method.

```
# cryptoadm list -m provider=aes
aes:
CKM_AES_ECB,CKM_AES_CBC,CKM_AES_CTR,CKM_AES_CCM,CKM_AES_GCM,CKM_AES_GMAC
,CKM_AES_CFB128
```


Data is encrypted using AES (Advanced Encryption Standard) with key lengths of 128, 192, and 256 in the CCM and GCM operation modes. The default encryption algorithm is `aes-128-ccm` when a file system's encryption value is `on`.

You can use the `encrypt -l` command to view the available encryption methods provided by the Solaris Cryptographic Framework, as shown in the first code example. To view the mechanisms available for each encryption method, use the `cryptoadm list` command. In the second code example, we display the mechanisms for the AES encryption method only.

Note: The `cryptoadm` utility is used to administer the cryptographic framework. This utility can be used to display cryptographic provider information for a system, configure the mechanism policy for each provider, and install or uninstall a cryptographic provider. For more information about this utility, see the `cryptoadm(1M)` man page.

Wrapping Keys

- A wrapping key is used to encrypt the actual data encryption keys.
- The `keysource` property is used to specify the format and location of the wrapping key.
- The `pktool` utility is used to generate keys for data encryption and decryption.

```
# pktool setpin
Enter token passphrase: changeme
Create new passphrase: oracle1
Re-enter new passphrase: oracle1
Passphrase changed.
# pktool genkey label=hraeskey keytype=aes keylen=256
Enter PIN for Sun Software PKCS#11 softtoken: oracle1
# pktool list objtype=key
Enter PIN for Sun Software PKCS#11 softtoken: oracle1
Found 1 symmetric keys.
Key #1 - AES:  hraeskey (256 bits)
```


A wrapping key is used to encrypt the actual data encryption keys. The wrapping key is passed from the `zfs` command to the kernel. Either a wrapping key is in a file (in raw or hex format) or it is derived from a passphrase.

The `keysource` property is used to specify the format and location of the wrapping key: `keysource=format,location`. The `keysource` property format and location options are as follows:

- Format options
  - `raw`: Raw key bytes
  - `hex`: Hexadecimal key string
  - `passphrase`: Character string that generates a key
- Location options
  - `prompt`: You are prompted for a key when the file system is created or mounted.
  - `file:///filename`: Key file location in a file system
  - `pkcs11`: URI describing the location of a key in a PKCS#11 token
  - `https://location`: Key file location on a secure server

If the keysource format is a passphrase, the wrapping key is derived from the passphrase. Otherwise, the keysource property value points to the actual wrapping key, as raw bytes or in hexadecimal format. For information about changing an encrypted ZFS file system's key, see "Changing an Encrypted ZFS File System's Keys" in the *Oracle Solaris Administration: ZFS File Systems* guide.

You can use the pktool command to manage the certificates and keys on multiple keystores, including PKCS#11 tokens (that is, the Solaris Cryptographic Framework). A *keystore* is a key database file that contains a combination of public and private keys. In this course, you learn how to use the pktool command to change the default passphrase and generate a new key as part of encrypting and decrypting a ZFS file. For more information about the uses of the pktool command, see the pktool (1) man page.

The example shows how to generate a new key by using the pktool. We first initialize and set the passphrase to a newly created token object store. For the Sun Software PKCS#11 softtoken keystore (default), the user must use the setpin command with the default passphrase changeme as the old passphrase to change the passphrase of the object store. Our new passphrase is oracle1. After we have established access to the token object store, we can generate keys. We then generate a new key called hraeskey by using the pktool genkey command. The label= subcommand specifies the key label or name of the key. The keytype= subcommand specifies the encryption method (aes, arcfour, des, 3des) to be used with the new key. The keylen= subcommand specifies the size or length of the key. In our example, the key size is 256 bits. We then use the pktool list command with the objtyp=key subcommand to verify creation of the key. We can now use this key to encrypt files. You are shown how to do this in the following slides.

Encrypting and Decrypting a ZFS File

Use `encrypt -a` *`algorithm`* `-i` *`input_file`* `-o` *`output_file`* to encrypt a ZFS file.

```
# encrypt -a aes -i /export/home/unencrypted \
-o /export/home/encrypted
```

Use `decrypt -a` *`algorithm input_file -o`* *`output_file`* to decrypt a ZFS file.

```
# decrypt -a aes -i /export/home/encrypted \
-o /export/home/decrypted
```

ORACLE

There are several approaches you can take to encrypt a ZFS file. You can use the ZFS `encryption` property, or you can use the `encrypt` command. If you use the ZFS `encryption` property approach, you cannot decrypt data after you have encrypted it. With the `encrypt` command approach, you can both encrypt and decrypt the data.

Note: You can use the `encrypt` command on single files, but the smallest unit of work for the ZFS `encryption` property is the file system. The `encrypt` command is not ZFS-specific and can be used on any file in any file system.

The syntax for encrypting a file is presented in the slide. In the example, we are encrypting a file called `unencrypted` by using an `aes` algorithm. The `-a` option specifies the name of the algorithm that is used to encrypt the file. The `-i` option specifies the input file. The `-o` option specifies the output file.

The syntax for decrypting a file is presented in the lower half of the slide. In the example, we are decrypting the file called `encrypted` by using an `aes` algorithm.

Encrypting and Decrypting a ZFS File: Example

```
# pktool genkey label=pwaeskey keytype=aes keylen=256
Enter PIN for Sun Software PKCS#11 softtoken: oracle1
# pktool list objtype=key
Enter PIN for Sun Software PKCS#11 softtoken: oracle1
Found 1 symmetric keys.
Key #1 - AES:  hraeskey (256 bits)
Key #2 - AES:  pwaeskey (256 bits)
# vi /export/home/unencrypted
…
FPL login password: APLe#984
# encrypt -a aes -K pwaeskey -i /export/home/unencrypted \
-o /export/home/encrypted
Enter PIN for Sun Software PKCS#11 softtoken: oracle1
# cat /export/home/encrypted
ï¿½ï¿½_Ztï¿½ï¿½<<@ï¿½ï¿½Ãï¿½6ï¿½ï¿½y^ï¿½.vkï¿½ï¿½ï¿½p
# decrypt -a aes -K pwaeskey -i /export/home/encrypted \
-o /export/home/decrypted
Enter PIN for Sun Software PKCS#11 softtoken: oracle1
# cat /export/home/decrypted
FPL login password: APLe#984
```


In this example, we create the `/export/home/encrypted` file in the root pool to store user passwords for the Fantastic Products LLC application and encrypt it according to the security requirement specified in the data management plan. The security requirement specifies that we must generate a 256-bit key of the `aes` type to replace the default encryption algorithm. Assume that we are using the same token object store that we initialized and set up earlier, which uses the `oracle1` PIN.

Our first step is to use the `pktool genkey` command to generate the new key called `pwaeskey` with the key length of 256 (as specified in the data management plan requirements). We verify creation of the new key by using the `pktool list objtype=key` command. The key is listed along with the first key we created earlier. Our next step is to create a file called `unencrypted` in the `/export/home` root pool file system. This file contains the login password for the Fantastic Products LLC application and is intended as a backup if someone forgets the password. We then encrypt the `unencrypted` file and verify that it is encrypted; verification indicates that it is encrypted. In the last two commands, we decrypt the encrypted file and verify that the file has been decrypted. The file has been decrypted.

Encrypting a ZFS Storage Pool

1. Use `zpool create -O encryption=on` *`poolname device`* to create an encrypted storage pool.
2. Use `zfs get encryption` *`poolname`* to verify that the `encryption` property is enabled for the pool.

```
# zpool create -O encryption=on securepool c3t2d0
Enter passphrase for 'securepool': oracle123
Enter again: oracle123
# zfs create securepool/newproducts
# zfs get encryption securepool/newproducts
NAME                    PROPERTY    VALUE         SOURCE
securepool/newproducts  encryption  on            inherited from securepool
# zfs get keysource securepool/newproducts
NAME                    PROPERTY   VALUE              SOURCE
securepool/newproducts  keysource  passphrase,prompt  inherited from securepool
```

ORACLE

Now let’s look at how to encrypt a ZFS storage pool. Keep in mind that setting the encryption at the pool level ensures that all the product-related data in the pool is encrypted automatically. The steps for encrypting a ZFS storage pool are presented in the slide.

Note for step 1: The `-O` option sets the given properties for the pool's top-level file system, which in this case is the `encryption` property. You are prompted to provide a passphrase for the new storage pool (see the example code in the slide).

In the example, according to our data management plan requirements, we create a ZFS storage pool called `securepool` for Fantastic Products LLC with the `encryption` property set to `on`. We provide a passphrase when prompted and confirm the passphrase. We then create the confidential file system called `newproducts` as specified in the data management plan. We then verify that the `newproducts` file system has inherited the `encryption` property setting from the pool; it has inherited the setting. To display information about the key for the new file system, we run the `zfs get keysource` command. As you can see, the `keysource` property for the `newproducts` file system has been inherited from the pool, just as the `encryption` property was.

Encrypting a ZFS File System

1. Use `zfs create -o encryption=on` *`poolname/filesystem`* to create an encrypted file system.
2. Use `zfs get encryption` *`poolname/filesystem`* to verify that the `encryption` property is enabled for the file system.

```
# pktool genkey keystore=file outkey=/investorkey keytype=aes keylen=256
# zfs create -o encryption=aes-256-gm -o \
keysource=raw,file:///investorkey investorpool/investordata
# zfs get encryption investorpool/investordata
NAME                        PROPERTY    VALUE        SOURCE
investorpool/investordata   encryption  aes-256-gcm  local
# zfs get keysource investorpool/investordata
NAME                        PROPERTY   VALUE                     SOURCE
investorpool/investordata   keysource  raw,file:///investorkey   local
```


You now know how to complete the encryption tasks for Fantastic Products LLC. You know how to encrypt a file and a storage pool. Now let's look at how you can address the requirements for the Outstanding Property Management Company by enabling encryption on a file system. These steps assume that you have already created the pool in which you are going to create the file system.

In the example, according to the data management plan requirements, we are creating the file system called `investordata` in an existing pool called `investorpool` and enabling encryption on the new file system. Because there is a business requirement to use a stronger key than is provided by the default encryption algorithm (`aes-128-ccm`), we generate a new key with a key length of 256 bits, which is the maximum key size that AES supports. When we create the file system, we specify that the file should be encrypted using the new key (`encryption=aes-256-gm`).

Note: If we did not have the requirement to increase the strength of the key, we could simply set the `encryption` property to `on`, as shown in the following example:

```
# zfs create -o encryption=on investorpool/investordata
```

We verify that encryption has been enabled and that the file system now points to the `aes-256-gm` mechanism that we specified in the previous command. We then run the `zfs get keysource` command to display information about the key for the new file system. As you can see, the `keysource` property value for the `investordata` file system is based on the keysource property format and location values that we specified when we created the file system: `keysource=raw,file:///investorkey`.

Quiz

What is the default encryption algorithm for ZFS?

a. `aes-128-ccm`

b. `aes-192-gcm`

c. `aes-256-ccm`

d. `aes-256-gcm`

Answer: a

Quiz

Which command do you use to generate an AES key named `rptaeskey` with a key size of 256?

a. `pktool genkey keytype=aes keysize=256`

b. `pktool genkey label=key keylen=256`

c. `pktool genkey label=rptaeskey objtype=aes keysize=256`

d. `pktool genkey label=rptaeskey keytype=aes keylen=256`

Answer: d

Practice 5-3 Overview: Encrypting ZFS Data

This practice covers the following topics:

- Encrypting and decrypting a ZFS file
- Encrypting a ZFS storage pool
- Encrypting a ZFS file system

This practice should take about 20 minutes to complete.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Hierarchy
- Configuring the ZFS Hierarchy for a Business Application
- Sharing ZFS File Systems in a Non-Global Zone
- Encrypting ZFS Data
- Migrating ZFS Data
- Upgrading ZFS Components

Migrating ZFS Data

This section covers the following topics:

- Migrating a ZFS storage pool
- Migrating file systems by using shadow migration

Fantastic Products LLC: Data Migration Requirements	
Storage pool	Migrate `mfrs` storage pool to remote system.
File systems	Migrate Asian manufacturer data (ZFS file system `mfrs/asia`) to `fplpool/asia` (remote).
	Migrate manufacturer UFS data (UFS file system `ufsdata`) to `fplpool/ufsdata` (remote).
	Migrate manufacturer UFS data (UFS file system `fplpool/ufsdata`) to `mfrs/ufsdata` (local).

According to the data management plan, your company has a requirement to migrate Fantastic Products LLC application data. This requirement includes migrating a storage pool and several file systems as shown in the table in the slide.

In this section, you learn how to migrate a storage pool between systems. You also learn how to use the shadow migration feature to migrate local and remote ZFS and UFS file systems to a target ZFS file system.

Migrating a ZFS Storage Pool

1. Use `zpool export` *`poolname`* to export a storage pool.
2. Use `zpool import` to determine which storage pools are available to import.
3. Use `zpool import` *`poolname`* to import a storage pool.

To move a storage pool between systems, the storage devices must be disconnected from the original system and reconnected to the destination system. You can accomplish this task by physically recabling the devices or by using multiported devices, such as the devices on a SAN. With ZFS, you can export the pool from one system and import it on the destination system, even if the systems are of different architectural endianness. The steps for exporting and importing a storage pool are presented in the slide.

Note for step 1: The `zpool export` command attempts to unmount any mounted file systems in the pool before continuing. If any of the file systems fail to unmount, you can forcefully unmount them by using the `-f` option. Storage pools should be explicitly exported to indicate that they are ready to be migrated. This operation flushes any unwritten data to disk, writes data to the disk indicating that the export was done, and removes all information about the pool from the system. If you do not explicitly export the pool, but if you instead remove the disks manually, you can still import the resulting pool on another system. However, you might lose the last few seconds of data transactions, and the pool will appear faulted on the original system because the devices are no longer present. By default, the destination system cannot import a pool that has not been explicitly exported. This condition is necessary to prevent you from accidentally importing an active pool that consists of network-attached storage that is still in use on another system.

Note: After you execute the `zpool export` command for a pool, the pool is no longer visible to the system. After you have removed the pool from the system (either through an explicit export or by forcefully removing the devices), you can attach the devices to the target system. ZFS can handle some situations in which only some of the devices are available, but a successful pool migration depends on the overall health of the devices. In addition, the devices do not necessarily have to be attached under the same device name. ZFS detects any moved or renamed devices and adjusts the configuration appropriately.

Note for step 2: Before you can actually import the storage pool, you must first run the `zpool import` command (with no options) to determine which storage pools are available to be imported. Each pool that is available to import is identified by a name as well as a unique numeric identifier. By default, the `zpool import` command only searches devices in the `/dev/dsk` directory. If devices exist in another directory or if you are using pools backed by files, you must use the `-d` option to search alternate directories.

Note for step 3: After you have determined that a pool is available for import, you can import it by specifying the name of the pool or its numeric identifier as an argument to the `zpool import` command. If multiple available pools have the same name, you must specify which pool to import by using the numeric identifier. If the pool name conflicts with an existing pool name, you can import the pool under a different name.

For more information about migrating ZFS storage pools, see "Migrating ZFS Storage Pools" in the *Oracle Solaris Administration: ZFS File Systems* guide.

Migrating a ZFS Storage Pool: Example

```
# zpool export mfrs
# zpool import

  pool: mfrs
    id: 14458949999061288397
 state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:

mfrs      ONLINE
  c3t4d0  ONLINE
# zpool import mfrs
pool: mfrs
 state: ONLINE
  scan: none requested
config:

NAME      STATE     READ WRITE CKSUM
mfrs      ONLINE       0     0     0
  c3t4d0  ONLINE       0     0     0

errors: No known data errors
```


In this example, we migrate the `mfrs` storage pool. We first export the pool and then determine if it is available for import; we see that it is. Note the ID number that is automatically assigned to the pool. The pool is in an `ONLINE` state and can be imported according to the information provided in the `action` field. The final step is to import the pool and, as you can see, the pool has been imported successfully. The pool is online with no known data errors.

Migrating File Systems by Using Shadow Migration

1. If you are migrating data from a remote NFS server, confirm that the name service information is accessible on both systems.
2. Use `pkg install shadow-migration` to install the shadow-migration package (if necessary).
3. Use `svcadm enable shadowd` to enable the `shadowd` service to assist with the migration process (if necessary).
4. Set the local or remote file system to be migrated to read-only.
   a. Use `zfs set readonly=on` *`path`* to set a local ZFS file system to read-only.
   b. Use `share -F nfs -o ro` *`path`* to share a remote file system as read-only.

In Oracle Solaris 11, ZFS supports shadow migration, which is a tool for migrating data from an existing file system to a new file system while simultaneously allowing access and modification of the new file system during the migration process. You can use the shadow property URI to identify the file system to be migrated in two ways:

- To migrate a local file system, use `shadow=file:///path`.
- To migrate an NFS file system, use `shadow=nfs://host:path`.

The steps for using shadow migration to migrate both local and remote file systems are presented in the slide.

Note for step 1: Migrating file system data over NFS can be slow, depending on your network bandwidth.

Note for step 3: If you do not enable the `shadowd` process, you will have to reset the `shadow` property to `none` when the migration process is complete. Otherwise, when the migration is complete, the `shadow` property is set to `none` automatically. The shadow migration daemon (`shadowd`) provides background worker threads to migrate data for a shadow migration. The `shadowd` service is managed by the service management facility (SMF).

Note: If the file system to be migrated is not set to read-only, the integrity of the data might be jeopardized.

Migrating File Systems by Using Shadow Migration

5. Create a new ZFS file system with the shadow property set to the file system to be migrated.
   a. Use `zfs create -o shadow=file:///`*`old_filesystem new_filesystem`* to migrate a local ZFS file system to a new ZFS file system.
   b. Use `zfs create -o shadow=nfs://`*`remote_filesystem new_ZFSfilesystem`* to migrate a remote file system to a new ZFS file system.
6. Use `shadowstat` to check the progress of the migration.
7. Use `zfs get -r shadow` *`filesystempath`* to verify that the migration process is completed. (The `shadow` property is set to `none`.)

ORACLE

The steps for using shadow migration to migrate file systems are continued in the slide.

Note for step 5: The target file system must be empty when you set the `shadow` property. If it is not empty, data migration will not begin. Keep these things in mind when migrating data:

- If you add or remove data from the file system to be migrated when the migration is in progress, those changes might not be migrated.
- You cannot change the mount point of the shadow file system when the migration is in progress.
- If you restart the system during a migration, the migration continues after the system is booted.
- Access to directory content or file content that is not completely migrated is blocked until *all* content is migrated.

Note for step 6: The `shadowstat` utility reports statistics about in-progress shadow migrations, such as the number of bytes transferred to the shadow file system, estimated bytes left to be transferred, errors, and elapsed time. If you specify a time count, updates occur only as specified by the time count, after which the program exits. If you do not specify a time count, updated values are presented every few seconds until the program is terminated.

Migrating a Remote ZFS File System: Example

```
root@server1:~# dfshares server2
RESOURCE                              SERVER      ACCESS    TRANSPORT
server2:/mfrs/asia                    server2     -         -
root@server1:~# pkg install shadow-migration
Refreshing catalog
<output omitted>
Creating Plan \

           Packages to install:  1
       Create boot environment: No
Create backup boot environment: No
            Services to change:  1
<output omitted>
root@server1:~# svcadm enable shadowd

root@server2:~# share -F nfs -o ro /mfrs/asia

root@server1:~# zfs create -o shadow=nfs://server2/mfrs/asia fplpool/asia
root@server1:~# shadowstat
                                        EST
                                BYTES   BYTES           ELAPSED
DATASET                         XFRD    LEFT    ERRORS  TIME
fplpool/asia                   45.5M   2.75M   -       00:02:31
<output omitted>
No migrations in progress
root@server1:~# zfs get -r shadow fplpool/asia
NAME          PROPERTY  VALUE   SOURCE
fplpool/asia  shadow    none    -
```


In this example, we create and then migrate a remote ZFS file system called `mfrs/asia` that resides on `server2` to a target ZFS file system called `fplpool/asia` that resides on `server1`.

We first ensure that the `mfrs/asia` file system is being shared (and it is). We install the shadow-migration package and enable the `shadowd` service. We then set the remote file system to be migrated (`/mfrs/asia`) to read-only.

Our next step is to create the new ZFS file system (`fplpool/asia`) with the shadow property set to the file system to be migrated (`/mfrs/asia`). The migration process begins.

We use the `shadowstat` utility to check the progress of the migration. To verify that the migration process is complete, we run the `zfs get -r shadow` command. The migration process is complete because the value status of the `shadow` property is set to `none`.

If the migration process is not complete, the file system being migrated appears in the `VALUE` field.

Sharing ZFS File Systems with SMB

Use `zfs set share` to create an SMB share of a ZFS file system.

```
# zfs create rpool/fs1
# zfs set share=name=fs1,path=/rpool/fs1,prot=smb rpool/fs1
name=fs1,path=/rpool/fs1,prot=smb
```

Use `zfs set sharesmb=on` to publish the share.

```
# zfs set sharesmb=on rpool/fs1
```

Use `zfs get sharesmb` *`filesystem`* to display the ZFS share information.

```
# zfs get sharesmb rpool/fs1
```


As you have already seen, you can share a ZFS file system over NFS. You can also share a ZFS file system over Server Message Block (SMB) and Common Internet File System (CIFS) protocols. NFS and SMB are both ZFS share properties that are used for sharing file systems across homogeneous and heterogeneous platforms. Both are vendor-independent protocols.

Note: The terms SMB and CIFS can be considered interchangeable. In this lesson, we use the term SMB.

In Oracle Solaris 11, you can create a ZFS file system share and define the SMB share properties by using the `zfs share` command. You use the `zfs set share` command to create an SMB share of a ZFS file system, and you publish the share by setting the `sharesmb` property to `on`, as shown in the examples in the slide. The share is stored in a file in the `.zfs/share` directory with the name of the share.

To display the ZFS share information, you use the `zfs get sharesmb` *`filesystem`* command.

Sharing ZFS File Systems with SMB

Use `zfs set sharesmb=off` to unshare a ZFS file system.

```
# zfs set sharesmb=off rpool/fs1
```

Use `zfs set -c share=name=name` *`filesystem`* to remove the ZFS share.

```
# zfs get share
NAME          PROPERTY   VALUE   SOURCE
rpool/ds      share      name=ds,path=/ds,prot=smb   local
# zfs set -c share=name=ds rpool/ds
share 'ds' was removed.
```


To unshare a ZFS file system, you set the `sharesmb` property to `off`.

To remove a ZFS share, you use the `zfs set -c` command. Note that you need to identify the *`share-name`* `name` with this command, as shown in the second example. In the example, we use the `zfs get share` command to display the share information, including the *`share-name`*. We then use the *`share-name`* in the removal of the share in the second command.

For more information about sharing ZFS file systems with SMB and the optional SMB properties, see the *Oracle Solaris Administration: ZFS File Systems* guide and the `share_smb`(1m) man page.

Behavior of `sharesmb` Property

- When the `sharesmb` property is set to on, all defined shares for the file system are published.
- When the `sharesmb` property is set to off:
  - All published shares for the file system are unpublished
  - No shares are removed
- When the `zfs unshare` command is issued:
  - All published shares for the file system are unpublished
  - No shares are removed

Let's look at how the `sharesmb` property settings affect shares that have been defined.

When you set the `sharesmb` property to `on`, all defined shares for the file system (and all descendent file systems inheriting the property) are published for the SMB protocol. All defined shares are also published when you issue the `zfs share` command. If no shares are defined, the file system is not shared. If shares are defined for the file system, then only those shares are published. The mount point of the file system is shared only if a share exists that explicitly shares it.

When you set the `sharesmb` property to `off`, all published shares for the file system (and all descendent file systems inheriting the property) are unpublished for the SMB protocol. These shares remain unshared until you set the `sharesmb` property back to `on`. No defined shares are removed when the property is set to `off`, and they are re-shared the next time you set the `sharesmb` property back to `on`.

When you issue the `zfs unshare` command, all published shares for the file system are unpublished. These shares remain unshared until you issue the `zfs share` command for the file system. No defined shares are removed when you issue the `zfs unshare` command, and they are re-shared the next time you issue the `zfs share` command.

Quiz

What is the proper sequence of commands to migrate a ZFS storage pool?

a. `zpool export` *pool* `-> zpool import` *pool*

b. `zpool export` *pool*`, zpool import, zpool import` *pool*

c. `zpool import, zpool import` *pool*`, zpool export` *pool*

d. `zpool export` *pool*`, zpool export, zpool import` *pool*

Answer: b

Quiz

Which two statements about ZFS shadow migration are true?

a. The target file system must be empty when you set the shadow property.
b. If you add or remove data from the file system to be migrated when the migration is in progress, those changes might not be migrated.
c. You can change the mount point of the shadow file system when the migration is in progress.
d. If you restart the system during a migration, the migration process is discontinued.

Answer: a, b

Practice 5-4 Overview: Migrating ZFS Data

This practice covers migrating the following:

- ZFS storage pools
- ZFS data to a remote target
- UFS data to a remote target
- UFS data to a local target

This practice should take about 30 minutes to complete.

Lesson Agenda

- Implementing the Plan to Manage the ZFS Hierarchy
- Configuring the ZFS Hierarchy for a Business Application
- Sharing ZFS File Systems in a Non-Global Zone
- Encrypting ZFS Data
- Migrating ZFS Data
- Upgrading ZFS Components

Upgrading ZFS Components

This section covers the following topics:

- Upgrading ZFS storage pools
- Upgrading ZFS file systems

Outstanding Property Management Company		Fantastic Products LLC	
Upgrade requirements	Pools: Upgrade as required to most current ZFS release.	**Upgrade requirements**	Pools: Upgrade as required to most current ZFS release.
	File systems: Upgrade as required to most current ZFS release.		File systems: Upgrade as required to most current ZFS release.

ORACLE

According to the data management plan, your company has a requirement to ensure that its Oracle Solaris systems are running the most current release of ZFS to take advantage of the latest pool and file system features. In this section, you learn how to upgrade ZFS storage pools and file systems.

Upgrading ZFS Storage Pools

Use `zpool upgrade` to:

- Identify a pool's on-disk version: `zpool upgrade`
- Display ZFS pool versions supported by the current software: `zpool upgrade -v`
- Upgrade all pools to the latest on-disk version: `zpool upgrade -a`
- Upgrade a specified pool to the latest on-disk version: `zpool upgrade -a` *`pool`*
- Upgrade a specified pool to a specified on-disk version: `zpool upgrade -V` *`version`* `-a` *`pool`*

If you have ZFS storage pools from a previous Solaris release, you can use the `zpool upgrade` command to upgrade your ZFS storage pools. The ways in which you can use the command are presented in the slide.

You do not have to upgrade your pools. However, if you continue to use older pool versions, some features might not be available. If you do upgrade a pool, you will not be able to access a pool of a later version on a system that runs an earlier software version.

Note: You cannot downgrade an on-disk format.

When you run the `zpool upgrade -v` command, the current ZFS pool versions and all previous supported versions are displayed, along with an explanation of the features provided with each version.

When you upgrade a pool to a specified version, the version you specify must be higher than the pool's current version. If you do not specify a version, the pool is upgraded to the latest on-disk version.

Upgrading ZFS File Systems

Use `zfs upgrade` to:

- Identify a file system version: `zfs upgrade`
- Display ZFS file system versions supported by the current software: `zfs upgrade –v`
- Upgrade all file systems to the latest on-disk version: `zfs upgrade -a`
- Upgrade a specified file system to the latest on-disk version: `zfs upgrade -r –a` *`filesystem`*
- Upgrade a specified file system to a specified on-disk version: `zfs upgrade -r -V` *`version`* `–a` *`filesystem`*

If you have ZFS file systems from a previous Solaris release, you can use the `zfs upgrade` command to upgrade your file systems. The ways in which you can use the command are presented in the slide.

You do not have to upgrade your file systems. However, if you continue to use older file system versions, some features might not be available. If you do upgrade a file system, you will not be able to access a file system of a later version on a system that runs an earlier software version.

When you run the `zfs upgrade -v` command, the current ZFS file system versions and all previous supported versions are displayed, along with an explanation of the features provided with each version.

When you upgrade a file system to a specified version, the version you specify must be higher than the file system's current version. If you do not specify a version, the file system is upgraded to the latest on-disk version.

Note: In general, the file system version is independent of the pool version. However, in some cases, the file system version and the pool version are interrelated and the pool version must be upgraded before the file system version can be upgraded.

Quiz

Which command do you use to upgrade a single ZFS file system to the most current release of ZFS?

a. `zfs upgrade`

b. `zfs upgrade –a`

c. `zfs upgrade` *`filesystem`*

d. `zfs upgrade -r –a` *`filesystem`*

Answer: d

Practice 5-5 Overview: Upgrading ZFS Components

This practice covers upgrading the following:

- ZFS storage pools
- ZFS file systems

This practice should take about 5 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan to configure and manage the ZFS hierarchy
- Configure ZFS pools for the business application data
- Create ZFS file systems for the business application data
- Share ZFS file systems in a non-global zone
- Encrypt ZFS data
- Migrate ZFS data
- Upgrade ZFS components

6

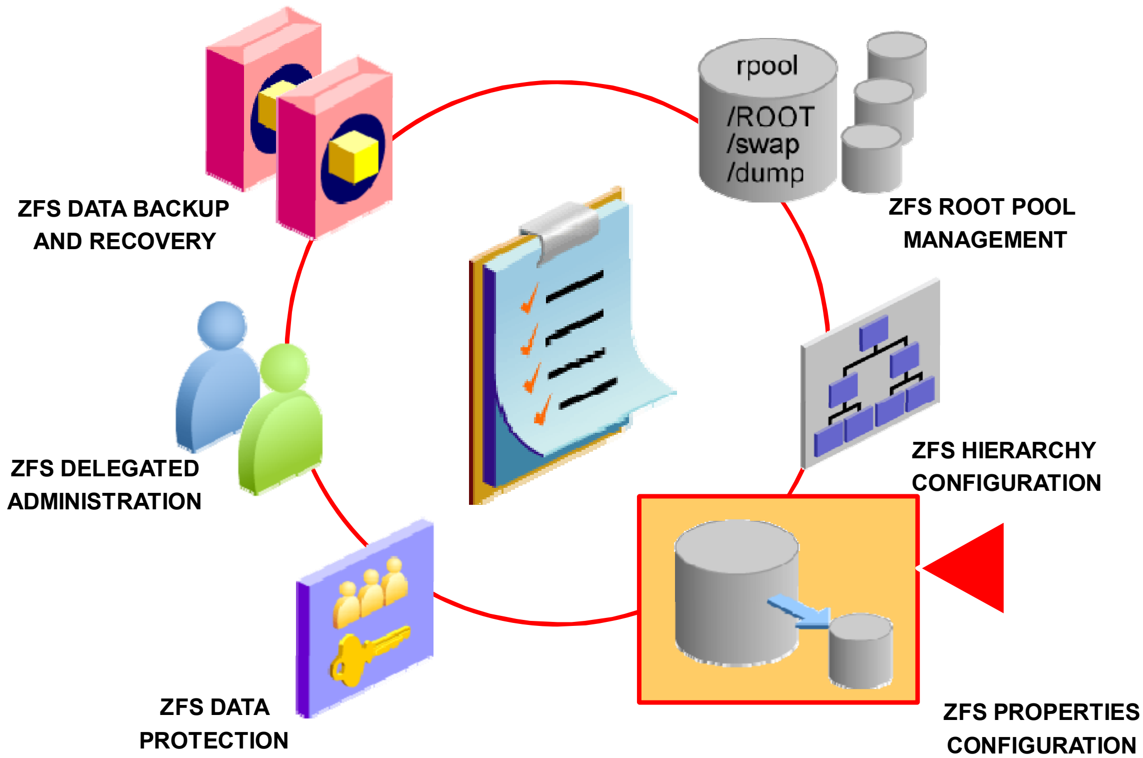
Configuring ZFS Properties

Objectives

After completing this lesson, you should be able to:

- Implement a plan to configure ZFS properties
- Configure ZFS storage pool properties
- Configure ZFS file system properties
- Manage ZFS properties in a non-global zone

Planning Workflow: Orientation



You have completed the ZFS hierarchy management tasks according to the data management plan. You are now ready to proceed to the next set of tasks, which require you to configure the ZFS storage pool and file system properties to support the requirements identified in the plan.

In this lesson, you are shown how to configure ZFS properties that support the data backup, storage, sharing, and recovery needs and requirements of the business. You are also shown how to manage ZFS properties in a non-global zone.

Lesson Agenda

- Implementing the Plan to Configure ZFS Properties
- Configuring ZFS Storage Pool Properties
- Configuring ZFS File System Properties
- Managing ZFS Properties in a Non-Global Zone

Implementing the Plan to Configure ZFS Properties

Outstanding Property Management Company		Fantastic Products LLC	
ZFS properties requirements		**ZFS properties requirements**	
Pool	Snapshot display Storage autoexpand Device autoreplace	**Pool**	Boot file system (`bootfs`)
File systems	Storage quota User storage quota Group storage quota Storage reservation	**File systems**	Compression File change ownership
		Non-global zone	Zoned Checksum

In the lesson titled "Planning for Data Management," you were presented with the requirements for configuring the ZFS pools and file system properties for the Outstanding Property Management Company and Fantastic Products LLC business application data, as listed in the table in the slide. It's now time to implement the specifics of that plan. In the slides that follow, you learn about each of the properties and how to configure them to meet the needs of the business.

Lesson Agenda

- Implementing the Plan to Configure ZFS Properties
- Configuring ZFS Storage Pool Properties
- Configuring ZFS File System Properties
- Managing ZFS Properties in a Non-Global Zone

Configuring ZFS Storage Pool Properties

This section covers setting the following:

- `listsnapshots` property
- `autoexpand` property
- `autoreplace` property
- `bootfs` property

Reviewing the `listsnapshots` Property Requirements

Outstanding Property Management Company	
ZFS properties requirements	
Pool	Enable snapshot display property on `/investorpool`. Create snapshot for the `/investorpool/investordata` file.

The Outstanding Property Management Company has a requirement to have a snapshot made of its investors' data. The data is located in the `/investorpool/investordata` file. The OPM investor committee reviews the list of investors monthly, so it is important that this data be backed up with a snapshot. In this section, you learn about the snapshot display property and how to set it.

`listsnapshots` Property: Overview

- By default, snapshots are not displayed in the `zfs list` output.
- To display snapshot information, you can either:
  - Enable the `listsnapshots` pool property

  or
  - Use the `zfs list -t snapshot` command

By default, snapshots are not displayed in the `zfs list` output. To display snapshot information, you must enable the `listsnapshots` pool property or use the `zfs list -t snapshot` command. You learn more about the `zfs list -t snapshot` command in the lesson on backing up ZFS data.

Note: If you disable the `listsnapshots` pool property, you must use the `zfs list -t snapshot` command to display snapshot information. You can list the snapshots that were created for a particular file system by using the `zfs list` command with the `-r` and `-t snapshot` options, followed by the file system name. In the example that follows, the snapshots created for the file system `hrpool/home` are listed. This information is displayed by using the name and creation properties.

```
# zfs list -r -t snapshot -o name,creation hrpool/home
NAME                                    CREATION
hrpool/home/qarpt@tuesday               Tue May 15 11:03 2011
hrpool/home/qarpt@wednesday             Wed May 16 11:03 2011
hrpool/home/qarpt@thursday              Thu May 17 11:03 2011
hrpool/home/bonus@now                   Fri May 18 11:04 2011
```

Note: You can access the snapshots of file systems in the `.zfs/snapshot` directory in the root of the file system. For example, if `hrpool/home/qarpt` is mounted on `/home/qarpt`, the `hrpool/home/qarpt@thursday` snapshot data is accessible in the `/home/qarpt/.zfs/snapshot/thursday` directory, as shown in the following example:

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

Setting the `listsnapshots` Property

1. Use `zpool get listsnapshots` *`poolname`* to determine the current setting of the snapshot display property for a pool.
2. Use `zpool set listsnapshots=on` *`poolname`* to enable the snapshot display property for a pool.
3. Use `zpool get listsnapshots` *`poolname`* to view the setting of the snapshot display property for a pool.
4. Use `zfs list -r /poolname` to display the snapshot entries for a pool.

The steps for setting the `listsnapshots` property are listed in the slide.

Note: To disable the listsnapshots pool property, change the property setting to `off` by using the `zpool set listsnapshots=off` *`poolname`* command.

Setting the `listsnapshots` Property: Example

```
# zfs snapshot investorpool/investordata@may2012
# zfs list -r /investorpool
NAME                         USED  AVAIL  REFER  MOUNTPOINT
investorpool                 143K   976M    32K  /investorpool
investorpool/investordata   34.5K   976M  34.5K  /investorpool/investordata
# zpool get listsnapshots investorpool
NAME          PROPERTY       VALUE      SOURCE
investorpool  listsnapshots  off        default
# zpool set listsnapshots=on investorpool
# zpool get listsnapshots investorpool
NAME          PROPERTY       VALUE      SOURCE
investorpool  listsnapshots  on        local
# zfs list -r /investorpool
NAME                         USED  AVAIL  REFER  MOUNTPOINT
investorpool                 146K   976M    32K  /investorpool
investorpool/investordata   34.5K   976M  34.5K  /investorpool/investordata
investorpool/investordata@may2012      0      -   34.5K  -
```


In this example, according to the data management plan requirement, we create a snapshot of the `/investorpool/investordata` file system for the Outstanding Property Management Company called `/investorpool/investordata@may2012`.

We then attempt to display the newly created snapshot by using the `zfs list -r /investorpool` command. The snapshot is not listed.

We then display the pool's `listsnapshots` property, and we see that the property is set to `off`. This explains why the snapshot is not displayed when we run the `zfs list -r` command.

We then set the `listsnapshots` property to `on`, verify the property setting, and rerun the `zfs list -r /investorpool` command. Now the `/investorpool/investordata@may2012` snapshot is displayed.

Reviewing the `autoexpand` Property Requirements

Outstanding Property Management Company	
ZFS properties requirements	
Pool	Increase size of `investorpool` from 1 GB to 10 GB. Enable storage `autoexpand` property on `investorpool`.

The Outstanding Property Management Company has another pool property requirement. Recall from the lesson titled "Planning for Data Management" that your business analyst expressed concern because the `investorpool` pool needed more storage capacity as more investor information was added.

To address this concern, the business has decided to replace the current 1 GB disk with a new 10 GB disk and to enable the pool property called `autoexpand` on the `investorpool` pool so that the pool can automatically expand to the full available storage capacity.

In this section, you learn about the `autoexpand` property and about how to set it.

`autoexpand` Property: Overview

- This property controls automatic pool expansion when the underlying device is grown
- By default, the `autoexpand` pool property is disabled.

The `autoexpand` pool property controls automatic pool expansion when the underlying device is grown. When you set the property to `on`, the pool resizes automatically according to the size of the expanded device. If the device is part of a mirrored storage pool or RAID-Z storage pool, all devices in that mirror or RAID-Z group must be expanded before the new space is made available to the pool. To avoid accidental expansions, this property is set to `off` by default.

Setting the `autoexpand` Property

1. Use `zpool get autoexpand` *`poolname`* to determine the current setting of the storage autoexpand property for a pool.
2. Use `zpool set autoexpand=on` *`poolname`* to enable the storage autoexpand property for a pool.
3. Use `zpool get autoexpand` *`poolname`* to view the setting of the storage autoexpand property for a pool.
4. Use `zpool list` *`poolname`* to verify that the pool size has expanded to the full capacity of the disk.

The steps for setting the `autoexpand` property are listed in the slide.

Note: To disable the `autoexpand` property, change the property setting to `off` by using the `zpool set autoexpand=off` *`poolname`* command.

Setting the `autoexpand` Property: Example

```
# zpool replace investorpool c3t4d0 c3t10d0
# zpool list investorpool
NAME           SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
investorpool  1008M   231K  1008M   0%  1.00x  ONLINE  -
# zpool get autoexpand investorpool
NAME          PROPERTY    VALUE   SOURCE
investorpool  autoexpand  off     default
# zpool set autoexpand=on investorpool
# zpool get autoexpand investorpool
NAME          PROPERTY    VALUE   SOURCE
investorpool  autoexpand  on    default
# zpool list investorpool
NAME           SIZE  ALLOC   FREE  CAP  DEDUP  HEALTH  ALTROOT
investorpool  9.98G   236K  9.98G   0%  1.00x  ONLINE  -
```


In this example, according to the data management plan requirement, we enable the `autoexpand` property on the `investorpool` pool for the Outstanding Property Management Company.

Let's assume that we have already replaced the older 1 GB disk with a higher capacity 10 GB disk according to the planning requirement to expand the pool to be able to add more investor information. Our first step is to replace the current disk, `c3t4d0`, with the new disk, `c3t10d0`. Using the `zpool list` command, we check the size of the pool, and it is 1008 MB, which is about 1 GB. Recall that this is the size we specified when we first created the `investorpool` pool.

We then check the current setting for the `autoexpand` property and see that it is currently set to `off`. To enable the property for the pool, we use the `zpool get autoexpand=on` command, and then we verify that the property is now set to `on`. Our final step is to verify that the pool has now expanded to the full capacity of 10 GB; we see that it has.

Setting the `autoreplace` Property

Outstanding Property Management Company	
ZFS properties requirements	
Pool	Enable device `autoreplace` property on `opmpool`.

The Outstanding Property Management Company has yet another pool property requirement: to have the `autoreplace` property enabled for the `opmpool` pool. Recall from the planning lesson that, because the application is relatively new, the business analyst expressed concern about disk failure. To address this concern, you enable the `autoreplace` property.

`autoreplace` Property: Overview

- This property controls automatic device replacement when a device fails.
- By default, the `autoreplace` pool property is disabled.

The `autoreplace` pool property controls automatic device replacement when a device fails.

If this property is set to `off`, you (the system administrator) must initiate the device replacement by using the `zpool replace` command.

If the `autoreplace` pool property is set to `on`, you can physically replace the failed device with a new device in the same location, and ZFS will automatically format the new device and places it in the pool.

The `autoreplace pool` property is set to `off` by default.

Setting the `autoreplace` Property

1. Use `zpool get autoreplace` *`poolname`* to determine the current setting of the `autoreplace` property for a pool.
2. Use `zpool set autoreplace=on` *`poolname`* to enable the `autoreplace` property for a pool.
3. Use `zpool get autoreplace` *`poolname`* to view the setting of the `autoreplace` property for a pool.

The steps for setting the `autoreplace` property are listed in the slide.

Note: To disable the `autoreplace` property, change the property setting to `off` by using the `zpool set autoreplace=off` *`poolname`* command.

Setting the `autoreplace` Property: Example

```
# zpool get autoreplace opmpool
NAME      PROPERTY     VALUE    SOURCE
opmpool   autoreplace  off      default
# zpool set autoreplace=on opmpool
NAME      PROPERTY     VALUE    SOURCE
opmpool   autoreplace  on       local
# zpool get autoreplace opmpool
NAME      PROPERTY     VALUE    SOURCE
opmpool   autoreplace  on      default
# zpool offline opmpool c3t6d0
# zpool status opmpool
<output omitted>
        NAME            STATE      READ WRITE CKSUM
        opmpool         ONLINE        0     0     0
          mirror-0      ONLINE        0     0     0
            c3t3d0      ONLINE        0     0     0
            c3t4d0      ONLINE        0     0     0
          mirror-1      ONLINE        0     0     0
            c3t5d0      ONLINE        0     0     0
            c3t20d0     ONLINE        0     0     0

errors: No known data errors
```


In this example, according to the data management plan requirement, we enable the `autoreplace` property on the `opmpool` pool for the Outstanding Property Management Company. We do this so that, if one of the disks fails, we will be able to replace the disk and have ZFS automatically format the new disk and place it in the pool.

We first check the current setting for the `autoreplace` property for the `opmpool` pool and see that the property is currently set to `off`. To enable the property, we use the `zpool set autoreplace=on` command and then we verify that the property is now set to `on`.

To ensure that the property is working as expected, we simulate a disk failure situation by taking disk `c3t6d0` in `mirror-1` offline, which causes `opmpool` to be in a DEGRADED state (not shown in the slide). We replace the “failed” disk with a new disk, `c3t20d0`, and place the new disk in the same physical location as the “failed” disk had been.

Note: The new disk address might differ based on how ZFS detects the disks and their slots.

Our final step is to verify that ZFS has automatically replaced the disk and that both the disk and the pool are once again online. We see that they are.

Reviewing the `bootfs` Property Requirements

Fantastic Products LLC	
ZFS properties requirements	
Pool	Create and configure a new disk for a second root pool called `fplrpool`. Set the boot file system property (`bootfs`) to the newly created root pool.

Now that you know how to address all the Outstanding Property Management Company pool property requirements, let's look at what you need to do to satisfy the pool property requirements for the Fantastic Products application. According to the data management plan, you must create and configure a new disk for a second root pool called `fplrpool`. This unique environment will be used solely for project testing. You then verify that the boot file system property (`bootfs`) specifies `fplrpool` as the primary boot disk.

`bootfs` Property: Overview

- This property identifies the bootable file system for the root pool.
- The `bootfs` pool property is typically set by the installation programs.

The boot file system property, `bootfs`, identifies the bootable file system for the root pool. Unlike the other pool properties, for which the system administrator controls the setting of the property, this property is typically set by the installation programs.

Setting the `bootfs` Property

Use `zpool get bootfs` *`poolname`* to view the current boot file system property setting.

```
# zpool get bootfs rpool
NAME    PROPERTY   VALUE                SOURCE
rpool   bootfs     rpool/ROOT/solaris   local
```

```
# zpool get bootfs fplrpool
NAME       PROPERTY   VALUE    SOURCE
fplrpool   bootfs     -        default
```


Although it is possible to set the `bootfs` property by using the `zpool set bootfs` command, it is better to let the system set the property as part of the installation process or as part of the boot environment activation and restart process. You can use the `zpool get bootfs` *`poolname`* command to view the current property setting.

In the first example in the slide, we display the `bootfs` property setting for the root pool. Note that the `VALUE` field is populated with the boot file system for the root pool: `rpool/ROOT/solaris`.

In the second example, we display the `bootfs` property setting for an alternate root pool we have created called `fplrpool`. Note that the `VALUE` field is not populated. This setting indicates that the boot file system for the alternate root pool is not the active boot environment.

Setting the `bootfs` Property: Example

```
# zpool create -f fplrpool c3t9d0s0
# zpool status fplrpool
  pool: fplrpool
 state: ONLINE
  scan: none requested
config:

NAME         STATE     READ  WRITE  CKSUM
fplrpool     ONLINE       0      0      0
  c3t9d0s0   ONLINE       0      0      0

errors: No known data errors
# cd /fplrpool
# cp -r /rpool/boot .
/fplrpool# ls
boot
/fplrpool# cd boot
/fplrpool/boot# ls
grub     solaris
# cd grub
/fplrpool/boot/grub# ls
menu.lst
```


Now let's return to our planning requirements. We need to set up the alternate root pool called `fplrpool` for the Fantastic Products LLC and make it the primary boot pool. Assume we have already selected and formatted the disk we are going to use for this root pool: disk `c3t9d0s0`. Our next step is to create the root pool and then verify that it is online; we see that it is.

We then copy the boot directory from the existing root pool into the `/fplrpool` file system. When you created the alternate root pool `fplrpool`, the boot information was not created. Therefore, it needs to be copied from the main root pool.

We change directories to the `/fplrpool` file system to verify that the boot directory is present, and it is. We then change directories to the boot directory to see what files are listed there. We see two files: `grub` and `solaris`.

We then go a level deeper into the `grub` directory, where we see the `menu.lst` file listed. Recall from our lesson on managing the ZFS root pool that the `menu.lst` file contains a `bootfs` entry that identifies the file system selected for booting the pool.

Setting the `bootfs` Property: Example

```
# more menu.lst
# default menu entry to boot
default 0
<output omitted>
title Oracle Solaris 11 11/11
bootfs rpool/ROOT/solaris
kernel$ /platform/i86pc/kernel/amd64/unix -B $ZFS-
    BOOTFS,console=graphics
module$ /platform/i86pc/amd64/boot_archive
# beadm create -p fplrpool fplsolaris
# beadm list
BE          Active Mountpoint Space Policy Created
--          ------ ---------- ----- ------ -------
fplsolaris -       -          4.21G static 2012-04-20 12:55
solaris    NR      /          4.95G static 2012-03-08 07:00
# beadm activate fplsolaris
[On an x86 system, select an alternate boot device from the BIOS. On a
SPARC system, use the boot -Z command.]

# zpool get bootfs fplrpool
NAME      PROPERTY  VALUE                   SOURCE
fplrpool  bootfs    fplrpool/ROOT/fplsolaris local
```


In this slide, we continue the example from the previous slide.

Our next step is to display the contents of the `menu.lst` file. We can see that the `bootfs` entry is still pointing to the `rpool/ROOT/solaris` file system in the main root pool.

To boot from the alternate root pool, we first create a boot environment (BE) called `fplsolaris` in the root pool and then verify that it has been created. We then activate the new BE. For the system to recognize the newly activated BE, we must first tell the system to boot from the new disk. On an x86-based system, this is done by selecting an alternate boot device from the BIOS; on a SPARC-based system, use the `boot -Z` command. After the system reboots, the new BE is activated.

As you learned in the lesson on managing the ZFS root pool, the `menu.lst` file now contains a `bootfs` entry for this boot file system. When we check the `bootfs` property for the pool, we see that `fplrpool/ROOT/fplsolaris` is identified as the bootable file system. This file system will be used for the next boot.

Quiz

Which three pool properties are disabled by default?

a. `autoexpand`
b. `autoreplace`
c. `bootfs`
d. `listsnapshots`

Answer: a, b, d

Practice 6-1 Overview: Configuring ZFS Pool Properties

This practice covers working with the following properties:

- `listsnapshots`
- `autoexpand`
- `autoreplace`
- `bootfs`

This practice should take about 90 minutes to complete.

Lesson Agenda

- Implementing the Plan to Configure ZFS Properties
- Configuring ZFS Storage Pool Properties
- Configuring ZFS File System Properties
- Managing ZFS Properties in a Non-Global Zone

Configuring ZFS File System Properties

This section covers setting the following:

- Quota and reservation properties
- Compression property
- File change ownership property

Reviewing the Quota and Reservation Property Requirements

Outstanding Property Management Company	
ZFS properties requirements	
File systems	Set 2 MB storage quota on the user `sstudent`'s home directory `/export/home/sstudent`. Set user and group storage quotas: 2 MB for the user Polly Anna and 10 MB for the staff group. Set storage reservation of 30 MB for the user `sstudent`.

According to the data management plan, you have several file system properties that you need to set for the Outstanding Property Management Company.

You first set the storage quota property to 2 MB on the home directory `rpool/export/home/sstudent` for one of the OPM application users, Super Student (`sstudent`), who is going to maintain vacation plans and holidays information in this directory.

You then set a 2 MB user quota for the user Polly Anna, who is responsible for maintaining the client file systems, and a 10 MB group quota for the staff group in the `opmpool/clients` file system, of which Polly Anna is a member. According to the planning specifications, the OPM application must maintain the US and APAC (Asia Pacific) clients centrally in the `opmpool/clients file` system.

The last file system property that you set for the OPM application is for the user Super Student (`sstudent`), who has been directed to work on a user interface prototype that is estimated to use around 28 MB of storage. According to the data management plan, you need to set a 30 MB storage reservation for this user.

In the slides that follow, you learn about these properties and about how to set them.

Quota Properties: Overview

- Use the `quota` property to set a limit on the pool space that is used by a file system.
- Use the `userquota` or `groupquota` property to set a limit on the amount of space consumed by the files that are owned by a particular user or group.
- Use the `userused` or `groupused` property to identify individual or group disk space usage.
- Use the `zfs userspace` or `groupspace` commands to display general user or group disk space usage.

You can use the `quota` property to set a limit on the amount of space that a file system can use. The property setting applies to the dataset on which it has been set and to all descendents of that dataset. For example, if a quota is set on the `hrpool/home` dataset, the total amount of space used by `hrpool/home` and all of its descendents cannot exceed the quota.

To limit the amount of space consumed by the files that are owned by a particular user or group, you use the `userquota` or `groupquota` property.

To display user and group disk space usage, use the `userused` or `groupused` property to report the amount of individual or group disk space usage by a dataset and all of its descendents. Or you can use the `zfs` command with the `userspace` or `usergroup` subcommand to display general user or group disk space usage.

Setting the `quota` Property

1. Use `zfs get quota` *`filesystem`* to determine the current quota setting for a file system.
2. Use `zfs set quota=`*`value filesystem`* to set a quota on a file system.
3. Use `zfs get quota filesystem` to verify the quota setting for a file system.
4. Use `zfs list` *`filesystem`* to see the used and available space for the file system.

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

As you just saw, you can use the `quota` property to limit the amount of space that a dataset and its descendents can consume.

This property enforces a hard limit on the amount of space used. This includes all space consumed by descendents, including file systems and snapshots. Setting a quota on a descendent of a dataset that already has a quota does not override the ancestor's quota, but instead it imposes an additional limit.

The steps for setting and displaying a storage quota property on a ZFS file system are listed in the slide.

Note: You cannot set a quota to an amount that is less than what is currently being used by a dataset.

Setting the `quota` Property: Example

```
# zfs get quota rpool/export/home/sstudent
NAME                          PROPERTY       VALUE      SOURCE
rpool/export/home/sstudent    quota          none       default
# zfs set quota=2M rpool/export/home/sstudent
# zfs get quota rpool/export/home/sstudent
NAME                          PROPERTY       VALUE      SOURCE
rpool/export/home/sstudent    quota          2M         local
# zfs list /export/home/sstudent
NAME                        USED  AVAIL  REFER  MOUNTPOINT
rpool/export/home/sstudent   35K  1.97M    35K  /export/home/sstudent
```


In this example, we show you how to address the `quota` property planning requirement.

We first display the current quota setting for the OPM application user, Super Student (`sstudent`), by using the `zfs get quota` command. The quota is currently set to `none`.

Next, according to the planning requirements, we set the storage quota property to 2 MB on the home directory `rpool/export/home/sstudent` for Super Student. We then verify the quota setting by using the `zfs get quota` command. Finally, we check the used and available space.

Setting the User Quota Property

1. Use `zfs get userquota@<name>` *`filesystem`* to determine the current user quota setting for a file system.
2. Use `zfs set userquota@<name>=`*`value`* *`filesystem`* to set a user quota on a file system.
3. Use `zfs get userquota@<name>` *`filesystem`* to verify the user quota setting for a file system.

```
# zfs create finance/tax
# zfs get userquota@rsmart finance/tax
NAME                PROPERTY            VALUE     SOURCE
finance/tax         userquota@rsmart    none      local
# zfs set userquota@rsmart=10g finance/tax
# zfs get userquota@rsmart finance/tax
NAME                PROPERTY            VALUE     SOURCE
finance/tax         userquota@rsmart    10g       local
```


You use the `userquota` property to limit the amount of ZFS file space that can be consumed by a specified user. Unlike the `quota` property, the `userquota` space calculation does not include space that is used by descendent datasets, such as snapshots and clones.

Note: You cannot use this property on volumes, on file systems before version 4, or on pools before version 15.

The steps for setting and displaying the `userquota` property are listed in the slide. In the example in the slide, assume we have created a pool called `finance`. We create a file system `finance/tax` in the pool and want to set the `userquota` to 10 GB for a user called `rsmart` on the file system.

We first check the current setting of the `userquota` property for the file system for this user. As you can see, the `userquota` setting for `rsmart` on the `finance/tax` file system is currently set to `none`. Using the `zfs set userquota` command, we set the `userquota` to 10 GB for the user `rsmart`.

Our last step is to verify the `userquota` setting.

Note: The user's name must be appended after the `@` symbol, using one of the following forms:

- POSIX name (for example, `panna`)
- POSIX numeric ID (for example, `60007`)
- SID name (for example, `polly.anna@mydomain`)
- SID numeric ID (for example, `S-1-123-456-789`)

Users can access only their own space usage. The root user, or a user who has been granted the `userquota` privilege with `zfs allow`, can get and set everyone's quota.

Note: We learn how to address the planning requirements for this property after we review the group quota property and discuss how to display the disk usage information.

Setting the Group Quota Property

1. Use `zfs get groupquota@<group>` *`filesystem`* to determine the current group quota setting for a file system.
2. Use `zfs set groupquota@<name>=`*`value`* *`filesystem`* name to set group quota on a file system.
3. Use `zfs get groupquota@<group>` *`filesystem`* to display the group quota setting for a file system.

```
# zfs create finance/ar
# zfs get groupquota@staff finance/ar
NAME                 PROPERTY             VALUE     SOURCE
finance/ar           groupquota@staff     none      local
# zfs set groupquota@staff=20GB finance/ar
# zfs get groupquota@staff finance/ar
NAME                 PROPERTY             VALUE     SOURCE
finance/ar           groupquota@staff     20G       local
```

ORACLE

You can use the group quota property to limit the amount of ZFS file system space that can be consumed by a specified group. The steps for setting and displaying the group storage property are listed in the slide. In the example in the slide, we first create the file system `finance/ar` in the `finance` pool that we created earlier. In this example, we want to set a group quota of 20 GB on this file system for the `staff` group. We first determine what the current group quota setting is on the file system for the group. As you can see, the group quota setting for the `staff` group on the `finance/ar` file system is currently set to `none`. Our next step is to set the group quota to 20 GB for the group. We then verify the setting of the group quota.

Users can access only their own groups' space usage. The root user, or a user who has been granted the `groupquota` privilege with `zfs allow`, can access and set all groups' quotas.

Note: If a user exceeds the individual or group quota, the system sends a “disk quota exceeded” notification. Enforcement of user or group quotas might be delayed, which means that users might exceed their quota before the system notices that they are over quota and refuses additional writes.

Identifying User and Group Space Usage

To identify the amount of space consumed by a specified user, use `zfs get 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 the amount of space consumed by a specified group, use `zfs get groupused@<name>` followed by the file system name.

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


As we discussed earlier, you can identify the amount of ZFS file system space that is being consumed by a specified user or a group.

You can use the `zfs get userused@<name>` command to determine the amount of space being consumed by a user, as shown in the first example in the slide. Here we identify the space used by the user `rsmart` in the file system `finance/tax`. We can see that `rsmart` is using 455 MB of space.

Note: The user's name must be appended after the `@` symbol, using one of the following forms:

- POSIX name (for example, `panna`)
- POSIX numeric ID (for example, `60007`)
- SID name (for example, `polly.anna@mydomain`)
- SID numeric ID (for example, `S-1-123-456-789`)

Similarly, you can use the `zfs get groupused@<name>` command to identify the amount of ZFS file system space consumed by a specified group, as shown in the second example. Here we identify the space used by the group `staff` in the `finance/ar` file system. We can see that the `staff` group is using 217 MB of space.

Space is charged to the user or group of each file, which you can display by using the `ls -l` command.

Users can access only their own or their own group's space usage. The root user, or a user who has been granted either the `userused` or `groupused` privilege with `zfs allow`, can access everyone's usage.

Note: The user and group quota properties are not displayed as part of the `zfs get all` *`dataset`* command output.

Displaying User and Group Space Usage

Use `zfs userspace` *`filesystem`* to display general user space usage.

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

Use `zfs groupspace` *`filesystem`* to display general group space usage.

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


You can display the amount of file system space consumed by, and the quotas on, each user and group in a specified file system or snapshot by using the `zfs userspace` and `zfs groupspace` subcommands, respectively, as illustrated in the examples in the slide.

The `zfs userspace` subcommand corresponds to the `userused@`*`user`* and `userquota@`*`user`* properties. The `zfs groupspace` subcommand corresponds to the `groupused@`*`group`* and `groupquota@`*`group`* properties.

Note: For a list of options that can be used with these two subcommands, see the `zfs` man page.

Setting the User and Group Properties: Example

```
# zfs get userquota@panna opmpool/clients
NAME             PROPERTY            VALUE              SOURCE
opmpool/clients  userquota@panna     none               local
# zfs get groupquota@staff opmpool/clients
NAME             PROPERTY            VALUE              SOURCE
opmpool/clients  groupquota@staff    none               local
# zfs set userquota@panna=2M opmpool/clients
# zfs set groupquota@staff=10M opmpool/clients
# zfs get userquota@panna opmpool/clients
NAME             PROPERTY            VALUE              SOURCE
opmpool/clients  userquota@panna     2M                 local
# zfs get groupquota@staff opmpool/clients
NAME             PROPERTY            VALUE              SOURCE
opmpool/clients  groupquota@staff    10M                local
# zfs userspace opmpool/clients
TYPE        NAME      USED         QUOTA
POSIX User  root      4.50K        none
POSIX User  panna     1.00M        2M
# zfs groupspace opmpool/clients
TYPE         NAME     USED         QUOTA
POSIX Group  root     4.50K        none
POSIX Group  staff    1.00M        10M
# zfs get userused@panna opmpool/clients
NAME             PROPERTY              VALUE             SOURCE
opmpool/clients  userused@panna        1.00M             local
# zfs get groupused@staff opmpool/clients
NAME             PROPERTY          VALUE             SOURCE
opmpool/clients  groupused@staff   1.00M             local
```


Now let's look at how to satisfy the planning requirements for the user and group quota properties. According to the data management plan for the OPM application, you need to set a 2 MB user quota for the user Polly Anna (`panna`) and a 10 MB group quota for the `staff` group in the `opmpool/clients` file system.

In the example, we begin by displaying the current quota setting for `panna` and the `staff` group in the `opmpool/clients` file system. Both quota properties are currently set to `none`. We then set the user quota to 2 MB for `panna`. We then set the group quota for the staff group to 10 MB. We then verify the user and group quota settings by using the `zfs get userquota` and `groupquota` commands, respectively. We then take a look at the user and group space consumption. Here we can see the amount of space used and the quota limit for the user and group, respectively. Currently, Polly Anna (`panna`) has used 1 MB of her 2 MB quota. Because she is a member of the `staff` group, her space consumption is reflected in the group usage as well. We then check the `userused` and `groupused` properties. At this time, because only one user (`panna`) has been active in the file system, the space usage value (1 MB) is the same for both the user and the group for the `opmpool/clients` file system.

Note: In the practice for this section, you gain direct experience of what happens when a user exceeds the `userquota` and `groupquota` property settings.

Removing Quotas

Use `zfs set quota=none` *`filesystem`* to remove a quota.

```
# zfs set quota=none rpool/export/home/sstudent
```

Use `zfs set userquota@<`*`name`*`>=none` *`filesystem`* to remove a user quota.

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

Use `zfs set groupquota@<`*`name`*`>=none` *`file system`* to remove a group quota.

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


You can remove a quota (`quota`, `userquota`, or `groupquota`) by using the `zfs set` command to set the `quota` property to `none`, as illustrated in the examples in the slide.

Now let's look at how to set reservations on ZFS files systems.

ZFS `reservation` Property: Overview

- 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 if:
  - Space is available in the pool that is unreserved
  - Its current usage is below its quota
- A dataset cannot consume space that is reserved for another dataset.
- The property is set to `off` by default.

ORACLE

Setting the `reservation` Property

1. Use `zfs get reservation` *`filesystem`* to determine the current space allocation.
2. Use `zfs set reservation=`*`amount filesystem`* to guarantee space allocation on a dataset and snapshot.
3. Use `zfs get reservation` *`filesystem`* to verify the space allocation.
4. Use `zfs list` *`filesystem`* to see the used and available space for the file system.

As you just learned, you use the `reservation` property to guarantee a minimum amount of space to a dataset and its descendents. When the amount of used space is below this value, the dataset is treated as if it were taking up the amount of space specified by its reservation. Reservations are accounted for in the parent datasets' space used, and they count against the parent datasets' quotas and reservations.

The steps for setting and displaying the storage reservation property are listed in the slide.

Setting the `reservation` Property: Example

```
# zfs set reservation=30M rpool/export/home/sstudent
cannot set property for 'rpool/export/home/sstudent': size is greater than
available space
# zfs set quota=none rpool/export/home/sstudent
# zfs get quota rpool/export/home/sstudent
NAME                        PROPERTY  VALUE  SOURCE
rpool/export/home/sstudent  quota     none   local
# zfs set reservation=30M rpool/export/home/sstudent
# zfs get reservation rpool/export/home/sstudent
NAME                        PROPERTY     VALUE  SOURCE
rpool/export/home/sstudent  reservation   30M    local
# zfs list rpool/export/home/sstudent
NAME                         USED  AVAIL  REFER  MOUNTPOINT
rpool/export/home/sstudent  2.04M  20.4G  2.04M  /export/home/sstudent
# zfs list rpool/export/home
NAME                        USED  AVAIL  REFER  MOUNTPOINT
rpool/export/home          30.2M  20.4G    37K  /export/home
```


In the example, we reserve 30 MB of storage in `rpool/export/home/` for the user Super Student (`sstudent`) as specified in the planning requirements for the OPM application. We have a problem, however, because earlier we set a quota of 2 MB for this same student and the reservation value cannot be greater than the available space.

To proceed, we set the quota property to `none`, verify the change, and then set the reservation. We verify the reservation setting by using the `zfs get reservation` command. We then check the amount of used and available space on the file system. By running the `zfs list` command for both `rpool/export/home/sstudent` and `rpool/export/home`, we can see how the reservation is accounted for in the parent's used space.

Reviewing the `compression` Property Requirements

Fantastic Products LLC	
ZFS properties requirements	
File systems	Enable compression on the `fplpool/iso` file system using the default compression algorithm.

We have looked at the file system property requirements for the Outstanding Property Management Company application. Now let's turn our attention to the Fantastic Products LLC business application and the tasks that you need to complete to satisfy their requirements.

The file system–level property task for Fantastic Products LLC is to save storage on the `fplpool/iso` file system by turning on the compression property. Note that you have been instructed to use the default compression algorithm.

ZFS `compression` Property: Overview

- The ZFS compression property is used to enable and disable compression for a file system.
- The compression property is disabled by default.
- The property is enabled by setting the compression property to on: `compression=on`.
- The values are `on`, `off`, `lzjb`, `gzip`, and `gzip-N`.
- Enabling compression on a file system with existing data will compress only new data.
- The compression ratio is inherited by child file systems.

ORACLE

You can use the ZFS `compression` property to minimize the amount of file system space that is needed to store data. This property can be used to either enable or disable compression for a file system.

The `compression` property for all file systems is set to `off` by default. The property is enabled by setting the `compression` property to `on`. The possible values for compression are `on`, `off`, `lzjb`, `gzip`, and `gzip-N`. Setting compression to `on` uses the `lzjb` compression algorithm. The `gzip` compression algorithm uses the same compression as the `gzip(1)` command. You can specify the `gzip` level by using the value `gzip-N`, where *N* is an integer from 1 (fastest) to 9 (best compression ratio). Currently, `gzip` is equivalent to `gzip-6`, which is also the default for `gzip`(1).

Enabling compression on a file system with existing data will compress only new data. Existing data will remain uncompressed. The compression ratio is calculated from the logical size of all files and the amount of referenced physical data. To view the compression ratio for a file system, use the `compressratio` property. After it is enabled, the selected `compression` property value is inherited by any descendent or child file systems that are created, as illustrated in the following example:

```
# zfs set compression=on datapool/software/solaris
    # zfs get -r compression datapool
    NAME                           PROPERTY     VALUE   SOURCE
    datapool                       compression  off     default
    datapool/software              compression  off     default
    datapool/software/solaris      compression  on      local
    datapool/software/solaris/ar   compression  on      inherited
         from datapool/software/solaris
```

In this example, we enable compression on the `datapool/software/solaris` file system. We then verify that the compression property has been turned on for that file system, and we see that it has. Note that the `SOURCE`, which had been `default`, has changed to `local`. Notice also how `datapool/software/solaris/ar` has inherited the `compression` property value of on from its parent file system.

Setting the ZFS `compression` Property

1. Use `zfs get compression` *`filesystem`* to determine the current setting of the `compression` property on the specified file system.
2. Use `zfs set compression=on` *`filesystem`* to enable the property on a specified file system.
3. Use `zfs get compression` *`filesystem`* to verify that compression has been enabled on the specified file system.

```
# zfs get compression fplpool/iso
NAME         PROPERTY     VALUE     SOURCE
fplpool/iso  compression  off       default
# zfs set compression=on fplpool/iso
# zfs get compression fplpool/iso
NAME         PROPERTY     VALUE     SOURCE
fplpool/iso  compression  on        local
```


The steps for setting the `compression` property are listed in the slide.

In the example, according to the planning requirements for Fantastic Products LLC, we enable the `compression` property on the `fplpool/iso` file system after first checking to see how the property was set. Because we do not specify the `gzip` or `gzip-N` compression algorithm, the default `lzjb` compression algorithm is used. Keep in mind that enabling compression on the `fplpool/iso` file system does not compress the existing data. Only new data is compressed. We then verify that compression has been turned on for the `fplpool/iso` file system, and we see that it has.

Note: In the practice that follows, you work with the `compression` property and experiment with different compression algorithms.

Reviewing the File Change Ownership Property Requirements

Fantastic Products LLC	
ZFS properties requirements	
File systems	Set the `rstchown` property to enable the user `jholt` to change ownership of his contacts file to user `panna`.

During planning, the business analyst for Fantastic Products LLC recognized that the user Polly Anna (`panna`) needs to have ownership of another user's file. The other user is John Holt (`jholt`), and he maintains the file of potential investors that Polly Anna needs to have ownership of to make changes to the file. To meet this requirement, you need to set the `rstchown` property to enable John Holt to change the ownership of the contacts file to Polly Anna.

In the slides that follow, you are introduced to the `rstchown` property and shown how to set it.

ZFS File Change Ownership Property: Overview

- The rstchown property is used to control a file system owner's ability to grant file ownership changes.
- By default, the rstchown property is set to on to restrict chown operations.
- Setting the rstchown property to off gives the user chown operation privileges.

Note: You can also grant the file_chown privilege to a user without changing the rstchown property default setting.

```
# usermod -K defaultpriv=basic,file_chown username
```


You use the restricted change ownership property (rstchown) to control whether a file system owner can grant file ownership changes. The default is to have this property set to on, which restricts chown operations. When rstchown is set to off, the user has the PRIV_FILE_CHOWN_SELF privilege for chown operations.

Note: In previous versions of Solaris, you had to modify the rstchown property in the /etc/system file and restart the system to make it effective. Now, because rstchown is a ZFS system property, you can change the property setting and have it become effective immediately without having to modify additional files or restart the system.

An alternative way to give a user file ownership change control without having to change the rstchown property setting is to grant the file_chown privilege by using the command described in the slide. This approach enables the user to grant file ownership to another user, but the rstchown property remains set to on.

Setting the ZFS File Change Ownership Property

1. Use `zfs get rstchown` *`user_directory`* to check the current setting of the `rstchown` property.
2. Use `zfs set rstchown=off` *`user_directory`* to grant file ownership change privileges to the user.
3. Use `zfs get rstchown` *`user_directory`* to verify that the property setting has changed.

The user can now use the `chown` command to change file ownership of files to another user.

The steps for setting the ZFS file change ownership property to grant file ownership change privileges to a user are listed in the slide.

Working with the ZFS File Change Ownership Property: Example

```
# zfs get rstchown rpool/export/home/jholt
NAME                      PROPERTY              VALUE           SOURCE
rpool/export/home/jholt   rstchown              on              default
# zfs set rstchown=off rpool/export/home/jholt
# zfs get rstchown rpool/export/home/jholt
NAME                      PROPERTY              VALUE           SOURCE
rpool/export/home/jholt   rstchown              off             default
```

```
jholt@server1:~$ chown panna /export/home/jholt/contacts
jholt@server1:~$ ls -l
total 8
-rw-r--r--   1 panna    staff         46 Apr 28 14:12 contacts
-rw-r--r--   1 jholt    staff        165 Mar 17 08:59 local.cshrc
-rw-r--r--   1 jholt    staff        170 Mar 17 08:59 local.login
-rw-r--r--   1 jholt    staff        130 Mar 17 08:59 local.profile
jholt@server1:~$ exit
logout
```


In this example, we address the planning requirement to allow John Holt to transfer file ownership of his potential investors contacts file to Polly Anna.

The first example in the slide shows the steps that the system administrator takes to set the user file change ownership property to `off`. We first verify the setting of the `rstchown` property for John Holt's home directory. The property is currently set to the default setting of `on`, which means that John does not have the privilege to transfer ownership of his files to another user. We then set the `rstchown` property for John Holt's home directory to `off` and verify the new property setting. John now has the privilege to change the ownership of his files to Polly Anna.

The second example shows John Holt granting ownership of the contacts file to Polly Anna and then verifying the change in file ownership.

Quiz

Which command do you use to display information about the amount of file system space consumed by, and about the quotas on, general users in the `finance/ar` file system?

a. `zfs userused@* finance/ar`

b. `zfs groupused@* finance/ar`

c. `zfs userspace finance/ar`

d. `zfs groupspace finance/ar`

Answer: c

Quiz

Space cannot be reserved for a dataset if that space is not currently available in the pool.

a. True

b. False

Answer: a

Quiz

When the ZFS `compression` property is set to `on`, which compression algorithm is used by default?

a. `lzjb`

b. `gzip`

c. `gzip-`*`N`*

Answer: a

Quiz

The ZFS `rstchown` property is set to `off` by default. When it is changed, it requires a system restart to make the change effective.

a. True

b. False

Answer: b

Practice 6-2 Overview: Configuring ZFS File System Properties

This practice covers working with the following:

- Storage quota property
- User and group storage quota properties
- Storage reservation property
- Compression property
- File change ownership property

This practice should take about 45 minutes to complete.

Lesson Agenda

- Implementing the Plan to Configure ZFS Properties
- Configuring ZFS Storage Pool Properties
- Configuring ZFS File System Properties
- Managing ZFS Properties in a Non-Global Zone

Managing ZFS Properties in a Non-Global Zone

This section covers the following topics:

- Working with the `zoned` property
- Verifying the `checksum` property setting

Fantastic Products LLC	
ZFS properties requirements	
Non-global zone	Ensure the `zoned` property is set to `on` in `fpzone`. Ensure that the `checksum` property on the `mfrfs` file system in the `fpzone` zone is set to `on`.

Recall from the lesson on configuring and managing the Oracle Solaris ZFS components that you had a requirement to create a zone called `fpzone` for Fantastic Products LLC, and then you delegated the ZFS file system called `mfrfs` as a dataset to that zone. You might also recall that one of the reasons for this initial requirement was to control the management of the dataset in the non-global zone.

Now you have a further requirement to ensure that the `zoned` property is set to `on` in `fpzone`. You learn the importance of this setting in the next slide. Because the manufacturer data kept in the `mfrfs` file system is critical to Fantastic Products LLC's business, they want you to ensure that the `checksum` property is set to `on`.

In this topic, you are introduced to the `zoned` and `checksum` properties and shown how to work with them.

`zoned` Property: Overview

- This property controls the management of a dataset from a non-global zone.
- This property is automatically enabled when a zone containing a ZFS dataset is first booted.
- When set, this property does not allow the dataset to be mounted or shared in the global zone.
- To prevent security risks and unpredictable system behavior, extreme caution must be taken when setting the `zoned` property to `off`.
- Only a global administrator can manually clear the `zoned` property.

The `zoned` property is used to control whether a dataset is managed from a non-global zone. The `zoned` property is automatically turned on when a non-global zone containing a ZFS dataset is first booted. A zone administrator does not need to manually enable this property.

If the `zoned` property is set, the dataset cannot be mounted or shared in the global zone.

When a dataset is removed from a zone or when a zone is destroyed, the `zoned` property is not automatically cleared. To prevent security risks, the `zoned` property must be manually cleared by the global zone administrator.

The global administrator should turn off the `zoned` property only if the dataset is no longer in use by a non-global zone. If the `zoned` property is disabled while a dataset is in use in a zone, the system might behave in unpredictable ways.

Working with the `zoned` Property

To determine if the `zoned` property is set on a dataset, use the `zfs list -o name,zoned,mountpoint -r` *`filesystem`* command.

```
# zfs list -o name,zoned,mountpoint -r /fplpool
NAME                      ZONED  MOUNTPOINT
fplpool                     off  /fplpool
fplpool/asia                off  /fplpool/asia
fplpool/bafs                off  /bafs
fplpool/custfs               on  legacy
fplpool/hpfs                off  /hpfs
fplpool/iso                 off  /fplpool/iso
fplpool/mfrfs                on  /zones/fpzone/root/mfrfs
fplpool/mfrfs/bamfr1         on  /zones/fpzone/root/mfrfs/bamfr1
fplpool/ufsdata             off  /fplpool/ufsdata
```


In the example in the slide, we are verifying that the `zoned` property is set to `on` in the `fplpool/mfrfs` file system to ensure that the planning requirement has been met.

`checksum` Property: Overview

- This property controls the checksum used to verify data integrity.
- The default value is `on`.
- When enabled, the property automatically selects an appropriate algorithm.

The default value is `on`, which automatically selects an appropriate algorithm, which is currently `fletcher4`. The values for this property are `on`, `off`, `fletcher2`, `fletcher4`, `sha256`, and `sha256+mac`.

Verifying the `checksum` Property Setting

1. Use `zfs get checksum` *`filesystem`* to verify the setting of the ZFS `checksum` property for a file system.
2. If necessary, use `zfs set checksum=on` *`filesystem`* to enable the `checksum` property for a file system.
3. Use `zfs get checksum` *`filesystem`* to verify that the ZFS `checksum` property is now set to `on` for the file system.

The steps for setting the `checksum` property are listed in the slide.

Note for step 1: By default, the `checksum` property should be set to `on`.

Note: To set the `checksum` property to `off`, use `zfs set checksum=off` *`filesystem`*. Because this action disables integrity checking on user data, this setting is not recommended.

Verifying the `checksum` Property Setting: Example

```
# zlogin fpzone
[Connected to zone 'fpzone' pts/2]
Oracle Corporation      SunOS 5.11      11.0    November 2011
root@fpzone:~# zfs get checksum mfrfs
NAME   PROPERTY  VALUE       SOURCE
mfrfs  checksum  on         local
root@fpzone:~# exit
logout
```


In the example, we satisfy the planning requirement to verify that the `checksum` property is set to `on` for the `mfrfs` file system in `pfzone`. We first log in to the zone. We then display the `checksum` property for the `mfrfs` file system. The property is currently set to `on`, which (as you know) is the default value. After we verify the property setting, we exit the zone.

Quiz

Which statement about the ZFS `zoned` property is true?

a. The property controls the management of a dataset from a non-global zone.

b. The property must be manually enabled when the ZFS dataset is created.

c. When set, this property allows the dataset to be mounted or shared in the global zone.

d. The zone administrator is responsible for immediately clearing the `zoned` property when the ZFS dataset is removed.

Answer: a

Quiz

The ZFS `checksum` property controls the checksum that is used to verify data integrity.

a. True
b. False

Answer: a

Practice 6-3 Overview: Configuring ZFS Properties in a Non-Global Zone

This practice covers the following topics:

- Working with the `zoned` property
- Verifying the `checksum` property setting

This practice should take about 10 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan to configure ZFS properties
- Configure the ZFS storage pool properties
- Configure the ZFS file system properties
- Manage ZFS properties in a non-global zone

7

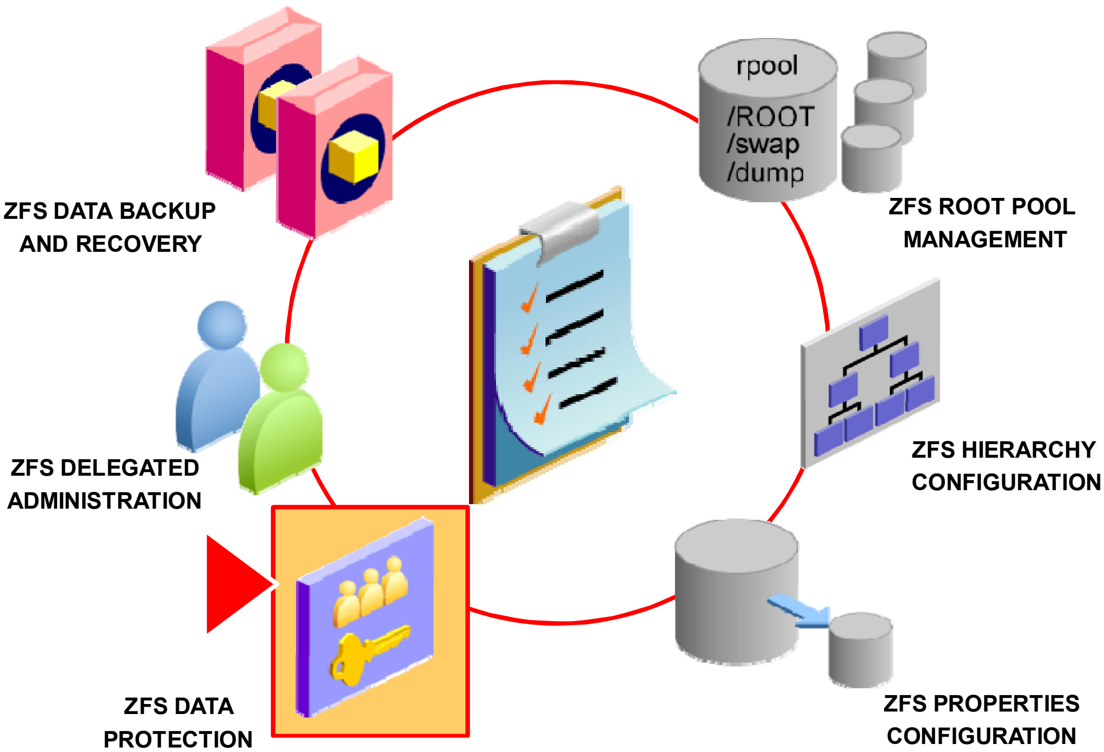
Protecting Data by Using ZFS ACLs

Objectives

After completing this lesson, you should be able to:

- Implement a plan for data protection by using ZFS ACLs
- Configure ZFS ACLs
- Manage ZFS ACLs

Planning Workflow: Orientation



You have completed the ZFS properties configuration tasks according to the data management plan. You are now ready to proceed to the next set of tasks, which require you to set ZFS ACLs on the ZFS file systems.

In this lesson, you learn about the new Solaris ACL model that ZFS uses, as well as how to set the ACLs in different formats.

Lesson Agenda

- Implementing the Plan for Data Protection by Using ZFS ACLs
- Configuring and Managing ZFS ACLs

Implementing the Plan for Data Protection by Using ZFS ACLs

Fantastic Products LLC	
ZFS ACLs	
Configure ACLs for Directories and Files in Verbose Format	Create a group called `fplgroup` and assign the users Polly Anna (`panna`) and Jerry Moose (`jmoose`) to this group. Set permissions for the members of `fplgroup` to read, write, and execute on the Beauty Aids Product Line (`baprodline`) and Health Products Product Line (`hpprodline`) file systems in the `mfrs` pool.

Manage the ACLs	Set permissions on the Beauty Aid product line data as follows:
	• Jerry Moose (`jmoose`): `baline1` directory: Grant write data and add subdirectory permissions.
	• Polly Anna (`panna`):
	  • `baline2` directory: Grant read and write.
	  Set `file_inherit`:
	    • `nails1`: Read only
	    • `skinbeauty`: Read, write, and execute (set `aclmode` property to `passthrough`)

Configure ACLs in Compact Format	• John Holt (`jholt`): `hairpins`: Read and execute • Polly Anna (`panna`): `hairpins`: Read, write, and execute

ORACLE

In the lesson titled "Planning for Data Management," you were presented with a requirement to provide additional privacy protection on Fantastic Products LLC's potential manufacturers and their product data by using ZFS ACLs. The table in the slide identifies each requirement and specifies the permissions that are needed by users and groups based on their required level of access to the data.

In the slides that follow, you learn about the ZFC ACL model and about setting, displaying, and managing trivial and non-trivial file and directory ACLs in both verbose and compact formats. You also learn how to work with the ACL inheritance flags and properties.

New Solaris ACL Model: Overview

- Is based on the NFSv4 specification and similar to Windows NT-style ACLs
- Uses the `chmod` and `ls` commands to set and display ACLs
- Provides the following:
  - More fine-grained access control through multiple access control entries (ACEs)
  - Granular set of access privileges
  - ACL inheritance option

ZFS uses a new Solaris ACL model, which, although it is similar to the old UFS Solaris ACL model, has some differences. The new ACL model is based on the Network File System Version 4 (NFSv4) specification and is similar to Windows NT-style ACLs. With this model, you use the `chmod` command to set ACLs and the `ls` command to display them.

The new ACLs are composed of multiple access control entries (ACEs), which provide more fine-grained access control. In addition, the new model provides a much more granular set of access privileges than the older model and provides richer inheritance semantics through the ACL inheritance option for designating how access privileges are applied from directory to subdirectories and files.

ACL Formats

- Trivial ACLs:

```
chmod [options] A[index]{+|=}owner@ |group@ |everyone@:access-
    permissions/...[:inheritance-flags]:deny | allow file

chmod [options] A-owner@, group@, everyone@:access-
    permissions/...[:inheritance-flags]:deny | allow file ...

chmod [options] A[index]- file
```

- Non-trivial ACLs:

```
chmod [options] A[index]{+|=}user|group:name:access-
    permissions/...[:inheritance-flags]:deny | allow file

chmod [options] A-user|group:name:access-permissions/...[:inheritance-
    flags]:deny | allow file ...

chmod [options] A[index]- file
```


The new ACL model has two basic formats:

- **Trivial ACLs:** Represent the traditional UNIX `owner/group/other` entries
- **Non-trivial ACLs:** Are specifically set for a particular user or group

The syntax options for setting trivial and non-trivial ACLs specifications are provided in the slide.

The uppercase letter `A` specifies the command as an ACL operation and has the following options:

- **`A[index]+acl_specification`:** Prepends the access control entries (ACE) specified in `acl_specification` to the beginning of the file's ACL
- **`A-`:** Removes all ACEs for the current ACL on the file and replaces the current ACL with a new ACL that represents only the current mode of the file
- **`Aindex-`:** Removes the ACE specified by the index number
- **`A-acl_specification`:** Removes the ACEs specified by the `acl_specification` (if they exist in the current file's ACL)
- **`A=acl_specification`:** Replaces a file's entire ACL with the `acl_specification`
- **`A[index]+acl_specification`:** Replaces ACEs starting at a specific index number in the current ACL on the file. If multiple ACEs are specified, each subsequent ACE in the `acl_specification` replaces the corresponding ACE in the current ACL.

Each ACE in an ACL consists of the following:

- ACL entry type(s)
  - For trivial ACL syntax: `owner@`, `group@`, `everyone@`
  - For non-trivial ACL syntax: `user or group:ACL-entry-ID=`*`username`* or *`groupname`*
- List of access permissions: `access-permissions/.../`
- (Optional) List of inheritance flags: `inheritance-flags`
- Deny or allow access setting: `deny` or `allow`

Trivial File ACL in Verbose and Compact Format: Examples

```
$ ls -v q4report
-rw-r--r--   1 root     root      93 May 2 14:15 q4report
       0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
       /read_attributes/write_attributes/read_acl/write_acl/write_owner
       :allow
       1:group@:read_data/read_xattr/read_attributes/read_acl:allow
       2:everyone@:read_data/read_xattr/read_attributes/read_acl:allow
```

ACEs

Index number

Entry type

Permissions

Deny | Allow

```
$ ls -V q4report
-rw-r--r--   1 root     root      93 May 2 14:20 q4report
                 owner@:rw-p--RWaAcCo-:-------:allow
                 group@:r-----R-a-c---:-------:allow
                 everyone@:r-----R-a-c---:-------:allow
```

ORACLE

The first example in the slide shows a trivial file ACL in verbose format. You display the file ACL by using the `ls -v` command. In the output, you can see how each ACE corresponds with the general owner, group, and other file permissions. The labels in the example identify each field in an ACE. From left to right, the fields are the index number, the entry type, the list of permissions, and the Deny/Allow specification.

Note: For purposes of demonstration, the `synchronize` permission, which is not currently implemented but appears as default permission, has been removed in both examples.

A basic or default ACL, similar to the one shown here, is generated when you create a file. The basic ACL conforms to the general permissions for the file. The permissions are as follows:

- **`0:owner@:`** The owner can read and modify the contents of the file, modify the file's attributes (such as time stamps, extended attributes, and ACLs), and modify the ownership of the file.
- **`1:group@:`** The group is granted read permission to the file and the file's attributes.
- **`2:everyone@:`** Everyone who is not a user or group is granted read permission to the file and the file's attributes.

For comparison purposes, the same file ACL is presented in compact format in the second example. Note that to display the ACL in compact format, you use the `ls` command with the uppercase `-V` option.

Note: In Solaris 10, as a practice, every ACE contains an allow entry and a deny entry. In Oracle Solaris 11, the deny entry is included only when specified.

Non-Trivial Directory ACL in Verbose and Compact Format: Examples

```
# ls -dv qtrrpts
drwxr-xr-x+  2 root     root           2 May 2 14:30 qtrrpts
     0:user:omai:list_directory/read_data/execute:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```

```
# ls -dV qtrrpts
drwxr-xr-x+  2 root     root           2 May 2 14:35 qtrrpts
             user:omai:r-x-----------:-------:allow
                owner@:rwxp-DaARWcCos:-------:allow
                group@:r-x---a-R-c--s:-------:allow
             everyone@:r-x---a-R-c--s:-------:allow
```


The first example in the slide shows a non-trivial directory ACL in verbose format. You display the directory ACL by using the `ls -dv` command. In the output, you can see that we have specified the user `omai` and given him read and execute permissions in this directory. The plus sign (+) that appears at the end of the general permissions indicates that this is a non-trivial ACL. For comparison purposes, the same directory ACL is presented in compact format in the second example. Note that to display the directory ACL in compact format, you use the `ls` command with the `-dV` option.

Note: When you create a new directory, the default directory ACL permissions are as follows:

- **`0:owner@`:** The owner can read and modify the directory contents and can read and modify the file's attributes (such as time stamps, extended attributes, and ACLs). In addition, the owner can search the contents, delete a file or directory, and modify the ownership of the directory.
- **`1:group@`:** The group can list and read the directory contents and the directory's attributes. In addition, the group has execute permission to search the directory contents.
- **`2:everyone@`:** Everyone who is not a user or group is granted read and execute permissions to the directory contents and the directory's attributes. The synchronize access permission is not currently implemented.

ACLs and `umask` Settings

- `umask` is set to 022:

```
-rw-r--r--   1 root     root          93 May  2 09:19 q4report
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

- `umask` is set to 000:

```
-rw-rw-rw-   1 root     root          82 May  2 09:25 mayreport
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/synchronize:allow
```


There is a relationship between ACLs and the `umask` value. When you create a file and specify an ACL for that file, the files permissions are set to whatever you specify minus whatever `umask` forbids. For example, if `umask` is set to the default value of `022` (as shown in the first example), the owner has general read/write permissions and the group and everyone else have read permission only; they are denied general write and execute permissions. If you look at the ACL for this file, `owner@`, `group@`, and `everyone@` have permissions that correspond to the `umask` value.

If you change the `umask` value to `000`, you effectively allow all files and directories to be created with all available permissions and no filtering by `umask`. Notice the effect on the ACL in the file in the second example. In this file, `owner@`, `group@`, and `everyone@` all have the same read/write permissions.

The examples in the slide illustrate how the general permissions (and therefore the ACL permissions) are generated for a file when `umask` is different. Similarly, the general permissions and the ACL entries for a directory are affected by `umask`.

ACL Entry Types

ACL Entry Type	Description
`owner@`	Specifies the access granted to the owner of the object
`group@`	Specifies the access granted to the owning group of the object
`everyone@`	Specifies the access granted to any user or group that does not match any other ACL entry
`user`	With a user name, specifies the access granted to an additional user of the object; must include the **ACL-entry-ID**, which contains a *username* or *userID*
`group`	With a group name, specifies the access granted to an additional group of the object; must include the **ACL-entry-ID**, which contains a *groupname* or *groupID*

ORACLE

The table in the slide displays the ACL entry types. The top three entries are used with the trivial ACL format; the bottom two entries are used with the non-trivial format.

ACL Access Privileges

Access Privilege	Compact Access Privilege	Description
`add_file`	`w`	Permission to add a new file to a directory
`add_subdirectory`	`p`	On a directory, permission to create a subdirectory
`delete`	`d`	Permission to delete a file
`delete_child`	`D`	Permission to delete a file or directory in a directory
`execute`	`x`	Permission to execute a file or search the contents of a directory
`list_directory`	`r`	Permission to list the contents of a directory
`read_acl`	`c`	Permission to read the ACL (`ls`)
`read_attributes`	`a`	Permission to read basic attributes (non-ACLs) of a file.

You can specify access privileges in either verbose or compact format. The access privileges are shown in both formats in the table in the slide.

Note for `delete` and `delete_child`: For additional details about `delete` and `delete_child` permission behavior, see “Syntax Descriptions for Setting ACLs” in the *Oracle Solaris Administration: ZFS File Systems* guide.

Note for `read_attributes`: Think of basic attributes as the stat-level attributes. Allowing this access mask bit means that the entity can execute `ls`(1) and `stat`(2).

ACL Access Privileges

Access Privilege	Compact Access Privilege	Description
`read_data`	`r`	Permission to read the contents of the file
`read_xattr`	`R`	Permission to read the extended attributes of a file or perform a lookup in the file's extended attributes directory
`write_xattr`	`W`	Permission to create extended attributes or write to the extended attributes directory
`write_data`	`w`	Permission to modify or replace the contents of a file
`write_attributes`	`A`	Permission to change the time stamp associated with a file or directory to an arbitrary value
`write_acl`	`C`	Permission to write the ACL or the ability to modify the ACL by using the `chmod` command
`write_owner`	`o`	Permission to change the file's owner or group; or the ability to execute the `chown` or `chgrp` commands on the file

The information displayed in the table is continued from the previous slide.

Note for `write_xattr`: Granting this permission to a user means that the user can create an extended attribute directory for a file. The attribute file's permissions control the user's access to the attribute.

Note for `write_owner`: This is permission to take ownership of a file or permission to change the group ownership of the file to a group of which the user is a member. If you want to change the file or group ownership to an arbitrary user or group, the `PRIV_FILE_CHOWN` privilege is required.

Note: ZFS supports using pre-defined ACL combinations that can be applied as an ACL set. For more information about these combinations, see “ZFS ACL Sets” in the *Oracle Solaris Administration: ZFS File Systems* guide.

ACL Inheritance: Overview

- You can control ACL inheritance at the directory and file levels.
- You use a combination of inheritance permission flags and ACL property settings to control ACL inheritance.
- If you make changes to an ACL entry at the directory level, the children are not updated.
- You must specify the inheritance of an ACL entry on a directory or file by using an ACL inheritance flag:
  – Subdirectory: `dir_inherit` or `d` flag
  – File: `file_inherit` or `f` flag
- To apply an ACL entry to children but not to the parent directory, use `the inherit_only` or `i` flag.
- To restrict ACL inheritance to the immediate children of a parent directory, use the `no_propagate` or `n` flag.

ORACLE

The new Solaris ACL model enables you to control ACL inheritance at the directory and file levels by using a combination of inheritance permission flags and ACL property settings. You can flag any ACE as inheritable. Setting an inheritance flag on an ACL entry tells ZFS to copy that entry to new subdirectories or files. If you make changes to an ACL entry at the directory level, the children are not updated.

By default, ACLs are not propagated through the directory structure. If you set a non-trivial ACL on a directory, it is not inherited to any subsequent directory. You must specify the inheritance of an ACL on a directory or file by using an ACL inheritance flag. To specify inheritance for a newly created subdirectory, you use the `dir_inherit` or `d` flag. To specify inheritance for a newly created file, you use the `file_inherit` or `f` flag. You can specify both these flags in a single ACL entry.

If you want an entry to apply to children but not to the parent directory itself, you use the `inherit_only` or i flag. You must use this flag in conjunction with the directory flag, the file flag, or both to specify what to inherit.

You can restrict the inheritance of an ACL entry to the immediate children of a parent directory by using the `no_propagate` or `n` flag. As with the `inherit_only` flag, you must use this flag in conjunction with the directory flag, the file flag, or both to specify what to inherit.

ACL Inheritance Flags

Inheritance Flag	Compact Inheritance Flag	Description
`file_inherit`	`f`	Inherit the ACL only from the parent directory to the directory's files
`dir_inherit`	`d`	Inherit the ACL only from the parent directory to the directory's subdirectories
`inherit_only`	`i`	Inherit the ACL from the parent directory, but this applies only to newly created files or subdirectories and not the directory itself. This flag requires the `file_inherit` flag, the `dir_inherit` flag, or both to indicate what to inherit.
`no_propogate`	`n`	Inherit the ACL only from the parent directory to the first-level contents of the directory (not the second-level or subsequent contents). This flag requires the `file_inherit` flag, the `dir_inherit` flag, or both to indicate what to inherit.
-	N/A	No permission granted

The table in the slide displays the current ACL inheritance flag options.

Note: There are several flags that are applicable only to an SMB client or server. For a list of these flags, see "ACL Inheritance" in the *Oracle Solaris Administration: ZFS File Systems* guide.

ZFS ACL Properties

- **`aclinherit`**: Determines the behavior of ACL inheritance
  - `discard`
  - `noallow`
  - `restricted` (default)
  - `passthrough`
  - `passthrough-x`
- **`aclmode`**: Modifies ACL behavior when a file is initially created, or controls how an ACL is modified during a `chmod` operation
  - `discard` (default)
  - `mask`
  - `passthrough`

The behavior of ZFS ACL inheritance is determined by the `aclinherit` property, which can have the following values:

- **`discard`:** For new objects, no ACL entries are inherited when a file or directory is created. The ACL on the file or directory is equal to the permission mode of the file or directory.
- **`noallow`:** For new objects, only inheritable ACL entries that have an access type of deny are inherited.
- **restricted:** For new objects, the `write_owner` and `write_acl` permissions are removed when an ACL entry is inherited. This is the default value for the `aclinherit` property.
- **`passthrough`:** Files are created with a mode determined by the inheritable ACEs. If no inheritable ACEs exist that affect the mode, the mode is set according to the requested mode from the application.
- **`passthrough-x`:** This has the same semantics as `passthrough`, except that when `passthrough-x` is enabled, files are created with the execute (`x`) permission, but only if execute permission is set in the file creation mode and in an inheritable ACE that affects the mode.

To modify ACL behavior when you initially create a file or to control how an ACL is modified during a `chmod` operation, you can use the `aclmode` property. This property can have the following values:

- **`discard`:** Deletes all ACL entries that do not represent the mode of the file. This is the default value for the `aclmode` property.
- **`mask`:** Reduces user or group permissions. The permissions are reduced so that they are no greater than the group permission bits, unless it is a user entry that has the same UID as the owner of the file or directory. In this case, the ACL permissions are reduced so that they are no greater than owner permission bits. The `mask` value also preserves the ACL across mode changes if an explicit ACL set operation has not been performed.
- **`passthrough`:** Indicates that no changes are made to the ACL other than generating the necessary ACL entries to represent the new mode of the file or directory

These properties are set using the `zfs set` command and displayed using the `zfs get` command.

Primary Rules of ACL Access on a ZFS File

- ZFS processes ACL entries in the order they are listed in the ACL (from the top down).
- Before an ACL entry is processed, a check is performed to ensure that the requester has appropriate permissions for the ACL command.
- After an allow permission has been granted, it cannot be denied by a subsequent ACL deny entry in the same ACL permission set.
- The owner of the file is granted the `write_acl` permission unconditionally, even if the permission is explicitly denied. Otherwise, any permission that is left unspecified is denied.

The slide lists the primary rules for ACL access on a ZFS file system. Take a moment to familiarize yourself with these rules before continuing.

In the cases of deny permissions or missing access permissions, the privilege subsystem determines what access request is granted for the owner of the file or for superuser. This mechanism prevents owners of files from getting locked out of their files and enables superuser to modify files for recovery purposes.

Quiz

You want to display the ACL entries for the `qtrrpts` directory in compact format. Which command do you use?

a. `ls -v qtrrpts`

b. `ls -dv qtrrpts`

c. `ls -V qtrrpts`

d. `ls -dV qtrrpts`

Answer: d

Quiz

Which permission set indicates the presence of a non-trivial ACL entry? (Select all that apply.)

a. `-rw-r--r--`
b. `-rw-r--r--+`
c. `drwxr-xr-x`
d. `drwxr-xr-x+`

Answer: b, d

Quiz

ACL inheritance at the directory level is set by default.

a. True

b. False

Answer: b

Quiz

What are the default settings of the ACL `aclinherit` and `aclmode` properties?

a. `aclinherit=passthrough`; `aclmode=passthrough`

b. `aclinherit=discard`; `aclmode=mask`

c. `aclinherit=restricted`; `aclmode=discard`

d. `aclinherit=noallow`; `aclmode=mask`

Answer: c

Lesson Agenda

- Implementing the Plan for Data Protection by Using ZFS ACLs
- Configuring and Managing ZFS ACLs

Configuring and Managing ZFS ACLs

This section covers the following topics:

- Adding an ACL entry
- Inserting an ACL entry by index ID
- Replacing an ACL entry by index ID
- Replacing an entire ACL entry
- Removing an ACL entry
- Removing an ACL entry by index ID
- Removing all ACL entries from a file
- Setting the ACL inheritance `file_inherit` flag
- Setting the ACL inheritance `dir_inherit` flag
- Setting ZFS ACL properties

Adding an ACL Entry

Use `chmod A+acl-specification` *`filename`* to add an ACL entry.

```
# ls -v q4reports
-rw-r--r--   1 root     root          82 May  2 09:25 q4reports
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
# chmod A+user:omai:read_data:allow q4reports
# ls -v q4reports
-rw-r—-r--+   1 root     root          82 May  2 09:30 q4reports
     0:user:omai:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

ORACLE

To add an ACL entry, use the command syntax shown in the slide. In the example, we add the user `omai` to the `q4reports` file and grant him read permission. We begin by displaying the current basic ACL for the file. Using the `chmod A+acl-specification` command, we add the read and execute permissions for `omai` to the `q4reports` file. We then verify the change by running the `ls -v` command again. We see that the change has been made. Notice that the new entry now has an index ID of `0`, and the original `owner`, `group`, and `everyone` entries now have index IDs of `1`, `2`, and `3`, respectively.

Note: This action is also referred to as prepending a new ACL entry on a ZFS file.

Inserting an ACL Entry by Index ID

Use `chmod Aindex-ID+acl-specification` *`filename`* to add an ACL entry by index ID.

```
# ls -v q4reports
-rw-r—-r--+   1 root     root          82 May  2 09:30 q4reports
     0:user:omai:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
# chmod A2+group:audit:read_data/write_data:allow q4reports
# ls -v q4reports
-rw-r—-r--+   1 root     root          82 May  2 09:39 q4reports
     0:user:omai:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group:audit:read_data/write_data:allow
     3:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     4:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


Similarly, you can insert an ACL into a specific position on a ZFS file by using the command syntax shown in the slide. This syntax inserts the new ACL entry at the specified index ID location. In the example, we add the group `audit` to the `q4reports` file and grant this group `read_data`, `write_data`, `execute`, and `read_acl` permissions. We begin by displaying the current ACL for the file. Using the `chmod Aindex-ID+acl-specification` command, we add the permissions for the `audit` group to the `q4reports` file.

Note: You must ensure that the audit group exists before you can use it.

We then verify the change by running the `ls -v` command again. We see that the change has been made. Notice that the new ACL entry has been inserted in position 2. This causes the entries that are currently in positions 2 to 4 to be pushed down.

Replacing an ACL Entry by Index ID

Use `chmod Aindex-ID=acl-specification` *`filename`* to replace an ACL entry by index ID.

```
# ls -v q4reports
-rw-r--r--+   1 root     root           82 May  2 09:39 q4reports
     0:user:omai:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group:audit:read_data/write_data:allow
     3:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     4:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
# chmod A0=user:omai:read_data:deny q4reports
# ls -v q4reports
-rw-r--r--+   1 root     root           82 May  2 09:44 q4reports
     0:user:omai:read_data:deny
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group:audit:read_data/write_data:allow
     3:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     4:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


You can replace a specific ACL entry by using the index ID. The command syntax for completing this action is shown in the slide. In the example, we are replacing the `read_data:allow` permission for the user `omai` with the `read_data:deny` permission. We then verify the change.

Replacing an Entire ACL Entry

Use `chmod A=acl-specification` *`filename`* to replace an entire ACL entry.

```
# ls -v q4reports
-rw-r—-r--+   1 root     root          82 May  2 09:44 q4reports
     0:user:omai:read_data:deny
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group:audit:read_data/write_data:allow
     3:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     4:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
# chmod A=owner@:read_data/write_data:allow,group@:read_data/ \
write_data:allow,user:omai:read_data:allow q4reports
# ls -v q4reports
-rw-rw----+   1 root     root          82 May  2 09:50 q4reports
     0:owner@:read_data/write_data:allow
     1:group@:read_data/write_data:allow
     2:user:omai:read_data:allow
```


If you want to replace an ACL entry in its entirety, you can do so by using the command syntax shown in the slide. In the example, we are replacing the entire ACL entry with a new entry for `owner@`, `group@`, and the user `omai`. We grant read/write data permissions to `owner@` and `group@` and read data permission to the user `omai`. We then verify the change. Notice the change to the general permissions.

Removing an ACL Entry

Use `chmod A-acl-specification` *`filename`* to remove an ACL entry.

```
# ls -v q4reports
-rw-rw----+   1 root     root           82 May  2 09:50 q4reports
     0:owner@:read_data/write_data:allow
     1:group@:read_data/write_data:allow
     2:user:omai:read_data:allow
# chmod A-user:omai:read_data:allow q4reports
# ls -v q4reports
-rw-rw----    1 root     root           82 May  2 09:53 q4reports
     0:owner@:read_data/write_data:allow
     1:group@:read_data/write_data:allow
```

ORACLE

You can remove an ACL entry in several ways. The first way is to use the command syntax shown in the slide. With this method, you specify a user.

In the example, we remove the user `omai` ACL entry from the `q4reports` file. We then verify that the entry has been removed, and we see that it has. Note that the plus sign (+) indicating a non-trivial ACL has been removed.

Removing an ACL Entry by Index ID

Use `chmod Aindex-ID- filename` to remove an ACL entry by index ID.

```
# ls -v q4reports
-rw-rw----    1 root     root          82 May  2 09:53 q4reports
     0:owner@:read_data/write_data:allow
     1:group@:read_data/write_data:allow
# chmod A1- q4reports
# ls -v q4reports
-rw-------    1 root     root          82 May  2 10:02 q4reports
     0:owner@:read_data/write_data:allow
```


You can also remove an ACL entry by specifying the index ID for that entry, as shown in the command syntax in the slide. In the example, we remove the `group@` entry from the `q4reports` file by specifying index ID `1` in the `chmod` command. We verify that this entry has been removed, and we see that it has. Again, note the change in the general permissions.

Removing All ACL Entries from a File

Use `chmod A- filename` to remove all ACEs from a file.

```
# ls -v q4reports
-rw-------    1 root     root          82 May  2 10:02 q4reports
     0:owner@:read_data/write_data:allow
# chmod A- q4reports
# ls -v q4reports
-rw-r--r--   1 root     root          82 May  2 10:05 q4reports
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


You can remove all the ACL entries from a file by using the command syntax shown here. As shown in the example, when we remove the ACL, it is automatically replaced with an ACL that represents the general permissions of the file.

Setting the ACL Inheritance `file_inherit` Flag

To grant inheritance to newly created files in a directory, create an inheritable ACL entry on the directory and include the `file_inherit` flag.

```
# chmod A+user:omai:read_data:file_inherit:allow qtrrpts
# ls -dv qtrrpts
drwxr-xr-x+  2 root     root           2 May 2 14:30 qtrrpts
     0:user:omai:read_data/file_inherit:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
# ls -v qtrrpts/junerpt
-rw-r--r--+  1 root    root          5 June 1   14:45 qtrrpts/mayrpt
     0:user:omai:read_data:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


Now let's look at how you set ACL inheritance flags on directories and files, as well as what the impact of these flags is. In the example, we want the user `omai` to be able to read any newly created files in the `qtrrpts` directory. To do this, we create an ACL entry for user `omai` that specifies the read data permission and the `file_inherit` flag. To verify that the inheritance is working as expected, we create a new file called `mayrpt` (not shown) and then run the `ls -v` command. As you can see, the `omai` ACL entry is inherited to the newly created `junerpt` file in the `qtrrpts` directory.

Setting the ACL Inheritance `dir_inherit` Flag

To grant inheritance to newly created files and subdirectories in a directory, create an inheritable ACL entry on the directory and include the `file_inherit` and `dir_inherit` flags.

```
# chmod A0=user:omai:read_data/write_data/execute:file_inherit/dir_inherit:allow qtrrpts
# ls -dv qtrrpts
drwxr-xr-x+  2 root     root           2 July 1 09:00 qtrrpts
     0:user:omai:list_directory/read_data/add_file/write_data/execute
         :file_inherit/dir_inherit:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
# ls -v qtrrpts/junerpt
-rw-r--r--+  1 root     root          5 July 1   10:03 qtrrpts/junerpt
     0:user:omai:read_data:allow
      1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


Now let's assume that the decision has been made to grant user `omai` read, write, and execute permissions that are inherited for newly created files and directories in the `qtrrpts` directory. In the example, we replace the existing ACL entry in the `qtrrpts` directory for the user `omai` and verify the change. Now, whenever a new file or subdirectory is created in the `qtrrpts` directory, the read, write, and execute permissions for the user `omai` ACL entry are inherited with some exceptions.

The behavior of the ACL inheritance is determined by the setting of the `aclinherit` property. In this scenario, suppose that the user `omai` is granted read, write, and execute permissions at the directory level. In this case, if the `aclinherit` property is set to the default of `restricted` (which means that `write_data` and `execute` permissions are not inherited, and the permission bits of the parent directory for `group@` and `everyone@` deny write and execute permissions), the user `omai` is also denied write and execute permissions.

In addition, because the `aclmode` property is set to the default value of `discard`, all ACL entries that do not represent the mode of the file are deleted by the file.

ACL Inheritance and ZFS ACL Properties

Level of Inheritance Restriction	ACL Property Settings	Resulting Behavior
Default	`aclmode: discard` `aclinherit: restricted`	All ACL entries that do not represent the mode of the file are deleted, and the `write_owner` and `write_acl` permissions are removed.
Strictest	`aclmode: discard` `aclinherit: discard`	All ACL entries that do not represent the mode of the file are deleted, and no ACL entries are inherited when a file or directory is created.
Stricter	`aclmode: discard` `aclinherit: noallow`	All ACL entries that do not represent the mode of the file are deleted, and the only ACL entries that have an access type of `deny` are inherited.
Less strict	`aclmode: discard` `aclinherit: passthrough`	All ACL entries that do not represent the mode of the file are deleted, and files are created with a mode determined by the inheritable ACEs.
Least strict	`aclmode: discard` `aclinherit: passthrough-x`	All ACL entries that do not represent the mode of the file are deleted, and files are created with a mode determined by the inheritable ACEs. In addition, files are created with the execute (`x`) permission, but only if the execute permission is set in the file creation mode and in an inheritable ACE that affects the mode.

As you have already seen, the settings of the ZFS ACL properties, along with the mode and `umask` settings, affect the behavior of ACL inheritance. The table in the slide displays a sampling of ACL property settings and their effect on the level of inheritance restriction. The level indications (strictest, stricter, less strict, and least strict) are based on the default setting for both properties and assume a `umask` value of `022` (default) and a mode of `644`. Note also that the `aclmode` setting is kept at `discard` to make the differences in the resulting behavior easier to discern.

Take a moment to review the information displayed in the table to familiarize yourself with the interaction between the ACL properties and the effect of the settings on inheritance behavior. Keep in mind that these combinations represent a small sampling. You can also affect the level of permission and inheritance restriction by changing the `umask` and mode values as well as through different ACL property setting pairings.

Setting the ZFS ACL `aclinherit` Property: Example

```
# zfs get aclinherit mfrs/baprodline
NAME              PROPERTY    VALUE          SOURCE
mfrs/baprodline   aclinherit  restricted     default
# zfs set aclinherit=noallow mfrs/baprodline
# zfs get aclinherit mfrs/baprodline
NAME              PROPERTY    VALUE          SOURCE
mfrs/baprodline   aclinherit  noallow        local

<New directory baline3 directory is created and a file called powders is created in
the baline3 directory.>

# chmod A+user:panna:read_data/write_data:file_inherit:allow baline3
# ls -v baline3/powders
-rw-r--r--   1 root     root          17 May  4 09:30 baline3/powders
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


In this example, we change the `aclinherit` property setting on the `mfrs/baprodline` file system from the default value of `restricted` to `noallow`. We verify that the change has been made to the property setting; we see that it has.

We then create an inherited ACL entry for user `panna` that grants her read and write file permissions in the `baline3` directory. We create a new file called `powders` in the `baline3` directory. When we check the ACL entries for the file, we see that the ACL entry for user `panna` is not included. This is expected because the `aclinherit` property for the `mfrs/baprodline` file system is set to `noallow`, which means that for any new files, only inheritable ACL entries that have an access type of `deny` are inherited.

See "Setting and Displaying ACLs on ZFS Files in Compact Format" in the *Oracle Solaris Administration: ZFS File Systems* guide for more examples of ACL inheritance behavior with the ACL properties.

Quiz

You want to remove the following entry from `q4reports`:

`2:user:omai:read_data:allow`

Which command do you use?

a. `chmod A=user:omai:read_data:allow q4reports`

b. `chmod A+user:omai:read_data:allow q4reports`

c. `chmod A-user:omai q4reports`

d. `chmod A2- q4reports`

Answer: d

Quiz

You have created an inheritable ACE whose mode you want to have inherited by any new files that you create. Assuming that the `aclmode` property is set to `discard`, how would you set the `aclinherit` property to achieve this behavior?

a. `discard`

b. `restricted`

c. `noallow`

d. `passthrough`

Answer: d

Reviewing the ACL Requirements

Fantastic Products LLC	
ZFS ACLs	
Configure ACLs for Directories and Files in Verbose Format	Create a group called `fplgroup` and assign the users Polly Anna (`panna`) and Jerry Moose (`jmoose`) to this group. Polly Anna can now set permissions for the members of `fplgroup` to read, write, and execute on the Beauty Aids Product Line (`baprodline`) and Health Products Product Line (`hpprodline`) file systems in the `mfrs` pool.

Manage the ACLs	Set permissions on the Beauty Aid product line data as follows:
	• Jerry Moose (`jmoose`): `baline1` directory: Grant write data and add subdirectory permissions.
	• Polly Anna (`panna`):
	  • `baline2` directory: Grant read and write.
	  Set `file_inherit`:
	    • `nails1`: Read only
	    • `skinbeauty`: Read, write, and execute (set `aclmode` property y to `passthrough`)

Configure ACLs in Compact Format	• John Holt (`jholt`): `hairpins1`: Read and execute • Polly Anna (`panna`): `hairpins1`: Read, write, and execute

You now have the information you need to address the ACL requirements of the data management plan. The details are displayed in the slide. In the slides that follow, you learn how to satisfy each requirement.

Configure ACLs for Directories and Files: Example

```
<Assume you have created fplgroup and assigned panna and jmoose to it. You
have also created the baprodline and hpprodline file systems in the mfrs
pool.>

panna@client1:/mfrs/baprodline$ chmod A1=group@:write_data/read_data:allow \
eurobeauty
panna@client1:/mfrs/baprodline$ ls -v eurobeauty
-rw-rw-r--   1 panna    fplgroup       88 May  2 09:40 eurobeauty
     0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:read_data/write_data:allow
     2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```

```
jmoose@client1:/mfrs/baprodline$ echo Dutch Beauty Products > eurobeauty
jmoose@client1:/mfrs/baprodline$ more eurobeauty
Dutch Beauty Products
jmoose@client1:/mfrs/baprodline$ exit
logout
```


The first requirement is for you to create a group called `fplgroup` and assign the users Polly Anna (`panna`) and Jerry Moose (`jmoose`) to this group. Assume that you have created the Beauty Aids Product Line (`baprodline`) and Health Products Product Line (`hpprodline`) file systems, which are part of the `mfrs` pool, and that you have created the `fplgroup` and assigned `panna` and `mjoose` to that group.

Now that the file systems and group have been created, Polly Anna can create files and set read and write group permissions on those files for the members of `fplgroup`. In the first example, Polly Anna (`panna`) sets the group permissions on the `eurobeauty` file. She then verifies the change.

Now that user `panna` has modified the ACL entry for the group, Jerry Moose (`jmoose`) has write permission on this file as well, as shown in the second part of the example, where `jmoose` is able to successfully add a record to the `eurobeauty` file.

Managing ZFS ACLs: Example

```
<Assume you have created the baline1 and baline2 directories.>

root@client1:/mfrs/baprodline# ls -dv baline1
drwxr-xr-x   2 root     root           2 May  4 08:25 baline1
     0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     1:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
root@client1:/mfrs/baprodline# chmod \
A+user:jmoose:write_data/add_subdirectory:allow baline1
root@client1:/mfrs/baprodline# ls -dv baline1
drwxr-xr-x+  2 root     root           2 May  4 08:25 baline1
     0:user:jmoose:add_file/write_data/add_subdirectory/append_data:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
```


The next requirement is for you to set permissions on the Beauty Aids Product Line data as follows:

- Jerry Moose (`jmoose`): `baline1` directory: Grant write data and add subdirectory permissions.
- Polly Anna (`panna`):
  - `baline2` directory
    - `nails1`: Read only
    - `skinbeauty`: Read, write, and execute

Assume that you have already created the `baline1` and `baline2` directories. In the example, you display the `baline1` directory ACL. You then grant `jmoose` the `write_data` and `add_subdirectory` permissions, and you verify the change.

Managing ZFS ACLs: Example

```
root@client1:/mfrs/baprodline# chmod \
A+user:panna:read_data/write_data:file_inherit:allow baline2
root@client1:/mfrs/baprodline# ls -dv baline2
drwxr-xr-x+  2 root     root           2 May  4 08:58 baline2
     0:user:panna:read_data/write_data:file_inherit:allow
     1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
         /append_data/read_xattr/write_xattr/execute/delete_child
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
     3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
         /read_acl/synchronize:allow
root@client1:/mfrs/baprodline# ls -v baline2/nails1
-rw-r--r--+  1 root     root          14 May  4 09:01 baline2/nails1
     0:user:panna:read_data:inherited:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


You now configure the ACL entries for Polly Anna (`panna`) in the `baline2` directory. In the example, you add the ACL entry for user `panna`. You grant her read/write data permissions and set the `file_inheritance` flag so that new files created for this directory inherit this ACL entry. You verify that the new entry has been added. You then verify that `panna` has read-only permission on the newly created `nails1` file in the `baline2` directory, and you see that she does.

Managing ZFS ACLs: Example

```
root@client1:/mfrs/baprodline# zfs get aclmode mfrs/baprodline
NAME             PROPERTY    VALUE          SOURCE
mfrs/baprodline  aclmode     discard        default
root@client1:/mfrs/baprodline# zfs set aclmode=passthrough mfrs/baprodline
root@client1:/mfrs/baprodline# zfs get aclmode mfrs/baprodline
NAME             PROPERTY  VALUE        SOURCE
mfrs/baprodline  aclmode   passthrough  local

<Assume you have created a new file called skinbeauty.>

root@client1:/mfrs/baprodline# ls -v baline2/skinbeauty
-rw-r--r--+  1 root     root          14 May  4 09:18 baline2/skinbeauty
     0:user:panna:read_data/write_data:inherited:allow
     1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
         /read_attributes/write_attributes/read_acl/write_acl/write_owner
         /synchronize:allow
     2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
     3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
         :allow
```


The next requirement is to grant user `panna` read, write, and execute permissions to the `skinbeauty` file in the `baline2` directory. Because you have already created an ACL entry for user `panna` at the directory level that inherits these permissions at the file level, you need to set the ZFS ACL `aclmode` property to `passthrough` to enable these permissions to be inherited at the file level.

In the example, you first check the current setting of the `aclmode` property. It is set to the default: `discard`. You then set the `aclmode` property to `passthrough` and verify the new property setting. You create the new `skinbeauty` file (not shown), and then you verify that the ACL entry for user `panna` contains both the read and write data inherited permissions. You see that it does.

Configuring ACLs in Compact Format: Example

```
<Assume you have created the hairpins1 file in /mfrs/baprodline/baline1.>

root@client1:/mfrs/baprodline/baline1# chmod A+user:jholt:rx:allow hairpins1
root@client1:/mfrs/baprodline/baline1# ls -V hairpins1
-rw-r--r--+  1 root     root          14 May  4 08:57 hairpins1
             user:jholt:r-x-----------:-------:allow
                 owner@:rw-p--aARWcCos:-------:allow
                 group@:r-----a-R-c--s:-------:allow
              everyone@:r-----a-R-c--s:-------:allow
root@client1:/mfrs/baprodline/baline1# chmod \
A+user:panna:rwx-----------:-------:allow hairpins1
root@client1:/mfrs/baprodline/baline1# ls -V hairpins1
-rw-r--r--+  1 root     root          14 May  4 08:57 hairpins1
             user:panna:rwx-----------:-------:allow
        user:jholt:r-x-----------:-------:allow
                 owner@:rw-p--aARWcCos:-------:allow
                 group@:r-----a-R-c--s:-------:allow
              everyone@:r-----a-R-c--s:-------:allow
```


The final planning requirement is to create ACL entries for the users John Holt (`jholt`) and Polly Anna (`panna`) on a file called `hairpins1` in the `/mfrs/baprodline/baline1` file system. The requirement calls for you to grant the user `jholt` read and execute permissions and the user `panna` read, write, and execute permissions.

Note: This example uses the compact format, although the verbose format could also be used.

In the example, assume that you have already created the `hairpins1` file. You now create the ACL entry for user `jholt`. You then verify the entry in the file, and you copy the `jholt` entry and modify it to create the ACL entry for user `panna`. Finally, you verify the `panna` entry.

Practices 7-1 and 7-2 Overview: Configuring ACLs for Directories and Files and Managing ZFS ACLs

Practice 7-1 covers the following topics:

- Working with trivial ACLs
- Managing access control entries (ACEs)

Practice 7-2 covers the following topics:

- Using ACL properties
- Working with the ACL compact format

These combined practices should take about 60 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for data protection by using ZFS ACLs
- Configure ZFS ACLs
- Manage ZFS ACLs

8

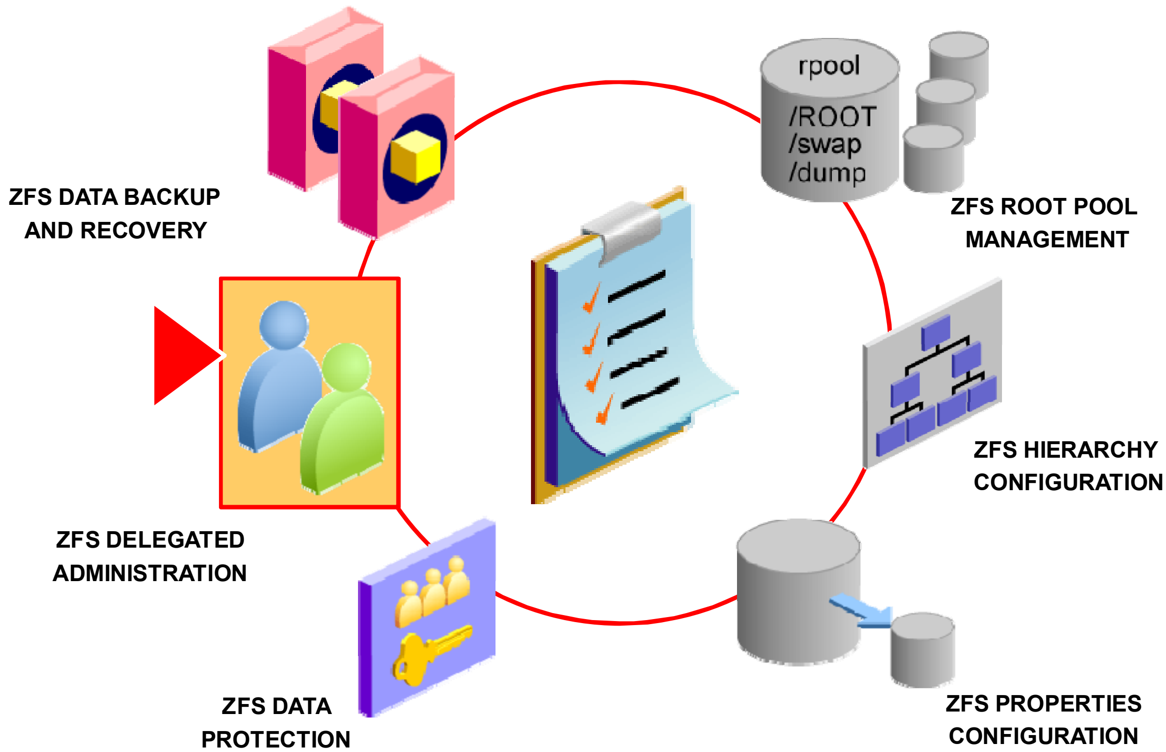
Using ZFS Delegated Administration

Objectives

After completing this lesson, you should be able to:

- Implement a plan for ZFS delegated administration
- Configure a ZFS rights profile
- Enable the ZFS `delegation` property
- Disable the ZFS `delegation` property
- Delegate ZFS permissions
- Remove ZFS delegated permissions

Planning Workflow: Orientation



You have completed configuring and managing the ZFS ACLs tasks according to the data management plan. You are now ready to proceed to the next set of tasks, which require you to use ZFS delegated administration.

In this lesson, you learn about ZFS delegated administration, configuring a ZFS rights profile, and delegating ZFS permissions.

Lesson Agenda

- Implementing the Plan for ZFS Delegated Administration
- Configuring the ZFS Delegated Administration Model

Implementing the Plan for ZFS Delegated Administration

This section covers the following topics:

- Reviewing the plan for ZFS delegated administration
- ZFS rights profile: overview
- ZFS delegated administration model: overview
- ZFS delegated permissions (sampling)
- ZFS delegated properties

Reviewing the Plan for ZFS Delegated Administration

Outstanding Property Management Company		Fantastic Products LLC	
ZFS Delegated Administration		**ZFS Delegated Administration**	
ZFS Rights Profiles	Create and assign the ZFS administration role (`zfsadmin`) to Jerry Moose (`jmoose`).	**ZFS Rights Profiles**	N/A
ZFS Permissions	N/A	**ZFS Permissions**	Grant ZFS delegated permissions to Polly Anna (`panna`) to include `destroy` on her home directory. Assign Polly Anna (`panna`), John Holt (`jholt`), and Jerry Moose (`jmoose`) to a group (`fplgroup`). Create a permission set (`adminset`) for the `mfrs` pool's file system and attach the permission set to `fplgroup`.

In the lesson titled "Planning for Data Management," you learned about the requirements for assigning ZFS administration tasks to non-privileged users for both the Outstanding Property Management Company and Fantastic Products LLC. The ZFS delegated administration requirements are shown in this slide.

In the slides that follow, you learn about ZFS rights profiles and ZFS delegated permissions. You learn how to use both to delegate ZFS administration tasks to users and groups.

ZFS Rights Profile: Overview

- You assign ZFS rights profiles to nonprivileged users through a role.
- You can assign one of two ZFS profiles:
  - ZFS Storage Management: Grants the privilege to create, destroy, and manipulate devices in a ZFS storage pool
  - ZFS File System Management: Grants the privilege to create, destroy, and modify ZFS file systems

There are several ways you can give nonprivileged users the authority to perform ZFS management tasks. One way is to assign them a role with either of the following two ZFS profiles:

- **ZFS Storage Management:** Provides the privilege to create, destroy, and manipulate devices in a ZFS storage pool
- **ZFS File System Management:** Provides the privilege to create, destroy, and modify ZFS file systems

ZFS Delegated Administration Model: Overview

- You can assign permissions to specific users or groups, or to everyone.
- You can use two types of delegated permissions:
  - Individual permissions
  - Permission sets (which are groups of permissions)
- You can delegate permissions to specific file systems in either of the following ways:
  - Locally to the current file system only
  - To all descendents of the current file system
- You control the delegated administration permissions by using a pool’s `delegation` property, which is enabled by default.

A second way to grant non-root users the authority to perform ZFS management tasks is by delegating ZFS permissions. ZFS has a delegated administration model that is similar to the RBAC security model. With this feature, you can assign more granular permissions to specific users or groups or everyone on specific file systems.

There are two types of delegated permissions:

- Individual permissions, such as create, destroy, mount, snapshot, and so on
- Permission sets, which are groups of permissions. You can update a permission set. When you do, all of the consumers of the set automatically get the change.

You can control how the permissions propagate through the file systems. You can delegate permissions locally to the current file system only or to all descendents of the current file system. With the latter option, newly created file systems automatically pick up the specified permissions.

You control the delegated administration permissions by using a pool's `delegation` property, which is enabled by default. The permissions follow the ZFS storage pool whenever a pool is migrated.

Note: You can use ZFS rights profiles and ZFS delegated permissions together. They are not mutually exclusive.

ZFS Delegated Permissions (Sampling)

Permission (subcommand)	Description	Dependencies
`clone`	Permission to clone any of the dataset's snapshots	Must also have the create permission and the mount permission in the original file system
`create`	Permission to create descendent datasets	Must also have the `mount` permission
`destroy`	Permission to destroy a dataset	Must also have the `mount` permission
`mount`	Permission to mount and unmount a file system, and to create and destroy volume device links	
`receive`	Permission to create descendent file systems with the `zfs receive` command	Must also have the `mount` permission and the `create` permission
`send`	Permission to send a snapshot stream	
`snapshot`	Permission to create a snapshot of a dataset	

Permissions are generally the ability to use a ZFS subcommand or change a ZFS property. Using the `zfs allow` command, you can delegate permissions and properties to non-privileged users for administering ZFS file systems in a ZFS storage pool.

A partial list of ZFS permissions (subcommands) is displayed in this slide, and a list of ZFS properties you can delegate to non-root users is displayed in the next slide. For a complete list of ZFS permissions and permission sets that can be delegated, see the `zfs_allow`(1M) man page.

Note that several of the permissions have dependencies.

ZFS Delegated Properties

aclinherit	encryption	recordsize	snapdir
aclmode	exec	refquota	sync
atime	keysource	refreservation	utf8only
canmount	logbias	reservation	version
casesensitivity	mountpoint	rstchown	volblocksize
checksum	nbmand	secondarycache	volsize
compression	normalization	setuid	vscan
copies	primarycache	shadow	xattr
dedup	quota	sharenfs	zoned
devices	readonly	sharesmb	

You can delegate the administration of these ZFS properties to non-root users.

Quiz

To grant a non-root user the authority to perform ZFS management tasks by using a ZFS rights profile, you must _____________.

a. Assign the ZFS rights profile directly to the user
b. Create the ZFS rights profile and assign permissions to it
c. Create a role, assign the ZFS rights profile to the role, and assign the role to the user
d. Create a permission set and include the ZFS rights profile as one of the permissions

Answer: c

Quiz

Which *two* statements about ZFS delegated permissions are correct? (Choose two that apply.)

a. Permission sets can be delegated to groups only.
b. Delegated permissions include ZFS subcommands and ZFS properties.
c. You use `zfs set` to delegate permissions.
d. You can delegate permissions locally to the current file system only and/or to all descendents of the current file system.

Answer: b, d

Quiz

A pool's ZFS delegation property is enabled by default.

a. True
b. False

Answer: a

Lesson Agenda

- Implementing the Plan for ZFS Delegated Administration
- Configuring the ZFS Delegated Administration Model

Configuring the ZFS Delegated Administration Model

This section covers the following topics:

- Configuring a ZFS rights profile
- Enabling and disabling the `delegation` property
- Delegating ZFS permissions
- Delegating a ZFS permission set
- Removing ZFS delegated permissions

Configuring a ZFS Rights Profile

1. Use `roleadd -m -d dir` *`rolename`* to create a role and attach it to the ZFS profile.
2. Use `usermod -R` *`rolename username`* to assign the role to the user.
3. Use `roles` *`username`* to verify the role assignment.
4. Use `profiles` *`rolename`* to verify that the profile is associated with the new role.
5. Use `passwd` *`rolename`* to create a password for the role.

ORACLE

As we discussed in the planning topic, you can assign a ZFS rights profile to a nonprivileged user by creating a role, attaching the ZFS rights profile to the role, and then assigning the role to the user. The user then has the privileges to perform the ZFS management tasks specified in the profile.

The steps for configuring a ZFS rights profile are listed in the slide.

Note: To create a role, you must be an administrator with the User Management rights profile. To assign a password to the role, you must be assigned the User Security rights profile.

Note for step 2: The `-R` option replaces any existing role setting.

Configuring a ZFS Rights Profile: Example

```
# roleadd -u 1000 -g 10 -d  /export/home/zfsadmin -m -P "ZFS
File System Management"  zfsadmin
80 blocks
# usermod -R zfsadmin jmoose
Found user in files repository.
# roles jmoose
zfsadmin
# profiles zfsadmin
zfsadmin:
          ZFS File System Management
          Basic Solaris User
          All
# passwd zfsadmin
New Password: oracle123
Re-enter new Password: oracle123
passwd: password successfully changed for zfsadmin
```

ORACLE

Recall from the data management plan that you have a requirement to create and assign the ZFS administration role (`zfsadmin`) to Jerry Moose (`jmoose`) for the Outstanding Property Management Company. Being assigned this role will enable Jerry Moose to administer the file systems in his home directory and maintain the new property information.

In the example in the slide, we demonstrate how to meet this requirement. We first create a role called `zfsadmin` that uses the ZFS File System Management profile. Next, using the `usermod -R` command, we assign the new role to `jmoose`. We then verify the role assignment to the user `jmoose` and find that this user does have the `zfsadmin` role. We then verify that the ZFS File System Management profile is included in the `zfsadmin` role, and we see that it is.

Our final step is to create a password for the new role.

Enabling and Disabling the `delegation` Property

- Use `zpool set delegation=on|off` *`dataset`* to set the `delegation` property.
- Use `zpool get delegation` *`dataset`* to view the `delegation` property setting.

```
# zpool get delegation rpool
NAME   PROPERTY     VALUE        SOURCE
rpool  delegation   on           default
# zpool set delegation=off rpool
# zpool get delegation rpool
NAME   PROPERTY     VALUE        SOURCE
rpool  delegation   off          local
# zpool set delegation=on rpool
# zpool get delegation rpool
NAME   PROPERTY     VALUE        SOURCE
rpool  delegation   on          local
```


Recall from the planning topic that you control the delegated administration permissions by using a pool's `delegation` property and that this property is enabled by default. This property controls whether a nonprivileged user is granted access based on the dataset permissions defined on the dataset. To set this property, use the `zpool set delegation=` command. To disable the property, set it to `off`. To enable it, set it to `on`. To view the property setting, use the `zpool get delegation` command.

In this example, we see that the `delegation` property is set to `on` (the default) on `rpool`. For demonstration purposes, we set the property to `off` on the pool, view the setting, and then return the delegation property setting to `on`. When then verify the setting.

Delegating ZFS Permissions

1. Use `zfs allow [-uge]|user|group|everyone [,...] filesystem|volume` to specify individual permissions.
2. Use `zfs allow filesystem` to verify the permissions.

```
# zfs allow panna create,destroy,mount rpool/export/home/panna
# zfs allow rpool/export/home/panna
---- Permissions on rpool/export/home/panna -------------------------
Local+Descendent permissions:
user panna create,destroy,mount
```


To delegate individual permissions to a user or group or to everyone, use the `zfs allow` command with the syntax shown in the slide.

Multiple entities can be specified as a comma-separated list. If you do not specify any `-uge` options, the argument is interpreted preferentially as the keyword `everyone`, then as a user name, and finally as a group name. To specify a user or group named "everyone," use the `-u` or `-g` option. To specify a group with the same name as a user, use the `-g` option.

Note for step 1: You can also specify where the permissions are delegated. You can specify the `-l` option to allow the permissions to be applied "locally" only for the specified file system, you can use the `-d` option to allow the permissions for the descendent file systems only, or you can use both together. If you don't specify either option, the permissions are allowed for the file system or volume and all of its descendents. For a full listing of the subcommand options that can be used with the `zfs allow` command, see the `zfs_allow`(1M) man page.

In the example in the slide, we show how to grant ZFS delegated permissions to Polly Anna (`panna`) according to the data management plan. The planning requirements indicate that we should grant her permission to create, destroy, and mount files on her home directory. Recall from the planning topic that the `create` and `destroy` permissions require the `mount` permission.

Using the `zfs allow` command, we then verify the permissions granted to `panna`.

Delegating a ZFS Permission Set

1. Use `zfs allow -s @setname [,...]` *`filesystem|volume`* to specify a permission set.
2. Use `zfs allow user|group|everyone @setname` *`filesystem|volume`* to delegate the permission set to a user, a group, or everyone.
3. Use `zfs allow` *`filesystem|volume`* to verify the delegated permission set.

ORACLE

In addition to granting individual delegated permissions, you can also delegate permission sets. Permission sets are identified by the `-s` option and specified by using the syntax shown in the slide. Permission sets follow the same naming requirements as ZFS file systems, but the name must begin with an “at” sign (`@`) and can be no more than 64 characters in length.

The steps for delegating a ZFS permission set are listed in the slide.

Delegating a ZFS Permission Set: Example

```
# grep fplgroup /etc/group
fplgroup::101:
# usermod -g 101 jholt
Found user in files repository.
# zfs allow -s @adminset create,destroy,snapshot,rollback,clone,
promote,rename,mount,send,receive,quota,reservation mfrs
# zfs allow mfrs
---- Permissions on mfrs ---------------------------------------
Permission sets:
    @adminset clone,create,destroy,mount,promote,quota,receive,
    rename,reservation,rollback,send,snapshot
# zfs allow fplgroup @adminset mfrs
# zfs allow mfrs
---- Permissions on mfrs ---------------------------------------
Permission sets:
      @adminset clone,create,destroy,mount,promote,quota,receive,
rename,reservation,rollback,send,snapshot
Local+Descendent permissions:
    group fplgroup @adminset
```


In this example, according to the ZFS delegated administration planning requirements for Fantastic Products LLC, we demonstrate how to:

- Assign Polly Anna (`panna`), John Holt (`jholt`), and Jerry Moose (`jmoose`) to the group `fplgroup` (which you created in the lesson on ZFS ACLs)
- Create a permission set called `adminset` for the `mfrs` file system
- Attach the permission set to `fplgroup`

Our first step is to identify the group ID for `fplgroup`. It is `101`. Next we add `jholt` to `fplgroup`.

Note: Because you assigned `panna` and `jmoose` to `fplgroup` as part of the last lesson, we only show how to add `jholt` to the group in this example.

Then, using the `zfs allow -s @`*`setname`* command, we create the permission set `adminset` and specify several permissions to be included in the set. These are the permissions that `panna`, `jholt`, and `jmoose` need to manage the data in this file system. Note that the permissions include both ZFS subcommand and property permissions. Our next step is to verify the creation of the permission set in the `mfrs` file system. We then attach the permission set to `fplgroup` and verify that the permission set has been delegated to the group. We see that it has.

Removing ZFS Delegated Permissions

Use `zfs unallow` *`user perm,perm filesystem`* to remove a specified permission for a user.

```
# zfs unallow panna destroy rpool/export/home/panna
# zfs allow rpool/export/home/panna
---- Permissions on rpool/export/home/panna --------------------------
Local+Descendent permissions:
user panna create,mount
```

Use `zfs unallow` *`user filesystem`* to remove all delegated permissions for a user.

```
# zfs unallow panna rpool/export/home/panna
# zfs allow rpool/export/home/panna
```

ORACLE

To remove or revoke delegated permissions, use the `zfs unallow` command. You can remove one or more specified permissions for a user (as shown in the first example in the slide), or you can remove all the delegated permissions (as shown in the second example).

Note: If you do not specify any permissions, all permissions are removed, resulting in the entire set being removed.

In the first example, we revoke `panna`'s delegated permission to destroy file systems in her home directory. We verify that the permission has been removed.

In the second example, we revoke all of `panna`'s ZFS delegated permissions. We verify that the permissions have been removed. As you can see, no permissions are returned when we execute the `zfs allow` command for the `panna` home directory.

Removing ZFS Delegated Permission Sets

Use `zfs unallow [-r] -s @setname [perm|@setname[,...]]` *`filesystem|volume`* to remove specified permissions in a permission set.

```
# zfs unallow -s @adminset quota,reservation mfrs
# zfs allow mfrs
---- Permissions on mfrs ---------------------------------------
Permission sets:
    @adminset clone,create,destroy,mount,promote,receive,
      rename,rollback,send,snapshot
Local+Descendent permissions:
    group fplgroup @adminset
```

Use `zfs unallow [-r] -s @setname` *`filesystem|volume`* to remove the entire set.

```
# zfs unallow -s @adminset mfrs
# zfs allow mfrs
```


You also use the `zfs unallow` command to remove or revoke delegated permission sets. You can remove one or more specified permissions from a permission set (as shown in the first example in the slide), or you can remove the entire set of permissions (as shown in the second example).

Note: If you do not specify any permissions, all permissions are removed, resulting in the entire set being removed.

In the first example, we revoke `the` delegated `quota` and `reservation` property permissions from `adminset` on the `mfrs` file system. We verify that the permissions have been removed.

Note: To remove the leftover permission entry, you use `zfs unallow fplgroup mfrs`.

In the second example, we revoke the entire `adminset` permission set. We verify that the `adminset` has been removed. As you can see, the `adminset` is no longer listed.

Quiz

You want to delegate the create and destroy permissions to the user `jholt` on his home directory. Which command do you use?

a. `zfs allow jholt create,destroy rpool/export/home/jholt`

b. `zfs allow jholt create,destroy,mount rpool/export/home/jholt`

c. `zfs set jholt create,destroy rpool/export/home/jholt`

d. `zfs set jholt create,destroy,mount rpool/export/home/jholt`

Answer: b

Quiz

You want to create a permission set called `zfsadminset` that contains the create, destroy, mount, snapshot, and clone permissions for the `hrrpts` file system and then give the permission set to the user `omai`. Which *two* of the following commands do you use? (Choose two that apply.)

a. `# zfs allow @zfsadminset create,destroy,mount,snapshot,clone hrrpts`

b. `# zfs allow -s @zfsadminset create,destroy,mount,snapshot,clone hrrpts`

c. `# zfs allow omai @zfsadminset hrrpts`

d. `# zfs allow @zfsadminset hrrpts`

Answer: b, c

Quiz

You want to revoke the `quota` property permission for the user `sbeach` on the `hrrpts` file system. Which command do you use?

a. `zfs unallow sbeach hrrpts`

b. `zfs unallow -s sbeach quota hrrpts`

c. `zfs unallow sbeach quota hrrpts`

Answer: c

Practices 8-1 and 8-2 Overview: Configuring a ZFS Rights Profile and Delegating ZFS Permissions

- Practice 8-1 covers configuring a ZFS rights profile.
- Practice 8-2 covers the following topics:
  - Allowing and unallowing ZFS permissions
  - Working with a ZFS permission set

These combined practices should take about 30 minutes to complete.

Summary

In this lesson, you should have learned how to:
- Implement a plan for ZFS delegated administration
- Configure a ZFS rights profile
- Enable the ZFS `delegation` property
- Disable the ZFS `delegation` property
- Delegate ZFS permissions
- Remove ZFS delegated permissions

9

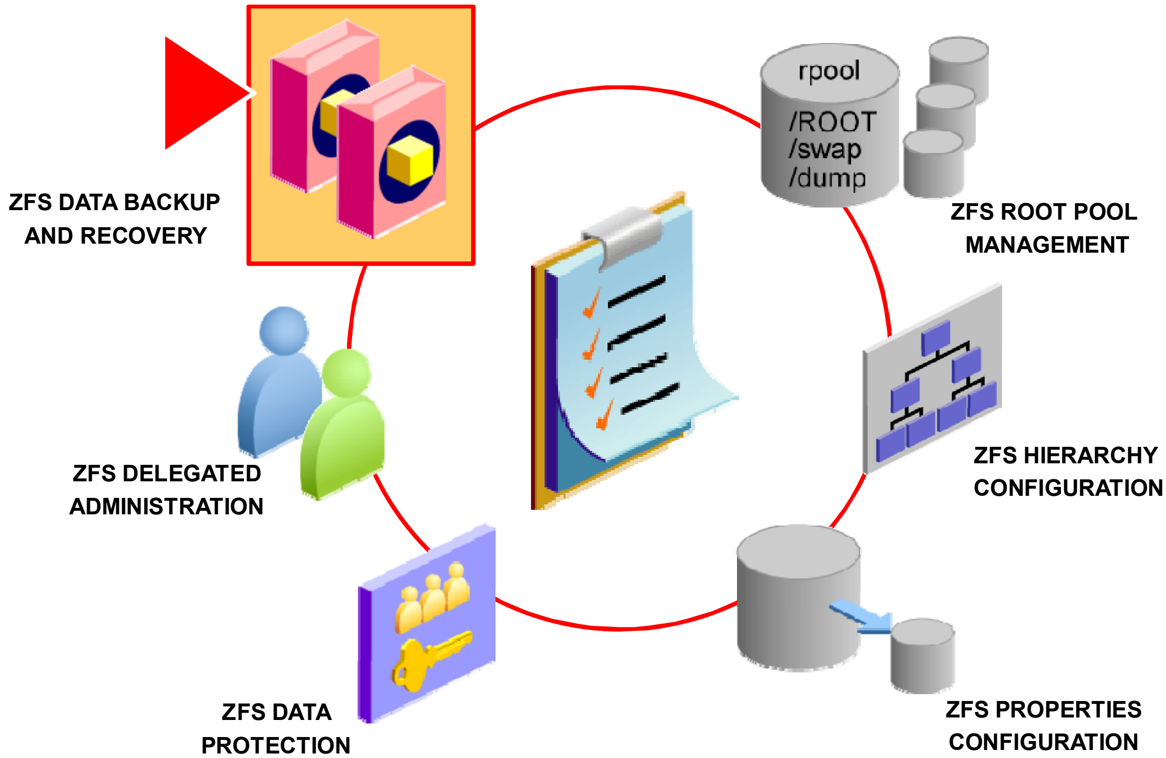
Backing Up and Recovering ZFS Data

Objectives

After completing this lesson, you should be able to:

- Implement a plan for backing up and restoring ZFS data
- Back up ZFS data by using snapshots
- Back up and restore ZFS data remotely

Planning Workflow: Orientation



You have completed the ZFS delegated administration tasks in accordance with the data management plan. You are now ready to proceed to the final set of tasks, which require you to back up and restore ZFS data.

In this lesson, you learn about the ways to back up and restore ZFS data, including how to back up data using snapshots and how to back up data remotely and recover it using ZFS streams.

Lesson Agenda

- Implementing a Plan for Backing Up and Restoring ZFS Data
- Backing Up ZFS Data by Using Snapshots
- Backing Up and Restoring ZFS Data Remotely

Implementing a Plan for Backing Up and Restoring ZFS Data

This section covers the following topics:

- Reviewing the plan for backing up and restoring ZFS data
- Solutions for backing up and recovering ZFS data
- ZFS snapshot streams overview
- Receiving a snapshot stream
- Advantages and disadvantages of using ZFS snapshot streams for backup and recovery
- File system backup and recovery with snapshots
- Time Slider service overview
- Clone creation and promotion overview

Reviewing the Plan for Backing Up and Recovering ZFS Data

Outstanding Property Management Company		Fantastic Products LLC	
ZFS Data Backup		**ZFS Data Backup**	
ZFS Data Backup Using Snapshots	Take a monthly backup called `may2012` of the `opmpool/clients` file system. Provide the differences between the `may2012` and `june2012` snapshots of `opmpool/clients/renters` to the application analyst. Create a clone of the `renters@june2012` snapshot and promote it.	**ZFS Data Backup Using Snapshots**	Set up automatic snapshots on `mfrs/fplusers` for every 10 minutes.
ZFS Data Remote Backup	Back up the `renters@june2012` snapshot to a remote system. Create a ZFS recovery archive for `admpool` and recover the data.	**ZFS Data Remote Backup**	Create a ZFS recovery archive for `admpool` and recover the data. Note: This is a shared requirement.

In the lesson titled "Planning for Data Management," you learned about the requirements for backing up and recovering ZFS data for both the Outstanding Property Management Company and Fantastic Products LLC applications. The ZFS data backup requirements are presented in this slide. In the slides that follow, you learn about solutions for backing up ZFS data and how to use a subset of these solutions to address the requirements specified in the data management plan.

Solutions for Backing Up and Recovering ZFS Data

- Enterprise backup products
- Remote file system replication with snapshot streams
- File system backup and recovery with snapshots
- Incremental file system changes with snapshots

Several solutions are available for backing up and recovering ZFS data. The solution that is right for your company depends on your company's data backup requirements. For example, if your company needs per-file restoration as well as media backup verification and management, you should consider enterprise backup products, such as the Sun ZFS Backup Appliance.

Note: See the online *ZFS Best Practices Guide* for recommendations on using ZFS with enterprise backup solutions.

If you do not want to deal with special configuration or hardware requirements, you can use the remote file system replication solution to back up and restore your ZFS data. The advantage of replicating a ZFS file system is that you can re-create a file system on a storage pool on another system and specify different levels of configuration for the newly created pool, such as RAID-Z, but with identical file system data. You can expand this solution to include creating a recursive snapshot archive of the root pool for system recovery purposes. You can create snapshot archives of non-root pools as well.

Another option is to create snapshots of your file systems as a backup, either manually or automatically. Then, if any files are accidentally changed or deleted, you can easily revert to a previous version of the file system by rolling back the snapshot.

The last option, incremental snapshots, does not provide a complete backup solution for saving your ZFS data, but it does enable you to save incremental changes between snapshots. You cannot restore individual files with this approach. Instead, you have to restore the entire file system snapshot.

In this lesson, you are shown how to back up and restore ZFS data using three out of the four solutions presented in the slide. Using enterprise backup products for ZFS data backup and recovery is out of scope for this course.

ZFS Snapshot Streams: Overview

- You can use ZFS snapshot streams to:
  - Back up data to another pool on the same system
  - Store backup data to a pool on a remote system
- The `zfs send` command converts a snapshot of a ZFS file system or volume into a snapshot stream.
- Stream format types
  - Full stream (default)
  - Incremental stream
- Stream package types
  - Replication stream package (`zfs send -R`)
  - Recursive stream package (`zfs send -r`)

As we just discussed, ZFS offers several solutions for backing up and restoring your data. If you have a business requirement to back up your ZFS data to another pool on the same system or to store your ZFS backup data remotely, you need to use snapshot streams. A snapshot stream is a converted snapshot of a ZFS file system or volume. The conversion takes place when you run the `zfs send` command. You then use the snapshot stream as the basis for recreating the ZFS file system or volume on the remote system by using the `zfs receive` command.

Generally speaking, there are two types of snapshot stream formats. The first format type is the full stream, and it is the default format. The full stream contains one file system or volume, up to and including the specified snapshot. The stream does not contain snapshots other than the specified snapshot. When you send a full stream, the destination file system must not exist. The second format type is the incremental stream, which consists of the differences between one snapshot and another snapshot.

You can use one or more full or incremental streams to create a stream package. There are two types of stream packages.

- Replication stream package, which consists of the dataset you have specified and its descendents. It includes all intermediate snapshots. A cloned dataset's origin is only included in the stream package if it is a descendent of the specified snapshot. To receive the stream, the origin dataset must exist in the destination storage pool. To create a replication stream package, you use the `zfs send -R` command.
- Recursive stream package, which consists of the dataset you have specified and its descendents. Unlike in a replication stream package, intermediate snapshots are not included unless they are the origin of a cloned dataset that is included in the stream. By default, if the origin of a dataset is not a descendent of the specified snapshot specified, the behavior is similar to that of a replication stream package. To create a recursive stream package, you use the `zfs send -r` command.

Receiving a ZFS Snapshot Stream

Keep the following key points in mind when you receive a file system snapshot stream:

- 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 file system that is sent to the remote system must not have the same name as a file system already on the remote system.
- If a conflicting file system name exists, you can use `zfs rename` to rename the file system.

Advantages and Disadvantages of Using ZFS Snapshot Streams for Backup and Recovery

- Advantages
  - Can store data on random access storage as long as you pipe the `zfs send` output through the `ssh` command directly into the `zfs receive` command
  - Can use incremental snapshot streams for faster performance
  - Can send and receive a file system's property settings
- Disadvantages
  - Cannot exclude directories or files from a ZFS send stream
  - Cannot select individual files or directories to receive
  - Cannot restore individual files or directories from a ZFS send stream
  - Cannot recover data in a ZFS send stream that has been stored on a file that has been corrupted

There are several advantages and disadvantages to using ZFS snapshot streams for backup and recovery. First, let's look at the advantages.

If you have random access storage (not tape) to receive onto, and you do not need to exclude anything, then `zfs send` and `zfs receive` can be used to store data, provided that you pipe the `zfs send` output through the `ssh` command directly into the `zfs receive` command.

You can create an incremental snapshot stream, which is generally much faster than incremental backups performed by file-level tools, such as `tar` and `rsync`. This is because ZFS already knows which blocks have changed on disk, and it can simply read those blocks as large sequential disk read operations.

Another advantage of using a ZFS send stream over a file-based backup tool is that you can send and receive a file system's property settings.

Now let's look at the disadvantages of using ZFS snapshot streams for backup and recovery.

First, you cannot exclude directories or files from a ZFS send stream.

You cannot select individual files or directories to receive because the `zfs receive` operation is an all-or-nothing event. You can get all of a file system snapshot or none of it.

You cannot restore individual files or directories from a ZFS send stream, although you can copy files or directories from a snapshot. This limitation means enterprise backup solutions and other archive tools, such as `cp`, `tar`, `rsync`, `pax`, `cpio`, are more appropriate for tape backup/restore, because you can restore individual files or directories.

If you store a ZFS send stream on a file or on tape, and that file becomes corrupted, you will not be able to receive it, and you will not be able to recover any of the data.

File System Backup and Recovery with Snapshots

- You can use ZFS snapshots to:
  - Take backups of your local file systems (`zfs snapshot` or `zfs snapshot -r`)
  - Recover a file system by rolling back a snapshot (`zfs rollback`)
- To view snapshots, you must enable the `listsnapshots` pool property (`zpool listsnapshots=on` *`pool`*).
- When you roll back a snapshot:
  - All changes made to a file system since a specific snapshot was created are discarded.
  - All intermediate snapshots and their clones must be destroyed (`zfs rollback -rR`).
  - The file system that you want to roll back must be unmounted.

If you do not have a business requirement to store your data remotely for disaster recovery purposes, you can take snapshots of your local file systems as backups and then, if necessary, recover them using the ZFS snapshot rollback feature.

You can use the `zfs snapshot` command to create a snapshot, which takes as its only argument the name of the snapshot that you want to create. The snapshot name must be in one of these two formats: *`filesystem@snapname`* or *`volume@snapname`*. You can also create a recursive snapshot by using the `-r` option. To view the snapshots that you have created, you must enable the `zpool listsnapshots` property for the pool in which the file systems reside. This property is disabled by default.

The `zfs rollback` command discards all changes made to a file system since a specific snapshot was created. The file system reverts to its state at the time the snapshot was taken. By default, the command cannot roll back to a snapshot other than the most recent snapshot. For the rollback to occur, all intermediate snapshots must be destroyed. You can destroy earlier snapshots by specifying the `-r` option. If clones of any intermediate snapshots exist, you must destroy them as well by specifying the `-R` option.

Note: The `-rR` options do not recursively destroy the child snapshots of a recursive snapshot. Only the top-level recursive snapshot is destroyed by either of these options. To completely roll back a recursive snapshot, you must roll back the individual child snapshots.

Before you roll back a file system, you must ensure that it is unmounted. If not, the rollback fails. If necessary, you can force the file system to be unmounted by using the `zfs rollback -f` command. After the rollback has completed, you can remount the file system.

Automating ZFS Snapshots: Overview

- You can use Time Slider to:
  - Generate snapshots
    - Frequent – Every 15 minutes (default)
    - Hourly
    - Daily
    - Weekly
    - Monthly
  - Restore individual files or directories
- The (`com.sun:`)`auto-snapshot` ZFS property is set to `false` by default.
- Time Slider services are disabled by default.
  - `time-slider`
  - `time-slider plugins: zfs-send` and `rsync`
  - `auto-snapshot`

You can automate your ZFS snapshot backups with the Time Slider service by using either the command line or GUI. With Time Slider, you can generate snapshots automatically every 15 minutes (the default setting, which is referred to as “frequent”), hourly, daily, weekly, or monthly as well as easily restore individual files or directories from the automatic snapshots.

Time Slider is supported by the `com.sun:auto-snapshot` ZFS property and several SMF services (`time-slider, time-slider plugins: zfs-send` and `rsync`, and `auto-snapshot`) that are disabled by default. After you have set the `auto-snapshot` property to true and have enabled the services, you can access Time Slider. If you are using the GUI, you access Time Slider from the GNOME file manager.

Note: You are shown how to use Time Slider in the section on backing up ZFS data using snapshots.

Automating ZFS Snapshots: Overview

- A certain number of automatic snapshots are kept according to type (frequent, hourly, daily, etc.) as long as space permits.

```
<instance name='hourly' enabled='true'>
<property_group name='zfs' type='application'>
<propval name='interval' type='astring' value='hours'/>
<propval name='keep' type='astring' value='23'/>
<propval name='period' type='astring' value='1'/>
</property_group>
<property_group name='general' type='framework'>
<property name='action_authorization' type='astring'/>
<property name='value_authorization' type='astring'/>
</property_group>
</instance>
```


In accordance with the service manifest (`/var/svc/manifest/system/filesystem/auto-snapshot.xml`), by default, the `auto-snapshot` service keeps three frequent, 23 hourly, six daily, four weekly, and 12 monthly snapshots. Snapshots are deleted, however, if space is needed, with oldest snapshots being deleted first.

Note: You can change these defaults to meet the specific business requirements of your company by using the appropriate commands and their options. In the example, you can see the automated snapshot default settings for the hourly frequency.

Clone Creation and Promotion: Overview

- You can use ZFS clones to:
  - Create a writeable and editable copy of a file system or volume from a snapshot (`zfs clone`)
  - Replace an active ZFS file system with a clone of that file system (`zfs promote`).
- When you create a clone:
  - An implicit dependency is created between the clone and snapshot
  - The clone does not inherit the properties of the dataset
- With the clone promotion process, you can:
  - Clone and replace file systems so that the original file system becomes the clone of the newly created file system
  - Destroy the file system from which the clone was originally created

If you need to modify production data but do not want to modify the data directly, you can use the ZFS clone technology to create a writable and editable copy of a file system or volume from a snapshot by using the `zfs clone` command. Clones can only be created from a snapshot. You can place the new file system or volume anywhere in the ZFS hierarchy. The new dataset is the same type (for example, file system or volume) as the snapshot from which the clone was created. You cannot, however, create a clone of a file system in a pool that is different from where the original file system snapshot resides.

After you modify the data and are sure that you want to replace the active file system with the clone, you can promote the clone by using the `zfs promote` command. (Under normal circumstances, you would not use this method. Instead, you would introduce the modifications through databases and other tools.)

When you clone a snapshot, an implicit dependency is created between the clone and snapshot. Even though the clone is created somewhere else in the file system hierarchy, the original snapshot cannot be destroyed as long as the clone exists. A clone does not inherit the properties of the dataset from which it was created. To view and change the properties of a cloned dataset, use the `zfs get` and `zfs set` commands.

Note: Initially, a clone's use property value is zero because at the start a clone shares all its disk space with the original snapshot. However, as you make changes to the clone, it uses more disk space.

ZFS uses a process called clone replacement (or promotion), which enables you to replace an active ZFS file system with a clone of that file system. This process facilitates the ability to clone and replace file systems so that the original file system becomes the clone of the newly created file system. In addition, this process makes it possible to destroy the file system from which the clone was originally created.

Note: Without clone promotion, you cannot destroy the original file system of active clones.

Quiz

If you have a business requirement to back up and restore snapshots for disaster recovery purposes, which method of snapshotting should you use?

a. Frequent snapshot backups saved locally
b. Full snapshot streams sent to a remote system
c. Frequent incremental snapshots sent to a remote system
d. Monthly snapshot backups saved locally and rolled back every three months

Answer: b

Quiz

You can take a snapshot of a ZFS file system manually or you can schedule an automatic snapshot of a ZFS file system using Time Slider.

a. True
b. False

Answer: a

Lesson Agenda

- Implementing a Plan for Backing Up and Restoring ZFS Data
- Backing Up ZFS Data by Using Snapshots
- Backing Up and Restoring ZFS Data Remotely

Backing Up ZFS Data by Using Snapshots

This section covers the following topics:

- Using snapshots for backup and recovery
- Differentiating between snapshots
- Creating and promoting a clone
- Using automatic snapshots
  - Enabling and scheduling automatic snapshots using the command line and the GUI
  - Modifying automatic snapshots scheduling using the command line and the GUI
  - Deleting unwanted automatic snapshots using the command line and the GUI
  - Browsing and recovering automatic snapshots using the GUI

Using Snapshots for Backup and Recovery

1. Use `zpool get listsnapshots` *`poolname`* to determine a pool's snapshot property setting.
2. Use `zpool set listsnapshots=on` *`poolname`* to enable a pool's snapshot property.
3. Use `zpool get listsnapshots` *`poolname`* to verify that a pool's snapshot property is enabled.
4. Use `zfs snapshot` *`filesystem@snapshot`* to create a snapshot of a file system.
5. Use `zfs list -r` *`filesystem`* to view the snapshots.
6. Use `zfs rollback` *`filesystem@snapshot`* to recover a file system, if necessary.

As we discussed in the planning topic, you can use snapshots to back up your file systems and, if necessary, recover a file system using the `zfs rollback` command. The steps for enabling the `zpool listsnapshots` property and using snapshots for file system backup and recovery are presented in the slide.

Notes for step 4: You can also create a recursive snapshot of a file system by using the `zfs snapshot -r` *`filesystem@snapshot`* command.

Notes for step 6: If intermediate snapshots exist for the file system, you must destroy them along with any clones of the intermediate snapshots by using the `zfs rollback -rR` *`filesystem@snapshot`* command.

Using Snapshots for Backup and Recovery: Example

```
# zpool get listsnapshots opmpool
NAME     PROPERTY       VALUE      SOURCE
opmpool  listsnapshots  off         default
# zpool set listsnapshots=on opmpool
# zpool get listsnapshots opmpool
NAME     PROPERTY       VALUE      SOURCE
opmpool  listsnapshots  on         local
# zfs snapshot -r opmpool/clients@may2012
# zfs list -r /opmpool
NAME                                        USED  AVAIL  REFER  MOUNTPOINT
opmpool                                    5.50M  1.93G    32K  /opmpool
opmpool/clients                            5.17M  1.93G  5.04M  /clients
opmpool/clients@may2012                        0      -  5.04M  -
opmpool/clients/owners                     31.5K  1.93G  31.5K  /owners
opmpool/clients/owners@may2012                 0      -  31.5K  -
opmpool/clients/renters                    31.5K  1.93G  31.5K  /renters
opmpool/clients/renters@may2012                0      -  31.5K  -
opmpool/props                              31.5K  1.93G  31.5K  /props

<opmpool/clients/owners file system becomes corrupted>

# zfs rollback opmpool/clients/owners@may2012
# cd /owners
root@server1:/owners# ls
smith  jones
```


You recall from the data management plan that you have a requirement to create monthly backups of the Outstanding Property Management Company application's `/opmpool/clients` file system, which includes owner and renter information. In this example, we show you how to complete this task for the May backup. Our first step is to check the pool's current `listsnapshot` property setting. We can see that currently the `listsnapshot` property is set to `off`. Next, we enable the `listsnapshots` property on the `opmpool` so that we will be able to display the snapshots for the file systems in the pool. Next, we take a recursive snapshot of the `clients` file system, so that each of the files within the `clients` file system has its own backup. We name the snapshot `clients@may2012`. Using the `zfs list -r` command, we verify that the snapshots have been created.

Note: Each of the snapshots is designated by `@may2012`.

Assume that the `owners` file system data has become corrupted before we have had the chance to create the June backup. In the example, we recover the data by rolling back the `owners@may2012` snapshot. We then verify that the data is there, and it is.

Differentiating Between Two Snapshots

Use `zfs diff` *`snapshot snapshot`* to determine the differences between two snapshots.

```
# zfs list -t snapshot -o name,creation | grep renters@may
opmpool/clients/renters@may2012                    Thu May 31 10:03 2012
# zfs list -t snapshot -o name,creation | grep renters@june
opmpool/clients/renters@june2012                   Sat Jun 30 10:15 2012
# zfs diff opmpool/clients/renters@may2012 \
opmpool/clients/renters@june2012
M /renters/
+ /renters/irving
# ls -l /renters/irving
-rw-r--r--   1 root     root          10 June 12 10:11 /renters/irving
```


To determine the differences between ZFS snapshots, you can use the `zfs diff` command. The output of this command provides a high-level description of the differences between a snapshot and a descendant dataset. The descendant can be either a snapshot of the dataset or the current dataset. For each file that has undergone a change between the original snapshot and the descendant, the type of change is described along with the name of the file. For a rename, both the old and new names are shown. The type of change follows any time stamp displayed and is described with a single character. The definition of each of these characters is provided in the next slide.

In the example, we demonstrate how to satisfy the data management planning requirement to provide information about the May and the June `renters` file system differences to the Outstanding Property Management Company application analyst. First, we determine when the snapshots were created. We can see that the `renters@may2012` was created at the end of May and `renters@june2012` snapshot was created at the end of June. We then run the `zfs diff` command against these two snapshots. The M in the `zfs diff` command output tells us that the `renters` file system has been modified, and the + indicates that a new file called `irving` was added to the `renters` file system. We run the `ls -l` command for the `irving` file to see when it was created. We can now report to the application analyst that one renter location named `irving` was added in the month of June.

ZFS Snapshot Difference Identifiers

File or Directory Change	Identifier
File or directory is modified or file or directory link has changed.	M
File or directory is present in the older snapshot but not in the newer snapshot.	-
File or directory is present in the newer snapshot but not in the older snapshot.	+
File or directory is renamed.	R

The table in this slide summarizes the file or directory changes that are identified by the `zfs diff` command. For more information about the `zfs diff` command and the options that can be used with it, see `zfs`(1M).

Creating and Promoting a Clone

1. Use `zfs clone` *`filesystem@snapshot clone`* to create a clone of a snapshot.
2. Use `zfs promote` *`clone`* to replace a file system with a clone.
3. Use `zfs rename` *`filesystem`* to rename the original file system and make it a backup.
4. Use `zfs rename` *`clone`* to rename the clone to the file system's original name.

ORACLE

As we discussed in the planning topic, there may be times when you need to clone a snapshot to modify a file system with the intention of replacing the file system with the clone. The steps for creating and promoting a clone are presented in the slide.

Notes for step 3: Another option is to remove the original file system by using the `zfs destroy` command.

Creating and Promoting a Clone: Example

```
# zfs clone opmpool/clients/renters@june2012 opmpool/clients/rentersclone
# zfs list -r /opmpool | grep renters
opmpool/clients/renters                       52K  1.93G    32K  /renters
opmpool/clients/renters@may2012               19K      -  31.5K  -
opmpool/clients/renters@june2012               1K      -    32K  -
opmpool/clients/rentersclone                  18K  1.93G    32K  /clients/rentersclone

<Analyst makes modifications to rentersclone.>

# zfs promote opmpool/clients/rentersclone
# zfs list name,used,refer -r /opmpool | grep renters
opmpool/clients/renters                        1K    32K
opmpool/clients/rentersclone                71.5K  33.5K
opmpool/clients/rentersclone@may2012          19K  31.5K
opmpool/clients/rentersclone@june2012         19K    32K
# zfs rename opmpool/clients/renters opmpool/clients/rentersbkup
# zfs rename opmpool/clients/rentersclone opmpool/clients/renters
# zfs list -o name,used,refer,mountpoint -r /opmpool | grep renters
opmpool/clients/renters                     71.5K  33.5K  /clients/renters
opmpool/clients/renters@may2012               19K  31.5K  -
opmpool/clients/renters@june2012              19K    32K  -
opmpool/clients/rentersbkup                   18K    32K  /renters
```


According to the data management plan, you have a requirement to create a clone of the `renters@june2012` snapshot. It seems that the Outstanding Property Management Company application analyst wants to use the clone of this snapshot as a baseline for adding information to the `renters` file system that is missing in the active version of the file system. After he is done modifying the file, the analyst wants you to promote the clone to production to overlay the original file system. In the example, we show you the steps that you take to accomplish these tasks. Assume that we have already created the `renters@june2012` snapshot.

Our first step is to create the clone. We name the clone `rentersclone`. We then verify that the clone has been created. Assume that the analyst has made his modifications to the file system. We promote `rentersclone` and then verify that the clone has replaced the original file system—the amount of disk space that the clone is now using indicates that it has. Our final step is to rename both the original file system and clone. We name the original file system `rentersbackup` and the clone `renters` to match the name of the original file system. After we have renamed the file system and the clone, we verify the changes that we have made.

Enabling and Scheduling Automatic Snapshots by Using the Command Line

1. Install the GNOME packages (if needed for text-install).
2. Use `zfs set com.sun:auto-snapshot=true` *`filename`* to set the `auto-snapshot` property to `true`.
3. Use `zfs get com.sun:auto-snapshot` *`filename`* to verify the `auto-snapshot` property setting.
4. Use `svcadm enable system/filesystem/zfs/auto-snapshot;`*`service`* to enable the `auto-snapshot` and `time-slider` services.
5. User `svcs – a |` *`service`* to verify that `auto-snapshot` and `time-slider` services are enabled.
6. Use `zfs list -t snapshots | grep` *`filesystem`* to confirm that the automatic snapshots are created.

As you learned in the planning topic, you can use the Time Slider service to automate ZFS snapshots. The steps for setting up the Time Slider service and for scheduling the automated snapshots from the command line are presented in the slide.

Notes for step 4: The `auto-snapshot` and `time-slider/plugin` services depend on the `time-slider` service; therefore, you should enable the `time-slider` service last (or restart it). If the GNOME packages are not installed when you enable the Time Slider services, they will go into maintenance mode.

Notes for step 5: The snapshots for the file system are created automatically in accordance with the default settings specified in the `auto-snapshot.xml` file (that is, every 15 minutes, hourly, daily, weekly, and monthly).

Notes for step 6: Automatic snapshot names use the ZFS dataset name followed by `@zfs-auto-snap <type>`, where type is `frequent`, `hourly`, and so on, along with a date and time stamp.

Modifying Automatic Snapshots Scheduling by Using the Command Line

- You can take snapshots under a given schedule for a dataset by specifying a frequency in the `auto-snapshot` property.

```
# zfs set com.sun:auto-snapshot=false mfrs/fplusers
# zfs set com.sun:auto-snapshot:weekly=true mfrs/fplusers
#
```

- You can change the frequent type setting of a given snapshot schedule by using `svccfg -s svc:/system/filesystem/zfs/auto-snapshot:frequent setprop zfs/period=`*`value`*.

```
# svcprop -p zfs/period system/filesystem/zfs/auto-snapshot:frequent
15
# svccfg -s svc:/system/filesystem/zfs/auto-snapshot:frequent setprop \
zfs/period=30
# svccfg -s svc:/system/filesystem/zfs/auto-snapshot:frequent refresh
# svcadm restart svc:/system/filesystem/zfs/auto-snapshot:frequent
#
```


There are several ways to modify the default schedule for automatic snapshots from the command line. You can change the given schedule for a dataset and all direct descendent datasets by specifying a frequency in the `auto-snapshot` property. In the first example, we set the `auto-snapshot` property to `weekly=true` for the `mfrs/fplusers` file system. This setting overrides all the other `auto-snapshot` default settings (frequent, hourly, and so on) and only takes an automatic snapshot of the `mfrs/fplusers` file system on a weekly basis.

You also have the option of modifying the period of the `frequent` type service from the default of every 15 minutes to some other minute interval. You use the `svccfg -s` command presented in the slide to perform this action. In the second example, we check the current setting and then change the period of the `frequent` service to 30 minutes. We then refresh and restart the service.

Deleting Unwanted Automatic Snapshots by Using the Command Line

To delete all unwanted automatic snapshots, use the following script:

```
# for s in `zfs list -H -o name -t snapshot | grep @zfs-auto-snap`;
do zfs destroy $s; done
#
```

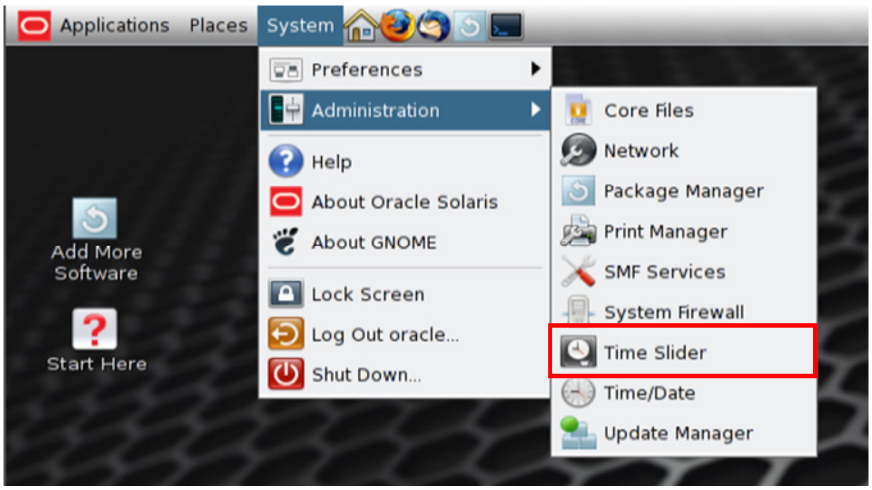
Caution: This script deletes all snapshots matching the pattern `@zfs-auto-snap`.

You can delete unwanted automatic snapshots from the command line by using the script presented in the slide. Be careful, however, when using this script, because it deletes all the snapshots matching the pattern `@zfs-auto-snap`.

Note: The script is very flexible. For example, you can modify the script to just delete the snapshots for a specific file system by indicating the partial or full name of the file system before the @ sign.

Enabling and Scheduling Automatic Snapshots by Using the GUI

1. Install the GNOME packages, if needed.
2. Select Administration > Time Slider from the file manager's System menu to enable the `time-slider` and the `auto-snapshot` services.



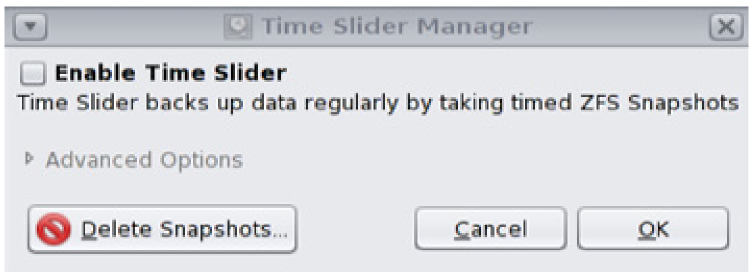
ORACLE

The steps for setting up the Time Slider service and for scheduling the automated snapshots by using the GUI are presented in the slide.

Notes for step 2: When you are using the GUI, the services are taken offline automatically when you turn off the Time Slider service.

Enabling and Scheduling Automatic Snapshots by Using the GUI

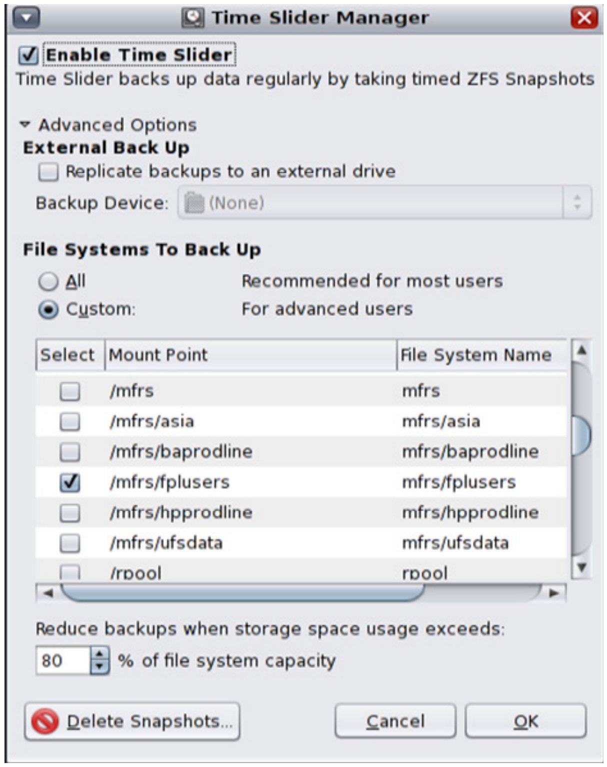
3. In the Time Slider Manager dialog box, do the following to enable the Time Slider service and select the file system to back up:
   – Select Enable Time Slider.
   – Verify that Custom, in File Systems To Back Up, under Advanced Options, is selected. This is the default option.
   – Select the file systems that you want to back up.
   – Click OK.



This step is a continuation of the task presented in the previous slide.

Notes for step 3: After you select Enable Time Slider, the Advanced Options become available (active). You can see an expanded view of the Time Slider Manager on the next slide.

Expanded View of the Time Slider Manager



This information presented on this slide is a continuation of the step presented in the previous slide.

Notes for step 3: In the File Systems To Backup Section, Custom is selected as the default to allow advanced users to select which file systems they want to back up. As part of the Advanced Options, you can choose to replicate backups to an external drive. You can also control the percentage of file system capacity for reducing backups when storage space usage is exceeded. The default is 80 percent. Based upon your storage situation, you can modify this setting accordingly. You want to make sure that the automatic snapshots are not consuming all the available storage. When the snapshots cross the 80% threshold, the system generates an error to notify you, so that you can then delete any redundant snapshots.

Enabling and Scheduling Automatic Snapshots by Using the GUI

4. Use `zfs list -t snapshots | grep` *`filesystem`* to display the automatic snapshots that have been taken.

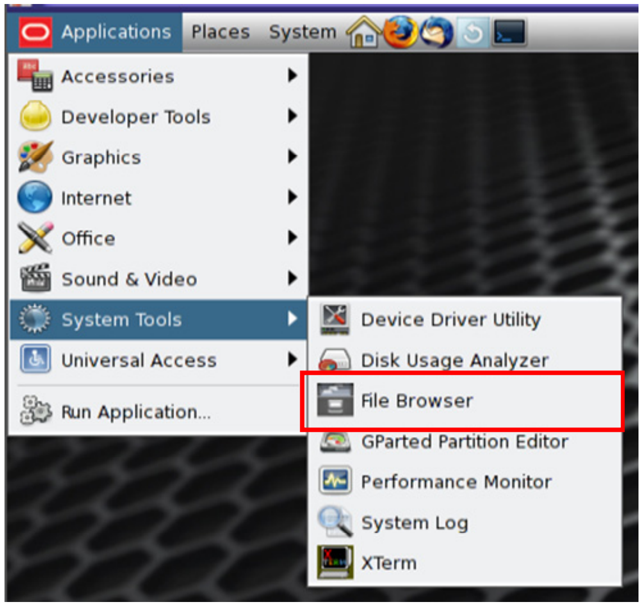
```
# zfs list -t snapshot | grep mfrs
mfrs/fplusers@zfs-auto-snap_monthly-2012-05-11-08h36     0K      -    31K  -
mfrs/fplusers@zfs-auto-snap_hourly-2012-05-11-11h36      0K      -    31K  -
mfrs/fplusers@zfs-auto-snap_frequent-2012-05-11-12h06    0K      -  31.5K  -
```


This step is a continuation of the task presented in the previous slide. In the example, we display the list of snapshots for the `mfrs` file system in a terminal window. Here we can see the automatic snapshots that have been taken for the `fplusers` file system.

Note: You might want to wait 15 minutes or so before running this command to allow the snapshots to be generated.

Browsing and Recovering Automatic Snapshots by Using the GUI

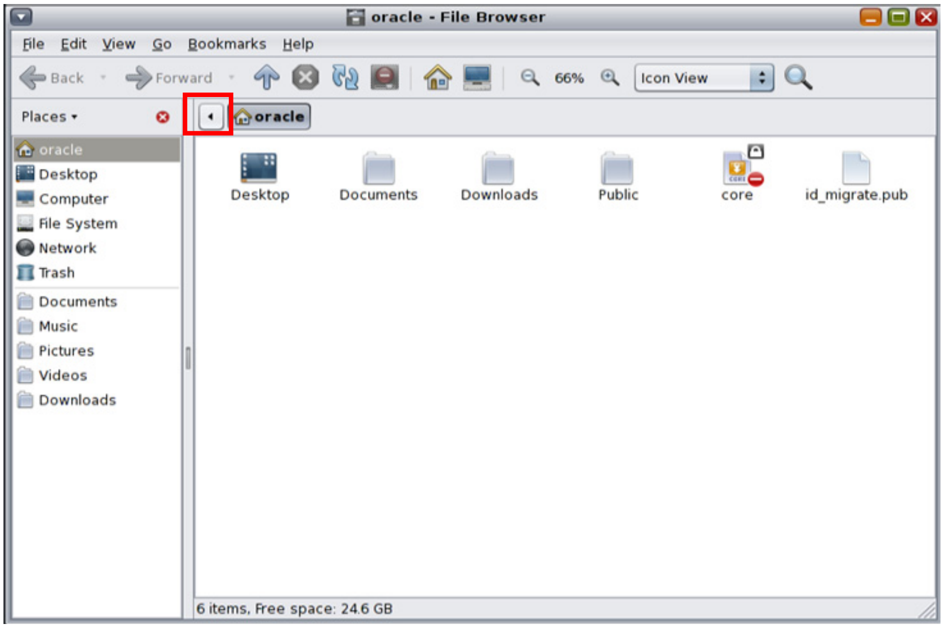
1. Select System Tools > File Browser from the file manager's Applications menu to open the Oracle – File Browser window.



You can browse and recover snapshots from the GNOME desktop file manager. The steps for how to access the snapshots are presented in this slide and the slides that follow.

Browsing and Recovering Automatic Snapshots by Using the GUI

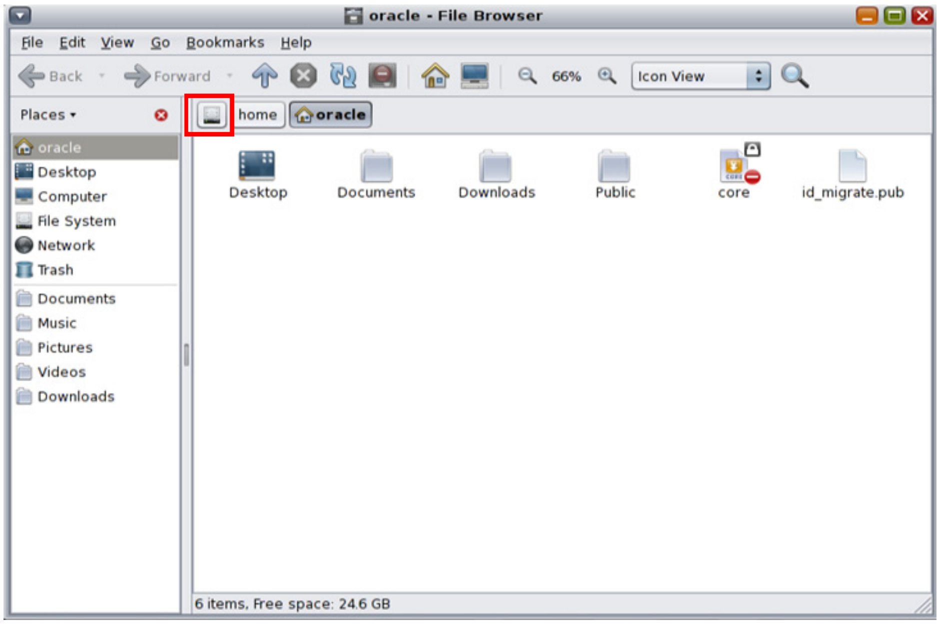
2. In the Oracle – File Browser window, click the left arrow button located next to the Oracle home button.



Notes for step 2: As you move through the directory structure, the home button changes to reflect the file system you are currently in. The name of the File Browser window changes as well.

Browsing and Recovering Automatic Snapshots by Using the GUI

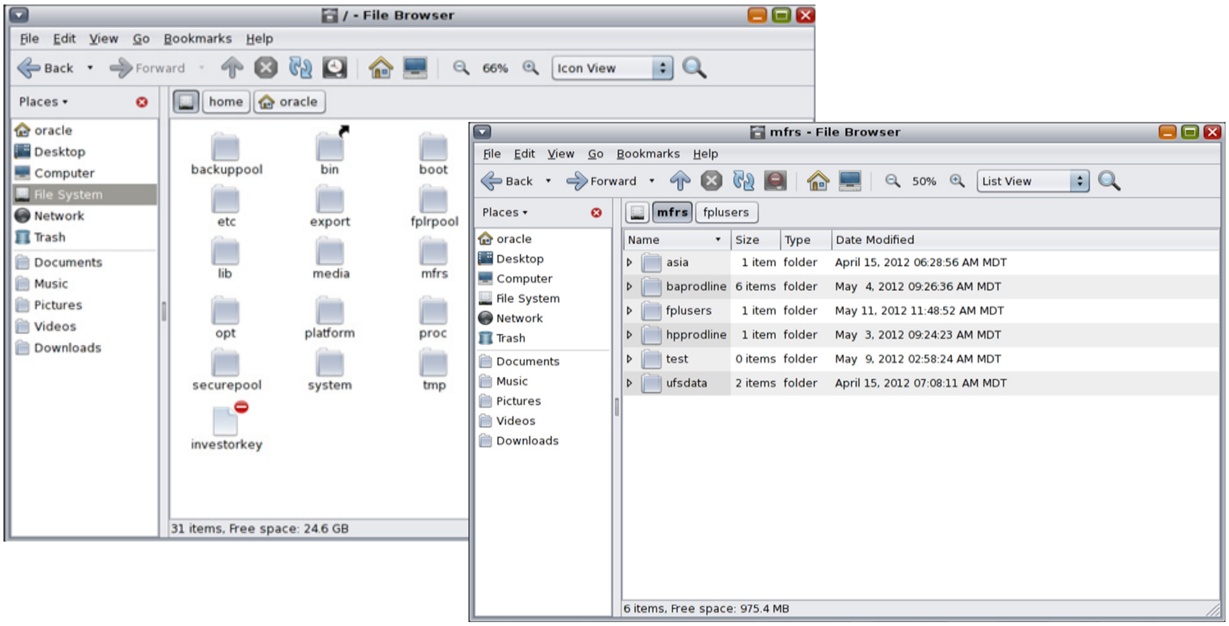
3. Click the Tree button located next to the home button to display the directory hierarchy for your system.



Notes for step 3: A list of snapshots for the file system appears between the navigation bar and the file system buttons.

Browsing and Recovering Automatic Snapshots by Using the GUI

4. Navigate to the file system that contains the snapshots that you want to recover, by double-clicking the appropriate file folders.



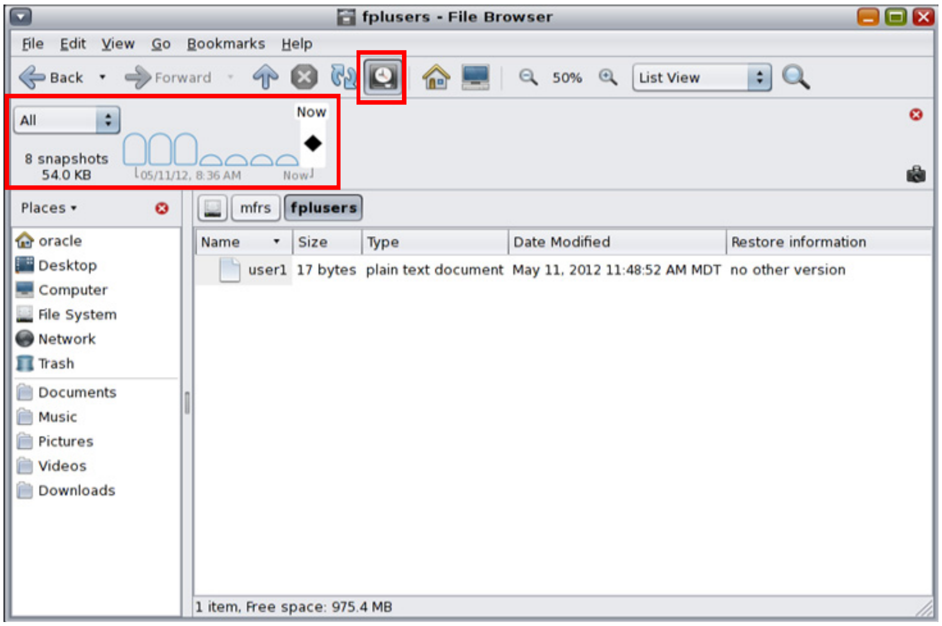
Notes for step 4: You can open all your files in read-only mode, or you can browse in list view mode (as oppose to icon view) so that a restore information column automatically appears. This column gives you contextual information about either of the following:

- The file version number, if you browse in the current or latest version of a directory.
- The difference, if any, between the file snapshot and the latest version of the file.

In the example screenshots, we select the `mfrs` file system folder in the left screenshot and then the `fplusers` file system folder in the right screenshot.

Browsing and Recovering Automatic Snapshots by Using the GUI

5. When you reach the file system that you want, click the clock icon on the toolbar to reveal the list of snapshots for this file system.

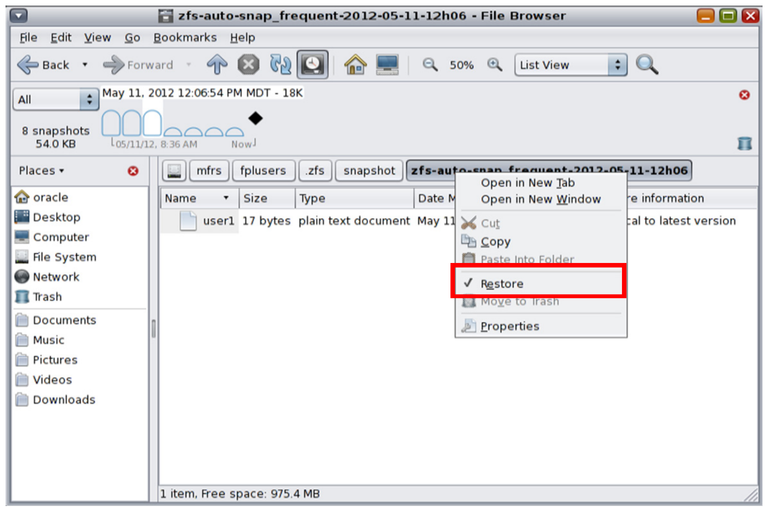


Notes for step 5: You can roll your mouse over each snapshot to see the details for that snapshot, including when the snapshot was created, the size of the snapshot, and the type of snapshot (frequent, daily, and so on).

In the example screenshot, we see that eight automatic snapshots have been taken of the `fplusers` file system.

Browsing and Recovering Automatic Snapshots by Using the GUI

6. To recover files from an automatic snapshot, select the snapshot and then do one of the following:
   – Drag and drop a snapshot into another file manager window.
   – Right-click a snapshot and select Restore. The snapshot is saved to the desktop. Then, copy and paste the selected snapshot to the present time.



ORACLE

In the example screenshot, we restore the May 11, 2012 snapshot by using the Restore menu option.

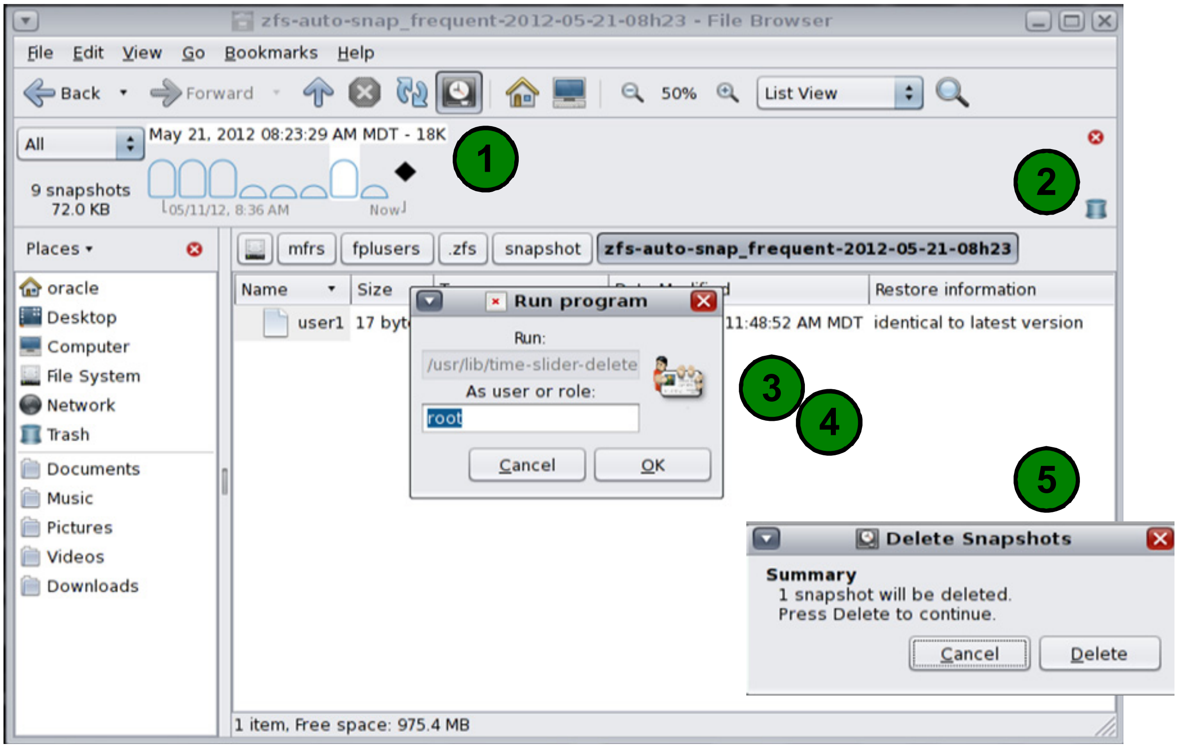
Deleting Unwanted Automatic Snapshots by Using the GUI

1. From within the File Browser, select the snapshot that you want to delete.
2. Click the “Delete this snapshot” icon.
3. When the “Run program” dialog box appears, verify user or role, and click OK.
4. Enter the password and click OK.
5. When the Delete Snapshots dialog box appears, click Delete.

You can delete unwanted automatic snapshots by using the steps presented in the slide.

Notes for step 5: The snapshot is removed from the snapshot section.

Deleting Unwanted Automatic Snapshots by Using the GUI: Example



In these example screenshots, we delete an automatic snapshot that was created for the `mfrs/fplusers` file system. To delete the snapshot, we select it from the snapshot timeline (#1). Next, we click the “Delete this snapshot” icon (#2). When the “Run program” dialog box appears, we verify our role as root and click OK (#3). When we are prompted for our password (not shown in the example), we enter our password and click OK (#4). When the Delete Snapshots dialog box appears, we click Delete to continue (#5). The snapshot is removed (not shown).

Enabling and Scheduling Automatic Snapshots by Using the Command Line: Example

```
# zfs set com.sun:auto-snapshot=true mfrs/fplusers
# zfs get com.sun:auto-snapshot mfrs/fplusers
NAME           PROPERTY               VALUE                  SOURCE
mfrs/fplusers  com.sun:auto-snapshot  true                   local
# svcadm enable system/filesystem/zfs/auto-snapshot;frequent
# svcadm enable system/filesystem/zfs/auto-snapshot:daily
# svcadm enable system/filesystem/zfs/auto-snapshot:hourly
# svcadm enable system/filesystem/zfs/auto-snapshot:weekly
# svcadm enable system/filesystem/zfs/auto-snapshot:monthly
# svcadm enable system/filesystem/zfs/time-slider
# svcs | grep auto-snapshot
online          8:36:52 svc:/system/filesystem/zfs/auto-snapshot:daily
online          8:36:52 svc:/system/filesystem/zfs/auto-snapshot:monthly
online          8:36:52 svc:/system/filesystem/zfs/auto-snapshot:weekly
online          8:36:52 svc:/system/filesystem/zfs/auto-snapshot:frequent
online          8:36:52 svc:/system/filesystem/zfs/auto-snapshot:hourly
# svcs -a | grep time-slider
disabled        8:11:10 svc:/application/time-slider/plugin:rsync
disabled        8:29:26 svc:/application/time-slider/plugin:zfs-send
online          8:36:52 svc:/application/time-slider:default
```

ORACLE

You recall from the data management plan that you have a requirement to set up automatic snapshots for the Fantastic Products LLC on the `mfrs/fplusers` file system for every 10 minutes instead of the default of every 15 minutes. In this example, we demonstrate how to complete this task. To begin, we turn the `auto-snapshot` property to `true` on the `mfrs/fplusers` file system and verify the setting. We then enable each of the `auto-snapshot` and `time-slider` services and verify that they are each online**.**

Note: When you use the GUI, these services are enabled automatically when you enable the auto snapshots (see Slide 33, step 2).

Note: The remaining steps are presented on the next slide.

Enabling and Scheduling Automatic Snapshots by Using the Command Line: Example

```
# svcprop -p zfs/period system/filesystem/zfs/auto-snapshot:frequent
15
# svccfg -s system/filesystem/zfs/auto-snapshot:frequent setprop zfs/period=10
# svccfg –s system/filesystem/zfs/auto-snapshot:frequent refresh
# svcadm restart system/filesystem/zfs/auto-snapshot:frequent
# zfs list -t snapshot | grep mfrs
mfrs/fplusers@zfs-auto-snap_monthly-2012-05-11-08h36     0K      -    31K  -
mfrs/fplusers@zfs-auto-snap_hourly-2012-05-11-11h36      0K      -    31K  -
mfrs/fplusers@zfs-auto-snap_frequent-2012-05-11-12h06    0K      -  31.5K  -
```


The steps presented here are continued from the previous slide.

Our next step is to verify that the current setting for the frequent service is still the default of every 15 minutes, which it is. Next, we change the period of the frequent service to every 10 minutes. We then refresh and restart the service.

Our final step is to display the snapshots for the `mfrs` file system. Here we can see the `frequent` snapshot that was just taken.

Note: In the practice that follows this section, you are given the opportunity to perform this same task by using the GUI.

Quiz

What does the output from the following command tell you? (Select two.)

```
# zfs diff opmpool/clients/renters@june2012
opmpool/clients/renters@july2012
M /renters/
R /renters/jrewing -> /renters/bewing
```

a. The `renters` file system has been modified.
b. A file or directory is present in the `renters@june2012` snapshot but not in the more recent snapshot.
c. A file or directory is present in the `renters@july2012` snapshot but not in the older snapshot.
d. A file in the `renters` file system has been renamed.

Answer: a, d

Quiz

Which statements about using automatic snapshots for a file system are true? (Select two.)

a. The Time Slider services must be disabled.
b. The `auto-snapshot` property for the file system must be set to `true`.
c. A file system can have only one frequency type set for automatic snapshots: frequent, hourly, daily, or monthly.
d. The default period for the `frequent` service is every 15 minutes.

Answer: b, d

Practice 9-1 Overview: Backing Up ZFS Data by Using Snapshots

This practice covers the following topics:

- Using snapshots for backup and recovery
- Differentiating between two snapshots
- Creating and promoting clones
- Using automatic snapshots

This practice should take about 45 minutes to complete.

Lesson Agenda

- Implementing a Plan for Backing Up and Restoring ZFS Data
- Backing Up ZFS Data by Using Snapshots
- Backing Up and Restoring ZFS Data Remotely

Backing Up and Restoring ZFS Data Remotely

This section covers the following topics:

- Sending and receiving a ZFS snapshot stream
- Applying ZFS properties to a ZFS snapshot stream
- Using a ZFS snapshot recovery archive

Sending and Receiving a ZFS Snapshot Stream

1. Configure Secure Shell (`ssh`).
   a. Create authentication keys for user `oracle`.
   b. Assign user `oracle` the ZFS File System profile for managing the ZFS file system.
2. Create a backup pool on the remote system.
3. Use `zfs send` *`filesystem@snapshot`* `| ssh` *`remotesystem`* `zfs recv -F` *`filesystem`* to send a full snapshot stream to a remote system.
4. Verify that the snapshot stream has been received on the remote system.

ORACLE

As we discussed in the planning topic, there may be times when you need to send snapshots of file systems to a remote system for disaster recovery purposes. You use ZFS snapshot streams for this type of requirement. The steps for sending and receiving snapshot streams are presented in the slide.

Notes for step 1: The root user is not available for remote `ssh` because root is a role. Consequently, you must use another user (such as `oracle`) for `ssh` authentication with a remote host. After configuring `ssh`, the user `oracle` is permitted to use the send and receive utilities.

Note: You are shown step by step how to configure `ssh` in the practice.

Notes to step 3: Sending the data to a remote system and receiving simultaneously is considered an efficient and highly recommended practice. If you make a change to the destination file system and you want to perform another incremental send of a snapshot, you must first roll back the receiving file system. However, you can use the `-F` option for the `zfs recv` command to eliminate the need to roll back the receiving file system to receive the snapshot.

Sending and Receiving a ZFS Snapshot Stream: Example

```
root@server1:~# zfs send opmpool/clients/renters@june2012 | ssh -l \
oracle -i ~/id_migrate client1 pfexec /usr/sbin/zfs recv -F \
backuppool/renters

root@client1:~# zpool set listsnapshots=on backuppool
root@client1:~# zfs list -r /backuppool
NAME                          USED  AVAIL  REFER  MOUNTPOINT
backuppool                    130K   976M    32K  /backuppool
backuppool/renters             32K   976M    32K  /backuppool/renters
backuppool/renters@june2012      0      -    32K  -
```


You recall from the data management plan that you have a requirement to back up the Outstanding Property Management Company `/opmpool/clients/renters` file system information on a remote system by using the `renters@june2012` snapshot. In this example, we demonstrate how to address this requirement. We send and receive the ZFS snapshot stream by using Secure Shell (`ssh`). Assume that we have already configured `ssh` and have created a pool on the remote system that we have called `backuppool`. Our remote system is called `client1`. Our first step is to send the `renters@june2012` snapshot to `client1`. As we discussed in the planning topic, we pipe the `zfs send` output through `ssh` directly into the `zfs receive` command. Note also the use of the `oracle` account and the privilege information. Because we are accessing a remote system and we are dealing with root as a role, we must access the remote system as the account that has root-assuming privileges, which in our case is `oracle`.

Next, on the remote system, we ensure that the `listsnapshots` property is set to `on` so that we can see the snapshots. Then we verify that we have received the file system—and we have.

Applying ZFS Properties to a ZFS Snapshot Stream: Overview

- You can send and receive the same file system property values to re-create the original file system.
- You can disable a file system property when a snapshot stream is received.
- You can specify a different property value when the snapshot stream is received.

You can send and receive property values in a ZFS snapshot stream. You can specify that the original property value be used when the snapshot stream is received to re-create the original file system.

Note: Not all of the file system properties in a send stream might apply to the receiving file system or local file system properties. This could interfere with your ability to restore the file system.

You can disable a file system property when the snapshot stream is received. You can also specify a different local property value to be applied to the file system when the snapshot stream is received.

Options for Applying ZFS Properties to ZFS Snapshot Streams

Action	Command Option
Generate a replication stream package that, when received, will preserve all properties, snapshots, descendent file systems, and clones.	`zfs send -R`
Generate a recursive stream package that, when received, preserves all properties and descendent file systems. Does not preserve intermediate snapshots.	`zfs send -r`
Include properties in the send stream without the -R option.	`zfs send -p`
Disable a property when it is received.	`zfs receive -x`
Restore the original snapshot properties for recovery purposes.	`zfs send -b`
Mask the received property value with the inherited property value.	`zfs inherit`
Revert a local property value to the received value.	`zfs inherit -S`
Use the last element of the sent snapshot name to determine the new snapshot name.	`zfs send -e`
Include the new non-default RECEIVED column.	`zfs get -o`
Include all columns, including RECEIVED.	`zfs get -o all`

ORACLE

The table presented in this slide contains some of the command options you can use for applying ZFS properties to ZFS snapshot streams.

Note: In the practice that follows this topic, you have the opportunity to apply ZFS properties to ZFS snapshot streams and see the results.

Using a ZFS Snapshot Recovery Archive

To create a ZFS snapshot archive:

1. Sign on to the local system.
2. Configure Secure Shell.
3. Create a backup pool on the remote system.
4. Create the ZFS snapshot archive.
5. Use `zfs send -Rv` *`filesystem@snapshot`* `| ssh` *`remotesystem`* `zfs recv -F` *`pool`* to send a ZFS snapshot archive to a remote system.
6. Verify that the snapshot archive has been received on the remote system.

A ZFS snapshot recovery archive is another name for the ZFS snapshot stream that you send to a remote system to have as a backup in case of system failure or data corruption. A ZFS snapshot archive can contain snapshot streams of the entire root pool, individual pools, or file systems. What you include in the archive is based on your business requirements. The steps for creating a ZFS snapshot archive and recovering data from it are presented in this slide and the next.

Notes to step 5: Generally speaking, you use a stream package format to send a ZFS snapshot recovery archive. In this step, we use the replication stream package. You might recall from the planning topic there are two types of stream packages:

- Replication stream package, which consists of the dataset that you have specified and its descendents (`zfs send -R`)
- Recursive stream package, which consists of the dataset that you have specified and its descendents (`zfs send -r`)

Using a ZFS Snapshot Recovery Archive

To recover a ZFS snapshot archive from a remote system to the local system:

1. Sign on to the remote system.
2. Configure Secure Shell.
3. Use `zfs send -Rv` *`filesystem@snapshot`* `| ssh` *`localsystem`* `zfs recv -F` *`pool`* to recover a ZFS snapshot archive on the local system.
4. Verify that the snapshot archive has been received on the local system.

The steps for recovering data from the ZFS snapshot archive from the remote system are presented in the slide.

Using a ZFS Snapshot Recovery Archive: Example

```
root@server1:~# zfs send -Rv admpool@snap1 | ssh -l oracle -i ~/id_migrate \
client1 pfexec /usr/sbin/zfs recv -F backuppool
sending from @ to admpool@snap1
sending from @ to admpool/serverapps@snap1
sending from @ to admpool/appanalysts@snap1

root@client1:~# zfs list -r /backuppool
NAME                           USED  AVAIL  REFER  MOUNTPOINT
backuppool                     368K   976M    33K  /backuppool
backuppool@snap1                  0      -    33K  -
backuppool/appanalysts          31K   976M    31K  /backuppool/appanalysts
backuppool/appanalysts@snap1      0      -    31K  -
backuppool/serverapps           32K   976M    32K  /backuppool/serverapps
backuppool/serverapps@snap1       0      -    32K  -
```


According to the data management plan, you have a requirement to create a ZFS recovery archive called `admpool` to serve as a backup for both the Outstanding Property Management Company and Fantastic Products LLC applications. In this example, we show you how to create the archive and recover data from it.

Assume that we have already configured ssh and are using the pool, called `backuppool`, that we created earlier on `client1`. We have deleted all the old files from `backuppool`. Our first step is to send the `admpool@snap1` snapshot stream package to `client1`. We send the package using a replication stream format as indicated by the `-R` option. We then verify on the remote system that we have received all of the files associated with the `admpool` snapshot archive.

Using a ZFS Snapshot Recovery Archive: Example

```
<Assume admpool has become corrupted.>

root@client1:~# zfs send -Rv backuppool@snap1 | ssh -l oracle –I \
~/id_migrate server1 pfexec /usr/sbin/zfs recv admpool
sending from @ to backuppool@snap1
sending from @ to backuppool/appanalysts@snap1
sending from @ to backuppool/serverapps@snap1

root@server1:~# zfs list -r /admpool
NAME                          USED  AVAIL  REFER  MOUNTPOINT
admpool                       183K   976M    33K  /admpool
admpool@snap1                    0      -    33K  -
admpool/appanalysts            31K   976M    31K  /admpool/appanalysts
admpool/appanalysts@snap1        0      -    31K  -
admpool/serverapps             32K   976M    32K  /admpool/serverapps
admpool/serverapps@snap1         0      -    32K  -
root@server1:~# cd /admpool/serverapps
root@server1:/admpool/serverapps# ls
fpl  opm
root@server1:/admpool/serverapps# cat fpl
fpl
```


Assume that admpool has become corrupted and we need to recover the data from the ZFS snapshot archive on the remote system. We have configured ssh behind the scenes and are ready to recover the admpool data from backuppool. Our first step is to send the backuppool@snap1 snapshot stream package to server1. As before, we send the package using a replication stream format.

On server1 after the recovery operation is complete, we verify that we have received all the files and that we can access them.

Quiz

Which statements about applying ZFS properties to snapshots streams are true? (Select two.)

a. You can send and receive the same file system property values to re-create the original file system.

b. You cannot disable a file system property when a snapshot stream is received.

c. You can specify a different property value when the snapshot stream is received.

d. You can apply ZFS properties to replication snapshot streams only.

Answer: a, c

Quiz

What command do you use to send a recursive stream package called `admpool@snap1` to `backuppool` on remote system `client1`.

a. `# zfs send -Rv backuppool@snap1 | ssh client1 zfs recv -F admpool`

b. `# zfs send -Rv admpool@snap1 | ssh client1 zfs recv -F backuppool`

c. `# zfs send -rv backuppool@snap1 | ssh client1 zfs recv -F admpool`

d. `# zfs send -rv admpool@snap1 | ssh client1 zfs recv -F backuppool`

ORACLE

Answer: d

Practice 9-2 Overview: Backing Up ZFS Data Remotely

This practice covers the following topics:

- Backing up a ZFS snapshot remotely
- Using a recovery archive

This practice should take about 50 minutes to complete.

Summary

In this lesson, you should have learned how to:

- Implement a plan for backing up and restoring ZFS data
- Back up ZFS data by using snapshots
- Back up and restore ZFS data remotely

10

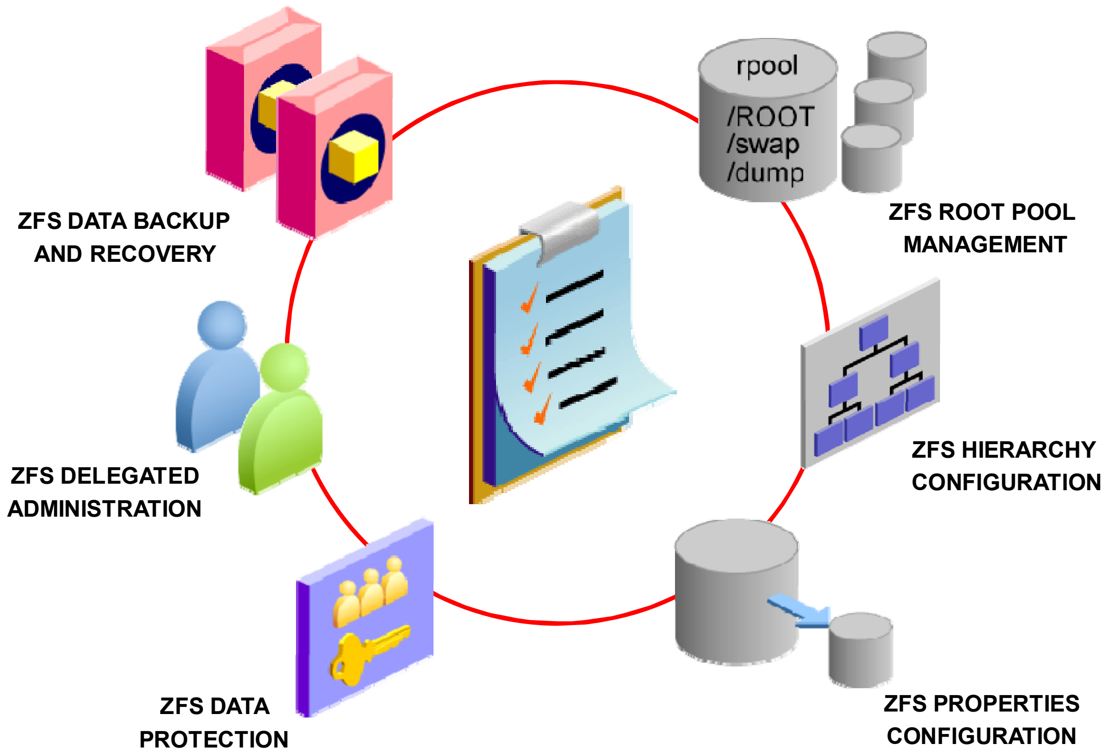
ZFS Data Management Challenge

Objectives

After completing this lesson, you should be able to:

- Create a ZFS infrastructure for a business application
- Configure ZFS pool and file system properties
- Manage ZFS properties within a non-global zone
- Migrate ZFS data
- Protect data by using ZFS ACLs
- Use ZFS delegated administration to grant permissions
- Configure a ZFS rights profile
- Back up and recover ZFS data by using ZFS snapshots and ZFS snapshot streams

Planning Workflow: Orientation



You have completed all the ZFS data management tasks that you were assigned as part of the data management plan. You should now be proficient in each of these tasks, but are you? In this lesson, which consists solely of a practice and no lecture, you are given the opportunity to prove to yourself what you have learned.

Reviewing the Requirements for the ZFS Data Management Challenge

Outstanding Property Management Company	Fantastic Products LLC
1. Create the OPM ZFS infrastructure.	1. Create the FPL ZFS infrastructure.
2. Automate device replacement on the OPM pool.	2. Control the management of a dataset from a non-global zone.
3. Create and assign a ZFS administration role.	3. Migrate data from one system to another.
	4. Set the number of copies for data in a non-global zone.
	5. Save multiple copies of data and take steps to minimize storage usage costs.
	6. Use ACLs to restrict user access to data.
	7. Back up data to and recover data from a remote system.

System
1. Create and encrypt a central file system to house application passwords.
2. Set storage and reservation quotas.

The table in this slide captures the high-level requirements that you will need to meet to successfully complete the ZFS Data Management Challenge. The details for each requirement are presented in the activity guide for this lesson.

Practice 10-1 Overview: ZFS Data Management Challenge

This practice covers the following topics:

- Creating a ZFS infrastructure for a business application
- Configuring ZFS properties
- Migrating ZFS data
- Using ZFS ACLs
- Using ZFS delegated administration
- Backing up and recovering ZFS data remotely

Note: This practice contains no step-by-step guidance. You are encouraged to use previous lesson materials as reference.

This practice should take about 2.5 hours to complete. Unlike previous practices, this practice contains no step-by-step guidance. If you need help remembering how to complete a specific task, you are encouraged to refer to previous lesson materials.

Summary

In this lesson, you should have learned how to:

- Create a ZFS infrastructure for a business application
- Configure ZFS pool and file system properties
- Managing ZFS properties within a non-global zone
- Migrate ZFS data
- Protect data by using ZFS ACLs
- Use ZFS delegated administration to grant permissions
- Configure a ZFS rights profile
- Back up and recover ZFS data by using ZFS snapshots and ZFS snapshot streams