Oracle Solaris Cluster 4.x Administration

Student Guide - Volume I

D74942GC10
Edition 1.0
February 2012
D75861

ORACLE

Authors

Raghavendra JS
Zeeshan Nofil
Venu Poddar

Technical Contributors and Reviewers

Thorsten Fruauf
Harish Mallya
Hemachandran
Namachivayam
Pramod Rao
Tim Read
Sambit Nayak
Venugopal Navilugon
Shreedhar
Lisa Shepherd
Mahesh Subramanya
Venkateswarlu Tella

Graphic Designer

Satish Bettegowda

Editors

Raj Kumar
Richard Wallis

Publishers

Michael Sebastian
Giri Venugopal

Contents

Preface

Profile

Before You Begin This Course

Before you begin this course, you should be able to:

- Administer the Oracle Solaris 10/11 Operating System
- Manage file systems and local disk drives
- Perform system boot procedures
- Manage user and role administration

How This Course Is Organized

Oracle Solaris Cluster 4.x Administration is an instructor-led course featuring lectures and hands-on practices. Online demonstrations and written practice sessions reinforce the concepts and skills that are introduced.

Related Publications

Oracle Publications

Title

Oracle Solaris Cluster 4.0 Information Library

http://docs.oracle.com/cd/E23623_01/index.html

Oracle Solaris 11 Information Library

http://docs.oracle.com/cd/E23824_01/

Related Publications

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."

Typographic Conventions (continued)

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

- Goals
- Agenda
- Introductions
- Your learning center

In this course, you learn the essential information and skills needed to install and administer Oracle Solaris Cluster hardware and software systems. To begin, we would like to take about 20 minutes to give you an introduction to the course as well as to your fellow students and the classroom environment.

Course Objectives

After completing this course, you should be able to:

- Describe the major Oracle Solaris Cluster hardware and software components and functions
- Configure access to node consoles and the cluster console software
- Install and configure the Oracle Solaris Cluster software
- Configure Oracle Solaris Cluster quorum devices and device fencing
- Configure Solaris Volume Manager software in the Oracle Solaris Cluster software environment

Course Objectives

After completing this course, you should be able to:

- Configure and use ZFS in the Oracle Solaris Cluster software environment
- Create Internet Protocol Multipathing (IPMP) failover groups in the Oracle Solaris Cluster environment
- Describe resources and resource groups, configure Network File System (NFS) as a failover data service, and configure an Apache as a scalable data service
- Configure HA for zones and zone clusters

Agenda: Day 1

- Lesson 1: Introduction
- Lesson 2: Planning the Oracle Solaris Cluster Environment
  - Define clustering.
  - Describe the Oracle Solaris Cluster features.
  - Identify the Oracle Solaris Cluster hardware and software environment.
  - Identify the Oracle Solaris Cluster-supported applications.
  - Identify the Oracle Solaris Cluster software HA framework.
  - Identify the global storage services.
  - Identify the virtualization support in Oracle Solaris Cluster.

Agenda: Day 1

- Lesson 3: Establishing Cluster Node Console Connectivity
  - Describe the different methods for accessing a console.
  - Overview of Oracle Solaris parallel console software.
  - Installing the `pconsole` utility.
  - Using the `pconsole` utility.

Agenda: Day 1

- Lesson 4: Preparing for the Oracle Solaris Cluster Installation
  - Prepare the Oracle Solaris operating system environment.
  - Configure the Oracle Solaris Cluster storage connections.
  - Describe the quorum votes and quorum devices.
  - Describe persistent quorum reservations and cluster amnesia.
  - Describe data fencing.
  - Configure a cluster interconnect.
  - Identify public network adapters.
  - Configure shared physical adapters.

Agenda: Day 2

- Lesson 5: Installing and Configuring the Oracle Solaris Cluster Software
  - Identify the cluster installation package groups.
  - List the prerequisites to install the Oracle Solaris Cluster software.
  - Install the Oracle Solaris Cluster software.
  - Set the root environment.
  - Configure the Oracle Solaris Cluster software.
  - Describe sample cluster configuration scenarios.
  - Identify settings automatically configured by `scinstall`.
  - Perform automatic quorum configuration.
  - Describe manual quorum selection.
  - Perform post-installation verification.

Agenda: Day 2

- Lesson 6: Performing Oracle Solaris Cluster Administration
  - Identify the cluster daemons.
  - Use cluster commands.
  - Use RBAC with Oracle Solaris Cluster.
  - Administer cluster global properties.
  - Administer cluster nodes.
  - Administer a quorum.
  - Administer disk path monitoring.
  - Administer SCSI protocol settings for storage devices.
  - Administer interconnect components.
  - Use the `clsetup` command.
  - Perform cluster operations.
  - Modify private network settings while in non-cluster mode.

Agenda: Day 3

- Lesson 7: Using ZFS with Oracle Solaris Cluster Software
  - Build ZFS storage pools and file systems.
  - Use ZFS for Oracle Solaris Cluster failover data.
- Lesson 8: Using Solaris Volume Manager with Oracle Solaris Cluster Software
  - Provide an overview of Solaris Volume Manager.
  - Provide an overview of shared disk sets.
  - Describe Solaris Volume Manager multi-owner disk set.
  - Describe creating and managing shared disks.
  - Use Solaris Volume Manager status commands.
  - Perform Oracle Solaris Cluster device group management.
  - Create cluster file systems.

Agenda: Day 3

- Lesson 9: Managing the Public Network with IPMP
  - Define the purpose of IPMP.
  - Define the concepts of an IPMP group.
  - List examples of network adapters in IPMP groups on a single Oracle Solaris OS server.
  - Describe the operation of the `in.mpathd` daemon.
  - List the options to the `ipadm` command that support IPMP.
  - Configure IPMP by using `ipadm` commands.
  - Perform a forced failover of an adapter in an IPMP group.
  - Describe the integration of IPMP into the Oracle Solaris Cluster software environment.

Agenda: Day 4

- Lesson 10: Managing Data Services, Resource Groups, and HA-NFS
  - Introduce data services in the cluster.
  - List the components of a data service agent.
  - Introduce data service packaging, installation, and registration.
  - Introduce resources, resource groups, and the resource group manager.
  - Describe failover resource groups.
  - List resources and resource types.
  - List the guidelines for using global and failover file systems.
  - Understanding resources and resource group dependencies
  - Describe flexible load-based distribution of resource groups into nodes.
  - Configure resource groups.

Agenda: Day 4

- Lesson 11: Configuring Scalable Services and Advanced Resource Group Relationships
  - Identify scalable services and shared addresses.
  - Describe the characteristics of scalable services.
  - Describe a `SharedAddress` resource.
  - Describe the properties of resource groups and scalable groups.
  - Describe how the SharedAddress resource works with scalable services.
  - Add auxiliary nodes for a `SharedAddress` property.
  - Review command examples for a scalable service.
  - Control scalable resources and resource groups.
  - View scalable resource and group status.
  - Describe advance resource group relationships.

Agenda: Day 5

- Lesson 12: Using Oracle Solaris Zones in Oracle Solaris Cluster
  – Describe Oracle Solaris Zones in Oracle Solaris 11.
  – Describe using HA for Zones.
  – Configure a failover zone.
  – Describe zone clusters.
  – Create a zone cluster.
  – Identify cross-cluster affinities and dependencies.

Note:

- Class is from 9:00 AM to 5:00 PM each day.
- There are several short breaks throughout the day with an hour for lunch.

Introductions

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

2

Planning the Oracle Solaris Cluster Environment

Objectives

After completing this lesson, you should be able to:

- Define clustering
- Describe Oracle Solaris Cluster features
- Identify:
  - Oracle Solaris Cluster hardware environment
  - Oracle Solaris Cluster software environment
  - Oracle Solaris Cluster–supported applications
  - Oracle Solaris Cluster software HA framework
  - Global storage services
  - Virtualization support in Oracle Solaris Cluster

Agenda

- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Identifying virtualization support in Oracle Solaris Cluster

Clustering

- *Clustering* is a term that describes a group of two or more separate physical or virtual machines operating as a harmonious unit.
- Cluster characteristics:
  - Non-shared copies of the operating system
  - Dedicated hardware interconnect
  - Multiported storage
  - Cluster software *framework*
  - High availability (HA) and scalability
  - Support for a variety of cluster-unaware and cluster-aware applications

Clusters generally have the following characteristics:

- Separate server nodes, each booting from its own non-shared copy of the OS
- Dedicated hardware interconnect, providing a private transport only between the nodes of the same cluster
- Multiported storage, providing paths from at least two nodes in the cluster to each physical storage device that stores data for the applications running in the cluster
- Cluster software framework, providing cluster-specific knowledge to the nodes in the cluster about the health of the hardware and the health and state of their peer nodes
- General goal of providing a platform for HA and scalability for the applications running in the cluster
- Support for a variety of cluster-unaware applications and cluster-aware applications

High-Availability (HA) Platforms

- HA standards
- How clusters provide HA
- HA benefits of planned and unplanned outages
- Why fault tolerant servers are not an alternative to HA clusters

- High availability is the process of reducing any kind of down time to a minimum.
- HA standards:
  - HA ensures up to 99.999% up time for the application or about 5 minutes of down time per year.
  - A clean server reboot would exceed this.
- Fault-tolerant servers are not an alternative to HA clusters.

HA can be defined as a minimization of down time rather than the complete elimination of down time. Most true standards of HA cannot be achieved in a stand-alone server environment.

HA standards are usually phrased with wording such as “provides 5 nines availability.” This ensures up to 99.999% up time for the application or about 5 minutes of down time per year. One clean server reboot often already exceeds that amount of down time.

Many vendors provide servers that are marketed as fault tolerant. These servers are designed to be able to tolerate any single hardware failure, for example, memory failure, central processing unit (CPU) failure, and so on, without any down time.

There is a common misconception that fault-tolerant servers are an alternative to HA clusters, or that a fault-tolerant server supersedes HA in some way. In fact, although fault-tolerant servers can hide any hardware failure, they are not designed to provide especially fast recovery in the case of a software failure, such as an Oracle Solaris OS kernel panic or an application failure. Recovery in these circumstances on a single fault-tolerant server might still require a full OS reboot which, as previously stated, might already exceed the maximum down time permitted by the HA standards to which you aspire.

How Clusters Provide HA

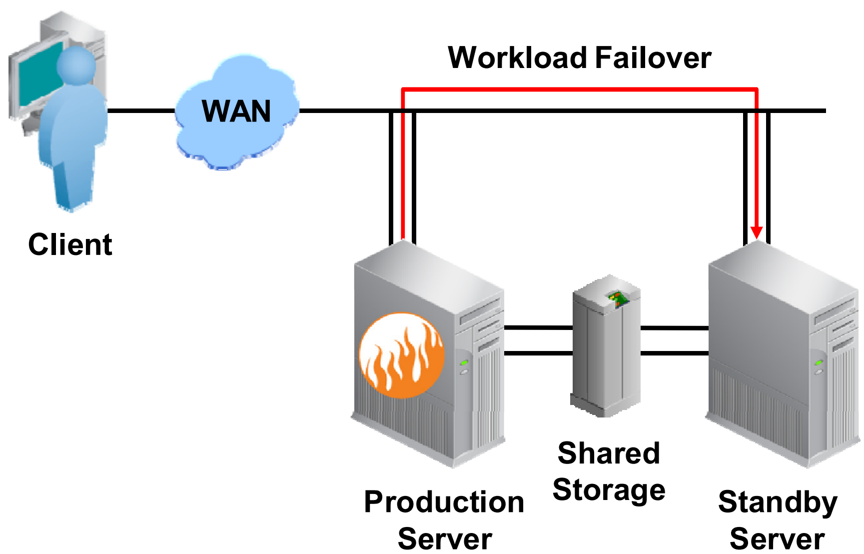
Inter-node failover:

- Application services and data are recovered automatically when there is any hardware or software failure.
- Application recovery is done without human intervention and is faster than a server reboot.

Clusters provide an environment where, in the case of any single hardware or software failure in the cluster, application services and data are recovered automatically (without human intervention) and quickly (faster than a server reboot). The existence of the redundant servers in the cluster and redundant server-storage paths makes this possible.

How Clusters Provide HA

Inter-node failover:



HA Benefits for Unplanned and Planned Outages

- Cluster can automatically relocate applications within the cluster in the case of failures.
- Reasons for down time:

Planned Down Time	Unplanned Down Time
Software upgrades	Environmental disasters
Hardware upgrades	Hardware failure
Testing	Application failure
Repairs	User error
Planned reboots	

The HA benefit that cluster environments provide involves not only hardware and software failures, but also planned outages. Although a cluster can automatically relocate applications within the cluster in the case of failures, it can also manually relocate services for planned outages. As such, normal reboots for hardware maintenance in the cluster affects only the up time of the applications for as much time as it takes to manually relocate the applications to different servers in the cluster.

Platforms for Scalable Applications

- Clusters enhance performance by running multiple instances of applications on multiple nodes of a cluster.
- HA and scalability are not mutually exclusive.

Clusters also provide an integrated hardware and software environment for scalable applications. Scalability is defined as the ability to increase application performance by supporting multiple instances of applications on different nodes in the cluster. These instances are generally accessing the same data as each other.

Clusters generally do not require a choice between availability and performance. HA is generally built into scalable applications as well as non-scalable ones. In scalable applications, you might not need to relocate failed applications because other instances are already running on other nodes. You might still need to perform recovery on behalf of failed instances.

Agenda

- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Virtualization support in Oracle Solaris Cluster

Oracle Solaris Cluster: Features

- Global device implementation
- Cluster file system implementation
- Cluster framework services implemented in the kernel
- Support for a wide variety of off-the-shelf applications in failover mode
- Support for several off-the-shelf applications in scalable mode
  - Client still sees these as single-server with a single IP.
  - Load balancing is implemented in the cluster itself.
- Tight integration with Oracle Solaris 11 zones

The Oracle Solaris Cluster hardware and software environment is the latest-generation clustering product. The following are the features of the Oracle Solaris Cluster product:

- **Global device implementation:** Although data storage must be physically connected on paths from at least two different nodes in the Oracle Solaris Cluster hardware and software environment, all the storage in the cluster is logically available from every node in the cluster by using standard device semantics. This provides the flexibility to run applications on nodes that use data that is not even physically connected to the nodes.
- **Global file system implementation:** The Oracle Solaris Cluster software framework provides a global file service independent of any particular application running in the cluster, so that the same files can be accessed on every node of the cluster, regardless of the storage topology.

Note: The global file system is also referred to as cluster file system.

- **Cluster framework services implemented in the kernel:** The Oracle Solaris Cluster software is tightly integrated with the Oracle Solaris OS kernels. Node monitoring capability, transport monitoring capability, and the global device and file system implementation are implemented in the kernel to provide higher reliability and performance.
- **Off-the-shelf application support:** The Oracle Solaris Cluster product includes data service agents for a large variety of cluster-unaware applications. These are tested programs and fault monitors that make applications run properly in the cluster environment.
- **Support for some off-the-shelf applications as scalable applications with built-in load balancing (global interfaces):** The scalable application feature provides a single Internet Protocol (IP) address and load-balancing service for some applications, such as Apache Web Server and Java System Web Server. Clients outside the cluster see the multiple node instances of the service as a single service with a single IP address.
- **Tight integration with Solaris 11 zones:** The Oracle Solaris Cluster framework is aware of Solaris 11 zones and can manage failover and scalable applications running in non-global Solaris Containers.

Agenda

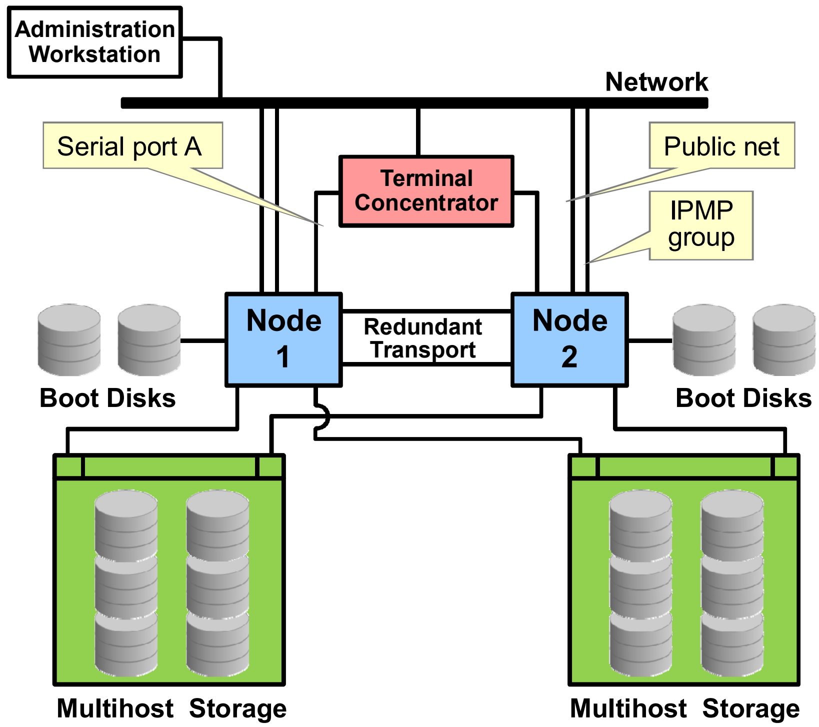
- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Identifying virtualization support in Oracle Solaris Cluster

Identifying Oracle Solaris Cluster Hardware Environment

Hardware components of a typical two-node cluster:

- Cluster nodes that are running Oracle Solaris OS
- Private cluster interconnect
- Public network interfaces
- Cluster disk storage
  - Multihost disks
  - Local disks
- Console access devices
- Administrative workstation

Oracle Solaris Cluster Hardware Environment



The Oracle Solaris Cluster hardware environment supports a maximum of 16 nodes. The hardware components of a typical two-node cluster comprise:

- Cluster nodes that are running Solaris 11 OS. Each node must run the same revision and same update of the OS.
- Separate boot disks on each node (with a preference for mirrored boot disks)
- One or more public network interfaces per system per subnet (with a preferred minimum of at least two)
- A redundant private cluster transport interface
- Dual-hosted, mirrored disk storage
- One terminal concentrator (or any other console access method)
- Administrative workstation

Cluster Nodes

- Oracle Solaris host systems
  - Rack-mounted servers: SPARC and x86 platforms
  - Desk-side and legacy servers
  - Enterprise-level (high-end) servers
- Oracle VM Server for SPARC
  - Fully supported with Oracle Solaris Cluster
  - Input/output (I/O) domains and guest domains
- Oracle VM VirtualBox
  - Virtualization software used for demonstration purpose rather than as a production platform

A wide range of server platforms are supported for use in the clustered environment. These range from small rack-mounted servers up to enterprise-level servers.

Different models of a server architecture are supported as nodes in the same cluster, based on the network and storage host adapters used. However, you cannot mix SPARC and x86 servers in the same cluster.

Oracle VM Server for SPARC

Oracle VM Server for SPARC is fully supported as cluster nodes. Both I/O domains and guest domains are supported.

Note: The term Oracle VM Server for SPARC, or Logical Domains, is the new name for LDoms (Sun Logical Domains). Throughout this course, the term *Logical Domain* is used as a short name to refer to Oracle VM Server for SPARC.

You can use one or more Logical Domains and one or more physical nodes that do not use Logical Domains in the same cluster.

Private Cluster Interconnect

All nodes in a cluster are linked by a private cluster transport interface:

- Cluster transport is used for the following purposes:
  - Cluster-wide monitoring and recovery
  - Global data access
  - A data path for cluster-aware applications
- Using two private networks is highly recommended.
  - You can add more for performance.
  - Configurations with only one are now allowed.
- The following hardware is supported:
  - Ethernet
  - `vnet` (in Logical Domains)
  - InfiniBand

All nodes in a cluster are linked by a private cluster transport. The transport can be used for the following purposes:

- Cluster-wide monitoring and recovery
- Global data access (transparent to applications)
- Application-specific transport for cluster-aware applications

It is highly recommended to use two separate private networks that form the cluster transport. You can have more than two private networks (and you can add more later). More private networks can provide a performance benefit in certain circumstances, because global data access traffic is striped across all the transports.

Oracle Solaris Cluster enables you to build configurations with a single private network forming the cluster transport. This would be recommended in production only if the single private network is already redundant (using a lower-level device aggregation).

Crossover cables are often used in a two-node cluster. Switches are optional when you have two nodes, and they are required for more than two nodes.

The following types of cluster transport hardware are supported:

- Ethernet (100 MB, 1 GB, and 10 GB adapters)
- `vnet` virtual Ethernet adapters in Logical Domains (Oracle VM Server for SPARC)
- InfiniBand

This is a relatively new industry standard interconnect used outside of the Oracle Solaris Cluster environment for interconnecting a variety of hosts and storage devices. In the Oracle Solaris Cluster environment, it is supported only as an interconnect between hosts, and not between hosts and storage devices.

Public Network Interfaces

- Clients connect to the cluster through the public network interfaces.
- Interfaces are controlled by IPMP.
- Using a minimum of two interfaces per IPMP group is recommended.
  - If one of the adapters fails, IPMP is called to fail over the defective interface to another adapter in the group.
- Configure as many IPMP groups as required.

Clients connect to the cluster through the public network interfaces. Each network adapter card can connect to one or more public networks, depending on whether the card has multiple hardware interfaces.

You can set up Oracle Solaris hosts in the cluster to include multiple public network interface cards that:

- Are configured so that multiple cards are active
- Serve as failover backups for one another

Each node must have public network interfaces that are under the control of the Oracle Solaris OS IP Multipathing (IPMP) software. It is recommended to have at least two interfaces in each IPMP group.

If one of the adapters fails, IP network multipathing software is called to fail over the defective interface to another adapter in the group. An Oracle Solaris Cluster server is not allowed to act as a router.

Cluster Disk Storage

- Multihost disks
  - Multihost disks are connected and shared with more than one Oracle Solaris host.
  - Multihost storage makes disks highly available.
- Multihost disks have the following characteristics:
  - Tolerance of single-host failures
  - Ability to store application data and configuration files
  - Protection against host failures
  - Global access through a primary host that “masters” the disks

ORACLE

Disks that can be connected to more than one Oracle Solaris host at a time are multihost devices. In the Oracle Solaris Cluster environment, multihost storage makes disks highly available.

Multihost devices have the following characteristics:

- Tolerance of single-host failures
- Ability to store application data, application binaries, and configuration files
- Protection against host failures. If clients request the data through one host and the host fails, the requests are switched over to use another host with a direct connection to the same disks.
- Global access through a primary host that “masters” the disks, or direct concurrent access through local paths

The Oracle Solaris Cluster hardware environment can use several storage models. They must all accept multihost connections. The StorEdge 6120 array has a single connection and must be used with a hub or a switch.

Some data storage arrays support only two physically connected nodes. Many other storage configurations support more than two nodes connected to the storage.

You can use ZFS and Solaris Volume Manager software to mirror the storage across controllers. You can choose not to use any volume manager if each node has multipathed access to HA hardware redundant array of independent disks (RAID) storage.

Cluster Disk Storage

Local disks:

- Local disks, also called *boot disks*, are the disks that are connected to only a single Oracle Solaris host.
- Boot disks must not be connected to multiple nodes.
- Boot disks are not visible to any other node.

Local disks are the disks that are connected to only a single Oracle Solaris host. The Oracle Solaris Cluster environment requires that boot disks for each node be local to the node. That is, the boot disks are not connected or not visible to any other node. For example, if the boot device was connected through a storage area network (SAN), it would still be supported if the LUN is not visible to any other nodes.

Note: Oracle Solaris Cluster software does not require that you mirror the ZFS root pool.

Console Access Devices

Terminal concentrator:

- A terminal concentrator (TC) is a typical way of accessing node consoles if you are using `ttya` console.
- A TC provides data translation from the network to serial port interfaces.
- There is a trade-off between security and convenience.

Servers supported in the Oracle Solaris Cluster environment have a variety of console access mechanisms.

If you are using a serial port console access mechanism (`ttya`), then you probably have a terminal concentrator in order to provide the convenience of remote access to your node consoles. A terminal concentrator (TC) is a device that provides data translation from the network to serial port interfaces. Each of the serial port outputs connects to a separate node in the cluster through serial port A.

There is always a trade-off between convenience and security. You might prefer to have only dumb-terminal console access to the cluster nodes, and keep these terminals behind locked doors requiring stringent security checks to open them. This is acceptable (although less convenient to administer) for Oracle Solaris Cluster hardware as well.

Administrative Workstation

To help you manage a cluster, administrative workstation software is available.

- It pops up windows and enables connections to nodes.
- It is just a convenience and is not required.

Included with the Oracle Solaris Cluster software is the administration console software, which can be installed on any SPARC or x86 Solaris OS workstation. The software can be a convenience in managing the multiple nodes of a cluster from a centralized location. It does not affect the cluster in any other way.

Quiz

The Oracle Solaris Cluster environment allows boot disks for a node to be shared with the other nodes in the cluster.

a. True
b. False

Answer: b

Oracle Solaris Cluster Hardware Redundancy: Features

- Redundant server nodes are required.
- Redundant transport is highly recommended.
- High-availability access to storage is required.
  - Multiple controllers for mirroring
  - Multipath access to hardware RAID
- Redundant public network interfaces per subnet are recommended.
- Redundant boot disks are recommended.

The following list summarizes the generally required and optional hardware redundancy features in the Oracle Solaris Cluster hardware environment:

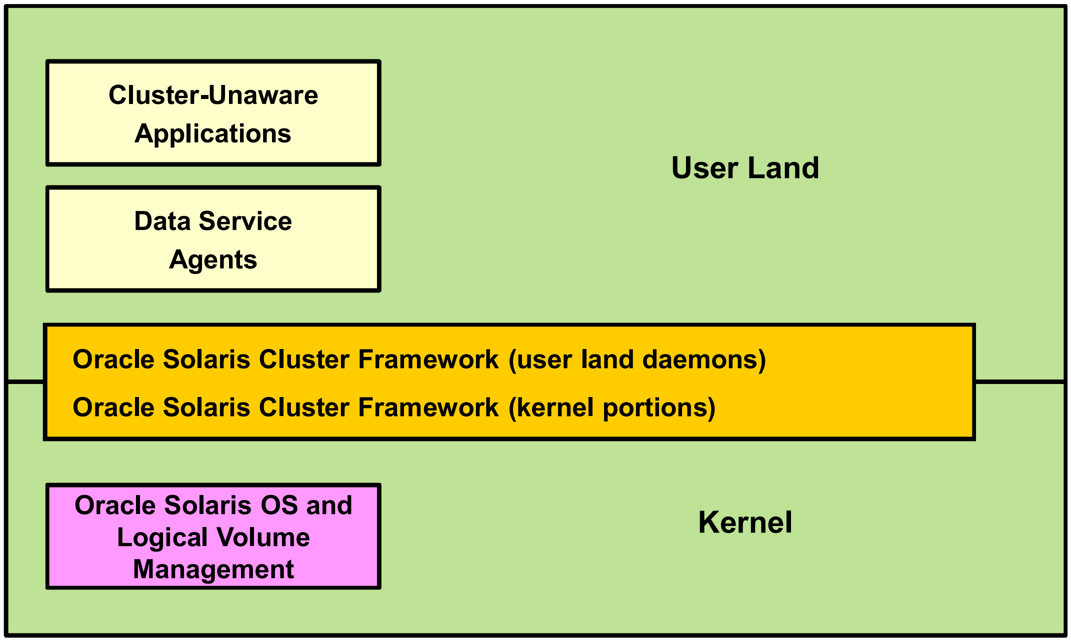
- Redundant server nodes are required.
- Redundant transport is highly recommended.
- HA access to data storage is required. That is, at least one of the following is required.
  - Mirroring across controllers for Just a Bunch of Disks (JBOD) or for hardware RAID devices without multipathing
  - Multipathing from each connected node to hardware RAID devices
- Redundant public network interfaces per subnet are recommended.

You should locate redundant components as far apart as possible. For example, on a system with multiple I/O boards, you should put the redundant transport interfaces, the redundant public nets, and the redundant storage array controllers on two different I/O boards.

Agenda

- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Identifying virtualization support in Oracle Solaris Cluster

Oracle Solaris Cluster Software Environment



The slide gives a graphical, high-level overview of the software components that work together to create the Oracle Solaris Cluster software environment.

To function as a cluster member, the following types of software must be installed on every Oracle Solaris Cluster node:

- Oracle Solaris OS software
- Oracle Solaris Cluster software
- Data service application software
- Logical volume management

An exception is a configuration that uses hardware RAID. This configuration might not require a software volume manager.

Agenda

- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Identifying virtualization support in Oracle Solaris Cluster

Identifying Applications That Are Supported by Oracle Solaris Cluster

- Cluster-unaware (off-the-shelf) applications
  - Failover applications
  - Scalable applications
- Cluster-aware applications:
  - Parallel database applications
  - Remote shared memory (RSM) applications

The Oracle Cluster software environment supports both cluster-unaware and cluster-aware applications.

Cluster-Unaware Applications

- Cluster-unaware applications run on a single server.
- The following are the common elements of these applications:
  - **Cluster's Resource Group Manager (RGM):** Coordinates all stopping and starting of the applications, which are never done by Solaris OS boot methods
  - **Data Service Agent:** Acts as "glue" to make applications work properly

A cluster-unaware application is an application that typically runs on a single server and is also called Off-the-Shelf application.

The majority of applications that run in a cluster are in the cluster-unaware category and are part of the main focus of this course.

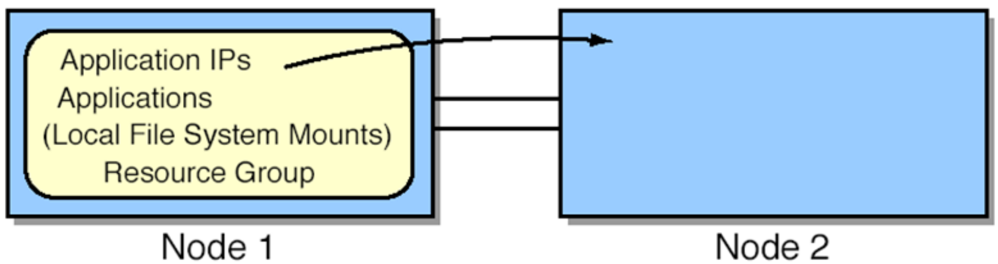
Two main categories of cluster-unaware applications that can run in the Oracle Solaris Cluster environment are failover applications and scalable applications.

The following are the common elements of all these applications:

- The cluster's Resource Group Manager (RGM) coordinates all stopping and starting of the applications, which are never started and stopped by traditional Solaris OS boot methods.
- A data service agent for the application provides the "glue" to make it work properly in the Sun Cluster software environment. This includes methods to start and stop the application appropriately in the cluster, as well as fault monitors specific to that application.

Failover Applications

- Failover applications run on only one node of the cluster at a time.
- They are usually paired with an "application IP."
  - IP fails over from node to node along with the application.
- Clients see a logical host name with no knowledge of which node a service is running on.
- Everything in the group fails over together.



The failover model is the easiest to support in the cluster. Failover applications run on only one node of the cluster at a time. The cluster provides HA by providing automatic restart on the same node or on a different node of the cluster.

Failover services are usually paired with an application IP address. This is an IP address that always fails over from node to node along with the application. In this way, clients outside the cluster see a logical host name with no knowledge of which node a service is running on. The client should not even be able to tell that the service is running in a cluster.

Note: Both IPV4 addresses and IPV6 addresses are supported.

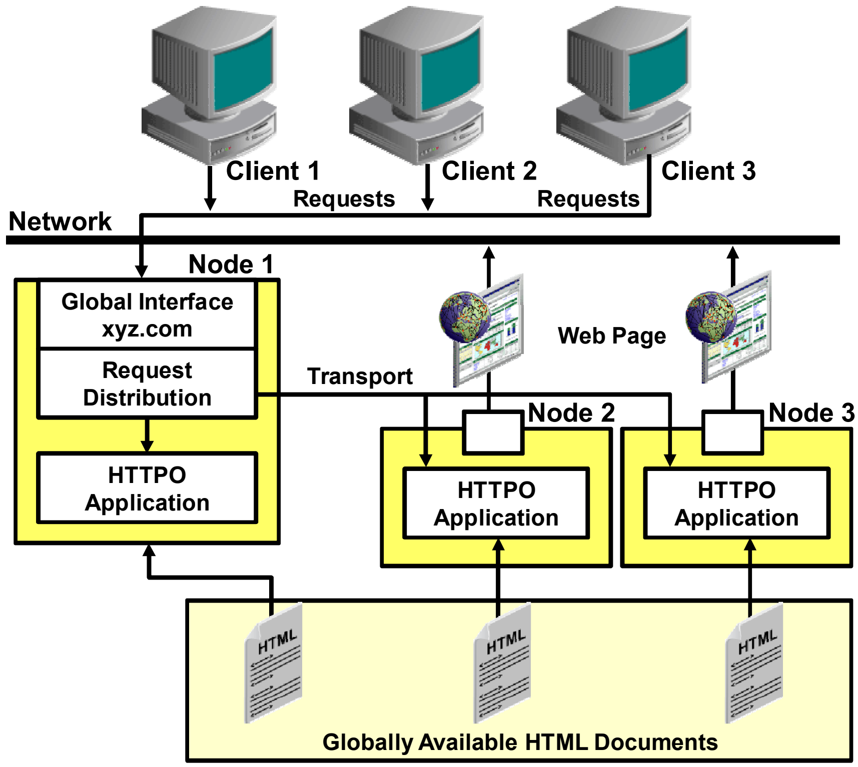
Multiple failover applications in the same resource group can share an IP address, with the restriction that they must all fail over to the same node together.

Scalable Applications

- Same application (same data) running on multiple nodes simultaneously
- Single IP and load balancing
- Off-the-shelf software
- Not all applications can do multiple access to data correctly.

Scalable applications involve running multiple instances of an application in the same cluster and making it look like a single service by means of a global interface that provides a single IP address and load balancing.

Scalable Applications



Although scalable applications are still off-the-shelf, not every application can be made to run as a scalable application in the Oracle Solaris Cluster software environment. Applications that write data without any type of locking mechanism might work as failover applications but do not work as scalable applications.

Cluster-Aware Applications

- Instances are aware of each other and communicate with each other across the transport.
- Cluster-aware applications are not necessarily managed by RGM.
- Applications are not necessarily logically grouped with external application IP.

ORACLE

Cluster-aware applications are applications in which knowledge of the cluster is built into the software. They differ from cluster-unaware applications in the following ways:

- Multiple instances of the application running on different nodes are aware of each other and communicate across the private transport.
- It is not required that the Solaris Cluster software framework RGM start and stop these applications. Because these applications are cluster-aware, they can be started in their own independent scripts or by hand.
- Applications are not necessarily logically grouped with external application IP addresses. If they are, the network connections can be monitored by cluster commands. It is also possible to monitor these cluster-aware applications with Solaris Cluster software framework resource types.

Cluster-Aware Applications

- Parallel databases application:
  - Capable of handling different queries on the same database
  - Supported parallel database application:
    - Oracle 11*g* RAC
- Remote Shared Memory (RSM) applications:
  - Highly efficient way for cluster-aware applications to share large amounts of data across the transport

Parallel database applications: Parallel database applications are a special type of cluster applications. Multiple instances of the database server cooperate in the cluster, handling different queries on the same database and even providing parallel query capability on large queries. A supported application is listed in the slide.

RSM applications: Applications that run on Oracle Solaris Cluster hardware can make use of an application programming interface (API) called Remote Shared Memory (RSM). This API maps data from an application instance running on one node to the address space of an instance running on another node. This can be a highly efficient way for cluster-aware applications to share large amounts of data across the transport. This requires the SCI interconnect.

Oracle Solaris Cluster Data Services

- Failover data services (many)
- Scalable data services (fewer)

Data service agents make cluster-unaware applications highly available, in either failover or scalable configurations.

HA and Scalable Data Service Support

The Oracle Solaris Cluster software provides preconfigured components that support HA data services.

For the list of the most recent components available, visit the Supported Products page at http://docs.oracle.com/cd/E23623_01/html/E23438/relnotes-6-products.html.

Quiz

In a cluster-unaware application, which component does all of the following?

- Acts as a glue between the cluster and the application
- Includes methods to start and stop the application appropriately in the cluster
- Includes fault monitors specific to that application

a. Cluster interconnect
b. Data service agent
c. Proxy agent

Answer: b

Agenda

- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Identifying virtualization support in Oracle Solaris Cluster

Oracle Solaris Cluster Software HA Framework

- It is a software layer that provides generic cluster services to the nodes in the cluster.
- Cluster software framework is implemented as a series of daemons and kernel modules.
- Services provided by the software HA framework include:
  - Node fault monitoring and cluster membership
  - Network fault monitoring
  - Application traffic striping
  - Cluster configuration repository

The Oracle Solaris Cluster software HA framework is the software layer that provides generic cluster services to the nodes in the cluster, regardless of which applications are running in the cluster. The Oracle Solaris Cluster software framework is implemented as a series of daemons and kernel modules. One advantage of the Oracle Solaris Cluster software environment is that much of the framework resides in the kernel, where it is fast, reliable, and always memory-resident. Some of the services provided by the framework are listed in the slide.

Oracle Solaris Cluster Software HA Framework

- Node fault monitoring and cluster membership:
  - Cluster membership monitor (CMM) generates heartbeats to monitor the cluster.
  - Heartbeats are automatically maintained across transport.
  - Cluster reconfiguration is initiated on heartbeat timeout.
- Network fault monitoring
  - Public network management (IPMP)
  - Cluster transport monitoring

The cluster membership monitor (CMM) is kernel-resident on each node and detects major cluster status changes, such as loss of communication between one or more nodes. The CMM relies on the transport kernel module to generate heartbeats across the transport medium to other nodes in the cluster. If the heartbeat from any node is not detected within a defined timeout period, it is considered as having failed, and a cluster reconfiguration is initiated to renegotiate cluster membership.

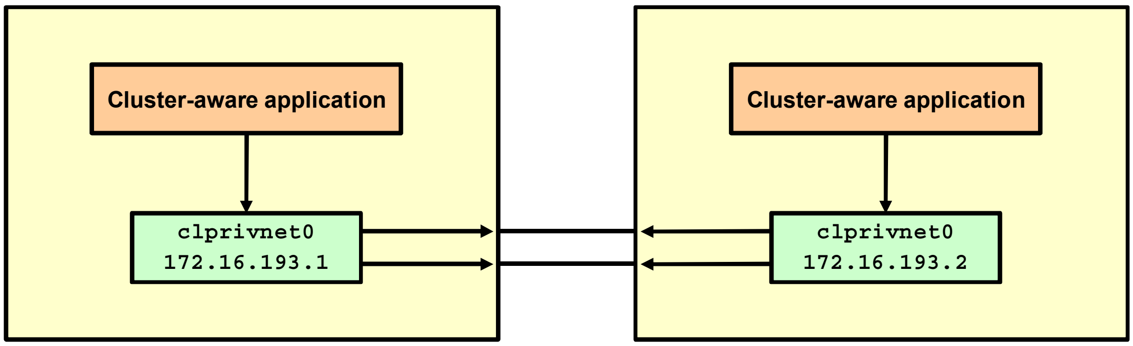
Network Fault Monitoring

Both the public network interfaces and the cluster transport interfaces are monitored for potential failures.

- **Public network management:** The Oracle Solaris Cluster software environment requires the use of IPMP, a standard Solaris OS feature, to control interface failures on a node. The Solaris Cluster software adds a layer of monitoring to detect total network failure on one node and to drive the possible failover of applications to another node.
- **Cluster transport monitoring:** The cluster transport interfaces are monitored on each node through heartbeats. If an active cluster transport interface on any node is determined to be inoperative, all nodes route interconnect traffic to functional transport interfaces. The failure is transparent to Oracle Solaris Cluster applications.

Oracle Solaris Cluster Software HA Framework

Application traffic striping



Applications written correctly can use the transport for data transfer. This feature stripes IP traffic sent to the per-node logical IP addresses across all private interconnects. Transmission Control Protocol (TCP) traffic is striped on a per connection granularity. User Datagram Protocol (UDP) traffic is striped on a per-packet basis. The cluster framework uses the `clprivnet0` virtual network device for these transactions. This network interface is visible with `ifconfig`. No manual configuration is required.

The application receives the benefit of striping across all the physical private interconnects, but needs to be aware of only a single IP address on each node configured on that node's `clprivnet0` adapter.

Cluster Configuration Repository

Repository for cluster configuration information:

- Cluster Configuration Repository (CCR) is kept consistent across nodes.
- CCR structures contain the following types of information:
  - Cluster, node, transport configuration
  - DID device configuration
  - Device group configuration (Solaris Volume Manager)
  - NAS information
  - Resource type, resource group, resource configuration

General cluster configuration information is stored in global configuration files collectively referred to as the cluster configuration repository (CCR). The CCR must be kept consistent between all nodes and is a critical element that enables each node to be aware of its potential role as a designated backup system.

Note: Never attempt to modify any of the CCR-related files. The files contain generation number information that is critical to the operation of the cluster software. The CCR is automatically modified as the result of administrative command execution.

The CCR structures contain the information listed in the slide.

Agenda

- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Identifying virtualization support in Oracle Solaris Cluster

Identifying the Global Storage Services

Global storage services:

- Is a flexible environment where logical access to storage is not dictated by physical connections
- Enables scalable services
- Provides more flexibility for failover services
- Has three different “layers”:
  - Global naming (DID devices)
  - Global devices
  - Cluster file system (global file system)

The Oracle Solaris Cluster software framework provides global storage services, a feature which greatly distinguishes the Oracle Solaris Cluster software product. Not only does this feature enable scalable applications to run in the cluster, but it also provides a much more flexible environment for failover services by freeing applications to run on nodes that are not physically connected to the data.

It is important to understand the differences and relationships between the services listed in the slide.

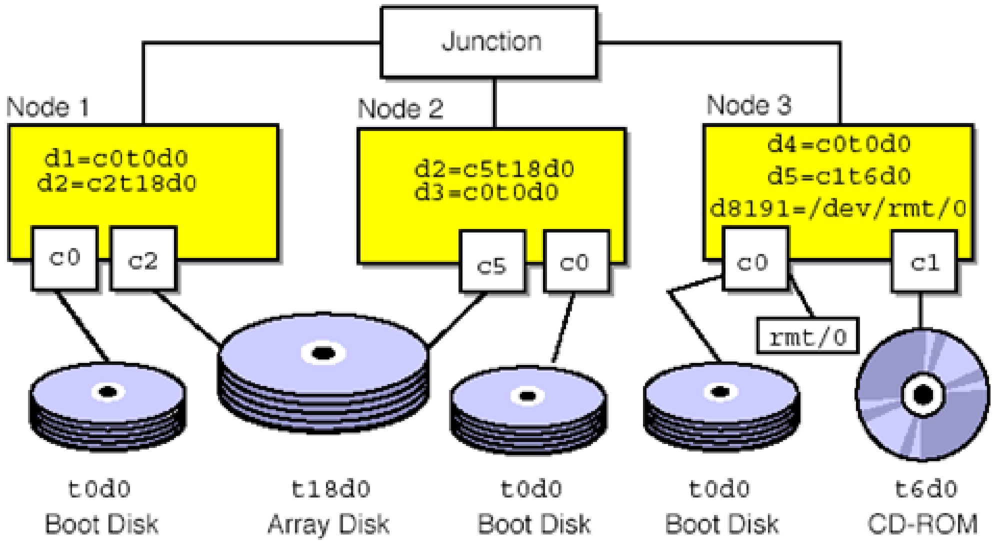
Note: The global file system is also referred to as *cluster file system*.

Global Naming

- Device ID (DID) provides unique naming scheme for devices:
  - Disk drive, CD-ROM drive, or tape drive in the cluster
- Shared storage devices have a unique name even if controller numbers are different on different nodes.
- Root storage devices have a unique name even if each node is using the same `c#t#d#`.
- Device files are created in both the `/dev/did/dsk` and `/dev/did/rdsk` directories.

The DID feature provides a unique device name for every disk drive, CD-ROM drive, or tape drive in the cluster. Shared disks that might have different logical names on different nodes (different controller numbers) are given a cluster-wide unique DID instance number. Different local disks that may use the same logical name (for example, `c0t0d0` for each node's root disk) are each given unique DID instance numbers.

Global Naming



The figure in the slide demonstrates the relationship between typical Oracle Solaris OS logical path names and DID instances.

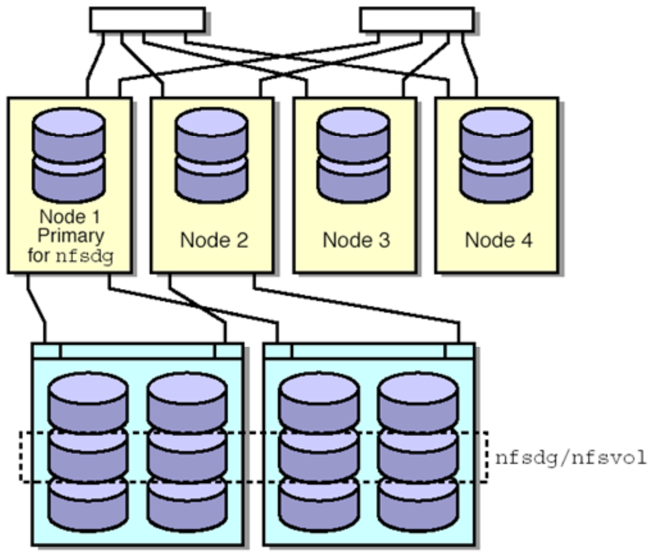
Device files are created for each of the standard eight Solaris OS disk partitions in both the `/dev/did/dsk` and `/dev/did/rdsk` directories (for example, `/dev/did/dsk/d2s3` and `/dev/did/rdsk/d2s3`).

DIDs themselves are just a global naming scheme and not a global access scheme.

DIDs are used as components of Solaris Volume Manager volumes and in choosing cluster quorum devices.

Global Devices

Provides cluster-wide, highly available access to any device in a cluster, from any node without regard to where the device is physically attached.



The global devices feature of Solaris Cluster software provides simultaneous access to the raw (character) device associated with storage devices from all nodes, regardless of where the storage is physically attached. This includes individual DID disk devices, CD-ROMs and tapes, as well as Solaris Volume Manager volumes.

The Solaris Cluster software framework manages automatic failover of the primary node for global device groups. All nodes use the same device path, but only the primary node for a particular device actually talks through the storage medium to the disk device. All other nodes access the device by communicating with the primary node through the cluster transport. All nodes have simultaneous access to the device `/dev/vx/rdsk/nfsdg/nfsvol`. Node 2 becomes the primary node if Node 1 fails.

In general, if a node fails while providing access to a global device, the Oracle Solaris Cluster software automatically discovers another path to the device. The Oracle Solaris Cluster software then redirects the access to that path. The local disks on each server are also not multiported, and thus are not highly available devices.

The cluster automatically assigns unique IDs to each disk, CD-ROM, and tape device in the cluster. This assignment enables consistent access to each device from any node in the cluster. The global device namespace is held in the `/dev/global` directory.

Device Files for Global Devices

- There is a `/global/.devices/node@nodeID` file system on each node.
  - *nodeID* is an integer representing a node in the cluster.
- Each file system is globally mounted (global file system).
- Solaris Volume Manager software directories are linked.

```
# df -h
/dev/lofi/127
94M 3.6M 81M 5% /global/.devices/node@1
/dev/lofi/126
94M 3.6M 81M 5% /global/.devices/node@2
proto192:/dev/vx# ls -l /dev/vx/rdsk/nfsdg
lrwxrwxrwx 1 root root 40 Nov 25 03:57
/dev/vx/rdsk/nfsdg ->/global/.devices/node@1/dev/vx/rdsk/nfsdg/
```

ORACLE

The Oracle Solaris Cluster software maintains a special file system on each node, completely dedicated to storing the device files for global devices. This file system has the mount point `/global/.devices/node@nodeID`, where *nodeID* is an integer representing a node in the cluster. The file system is stored on a dedicated partition on the boot disk.

All the `/global/.devices` file systems, one for each node are visible from each node. In other words, they are examples of global file systems.

The device names under the `/global/.devices/node@nodeID` arena can be used directly. However, because they are unwieldy, the Oracle Solaris Cluster environment provides symbolic links into this namespace.

For Solaris Volume Manager, the Oracle Solaris Cluster software links the standard device access directories into the global namespace.

```
proto192:/dev/vx# ls -l /dev/vx/rdsk/nfsdg
lrwxrwxrwx 1 root root 40 Nov 25 03:57
/dev/vx/rdsk/nfsdg ->/global/.devices/node@1/dev/vx/rdsk/nfsdg/
```

Device Files for Global Devices

- DID devices have `/dev/global` links that are global.
- `/dev/did/dsk`, `/dev/did/rdsk`, and `/dev/did/rmt` paths are not global.

```
proto192:/dev/md/nfsds# ls -l /dev/global
lrwxrwxrwx 1 root other 34 Nov 6 13:05
/dev/global -> /global/.devices/node@1/dev/global/

proto192:/dev/md/nfsds# ls -l /dev/global/rdsk/d3s0
lrwxrwxrwx 1 root root 39 Nov 4 17:43
/dev/global/rdsk/d3s0 -> ../../../devices/pseudo/did@0:3,3s0,raw

proto192:/dev/md/nfsds# ls -l /dev/global/rmt/1
lrwxrwxrwx 1 root root 39 Nov 4 17:43
/dev/global/rmt/1 -> ../../../devices/pseudo/did@8191,1,tp
```


For individual DID devices, the `/dev/did/dsk`, `/dev/did/rdsk`, and `/dev/did/rmt` standard directories are *not* global access paths. Instead, Oracle Solaris Cluster software creates alternate path names under the `/dev/global` directory that link to the global device space as shown in the slide.

Note: You do have raw (character) device access from one node, through the `/dev/global` device paths, to the boot disks of other nodes. In other words, though you cannot mount one node's root disk from another node, you can overwrite it with `newfs` or `dd`. It is not necessarily advisable to take advantage of this feature.

The cluster file system feature makes file systems simultaneously available on all nodes, regardless of their physical location.

The cluster file system capability is independent of the structure of the actual file system layout on disk. However, only certain file system types are supported by Oracle Solaris Cluster to be file systems underlying the global file system. One of these is:

- UNIX file system (UFS)

The Oracle Solaris Cluster software makes a file system global with a `global` mount option. This is typically in the `/etc/vfstab` file, but can be put on the command line of a standard `mount` command:

```
# mount -o global /dev/vx/dsk/nfs-dg/vol-01 /global/nfs
```

Highly Available Local File Systems

Highly Available Local File Systems (failover file system):

- Are an alternate way of managing a file system in shared storage
- Are mounted only on the node running the associated application
- Support UFS and ZFS
- Cannot be used for scalable services or for applications running on nodes not physically connected to storage

Oracle Solaris Cluster software also has support in the cluster for failover file system access. Highly Available Local File Systems, also known as failover file systems, are available only on one node at a time, on a node that is running a service and has a physical connection to the storage in question. Failover file systems are also called *non-global file systems*.

In Oracle Solaris Cluster, more file system types are supported as a failover file system than as underlying file systems for a global file system.

UFS and ZFS are the supported types of failover file systems.

Failover file system access is appropriate for failover services that run only on the nodes that are physically connected to storage devices. Failover file system access is not suitable for scalable services.

Failover file system access, when used appropriately, can have a performance benefit over global file system access. The global file system infrastructure has an overhead of maintaining replicated state information on multiple nodes simultaneously.

Quiz

What is the function of a DID in the cluster environment?

a. Provides simultaneous access to the raw (character) device associated with storage devices
b. Stores the device files for global devices
c. Provides a unique naming scheme for devices

Answer: c

Quiz

What is the function of global devices in the cluster environment?

a. Provide a unique naming scheme for devices

b. Provide simultaneous access to the raw (character) device associated with storage devices

c. Store the device files for global devices

Answer: b

Agenda

- Defining clustering
- Describing Oracle Solaris Cluster features
- Identifying Oracle Solaris Cluster hardware environment
- Identifying Oracle Solaris Cluster software environment
- Identifying Oracle Solaris Cluster–supported applications
- Identifying Oracle Solaris Cluster Software HA framework
- Identifying the global storage services
- Identifying virtualization support in Oracle Solaris Cluster

Virtualization Support in Oracle Solaris Cluster

- Oracle VM server for SPARC
- Oracle Solaris cones
- Zone clusters

Oracle VM Server for SPARC

- Fully supported with Oracle Solaris Cluster
- I/O domains and guest domains

Oracle VM Server for SPARC is fully supported as cluster nodes. Both I/O and guest domains are supported.

Note: The term Oracle VM Server for SPARC, or Logical Domains for short, is the new name for LDoms. Throughout this course, the term Logical Domain is used as a short name to refer to Oracle VM Server for SPARC.

You can use one or more Logical Domains and one or more physical nodes not using Logical Domains in the same cluster.

Oracle Solaris Zones

- Virtualized operating system services
- Isolated and secure environment for running applications
- Enable administrators to dedicate system resources to individual zones
- Reduce system administration complexity
- Reduce IT infrastructure costs

The Oracle Solaris Zones technology enables software partitioning of a Solaris 11 OS to support multiple independent operating systems with independent process space, allocated resources, and users. Zones are ideal for environments that consolidate a number of applications on a single server.

The cost and complexity of managing numerous machines make it advantageous to consolidate several applications on larger, more scalable servers.

Zone Clusters

- A zone cluster is a cluster of non-global Oracle Solaris zones.
- Zone cluster nodes are configured as non-global zones of the `solaris` brand that are set with the cluster attribute.
- No other brand type is permitted in a zone cluster.
- The isolation feature of Oracle Solaris Zones enables you to run supported services on the zone cluster in a similar way as running the services in a global cluster.

Summary

In this lesson, you should have learned how to:

- Define clustering
- Describe Oracle Solaris Cluster features
- Identify:
  - Oracle Solaris Cluster hardware environment
  - Oracle Solaris Cluster software environment
  - Oracle Solaris Cluster–supported applications
  - Oracle Solaris Cluster Software HA framework
  - Global storage services
  - Virtualization support in Oracle Solaris Cluster

Practice 2 Overview: Guided Tour of the Virtual Training Lab

This practice provides a guided tour of the virtual training lab.

While participating in the guided tour, you identify the Oracle Solaris Cluster hardware components, including the cluster nodes, terminal concentrator, and administrative workstation.

3

Establishing Cluster Node Console Connectivity

Objectives

After completing this lesson, you should be able to:

- Describe the different methods for accessing a console
- Give an overview of Oracle Solaris parallel console software
- Install the `pconsole` utility
- Use the `pconsole` utility

Agenda

- Describing the different methods for accessing a console
- Describing the Oracle Solaris Parallel Console Software
- Installing the `pconsole` utility
- Using the `pconsole` utility

Accessing the Cluster Node Consoles

- Required for emergency and information purposes
- Not necessarily required for most administrative tasks

This section describes different methods for achieving access to the Solaris Cluster node consoles. It is expected that a Solaris Cluster environment administrator:

- Does not require node console access for most operations described during the duration of the course. Most cluster operations require only that you be logged in on a cluster node as root or as a user with cluster authorizations in the Role-Based Access Control (RBAC) subsystem. It is acceptable to have direct `telnet`, `rlogin`, or `ssh` access to the node.
- Must have console node access for certain emergency and informational purposes. If a node is failing to boot, the cluster administrator will have to access the node console to figure out why. The cluster administrator might like to observe boot messages even in normal, functioning clusters.

Accessing Serial Port Consoles on Traditional Nodes

- Oracle Solaris Cluster nodes use the serial port `ttyA` as a console.
- With a graphics device, you are still supposed to redirect the console to `ttyA`.
- Connect to a serial console however you want.
- Beware of spurious `BREAK` signals.
- You can disable normal `BREAK` signals by:
  - Physical keyswitch
  - Logical keyswitch
  - Uncomment the following line:
    - `KEYBOARD_ABORT=alternate` in the file `/etc/default/kbd`

Traditional Oracle Solaris Cluster nodes usually use serial port `ttyA` as the console. Even if you have a graphics monitor and system keyboard, you are supposed to redirect console access to the serial port or an emulation thereof.

The rule for console connectivity is simple. You can connect to the node `ttyA` interfaces any way you prefer, if whatever device you have connected directly to the interfaces does not spuriously issue `BREAK` signals on the serial line. `BREAK` signals on the serial port bring a cluster node to the OK prompt, killing all cluster operations on that node.

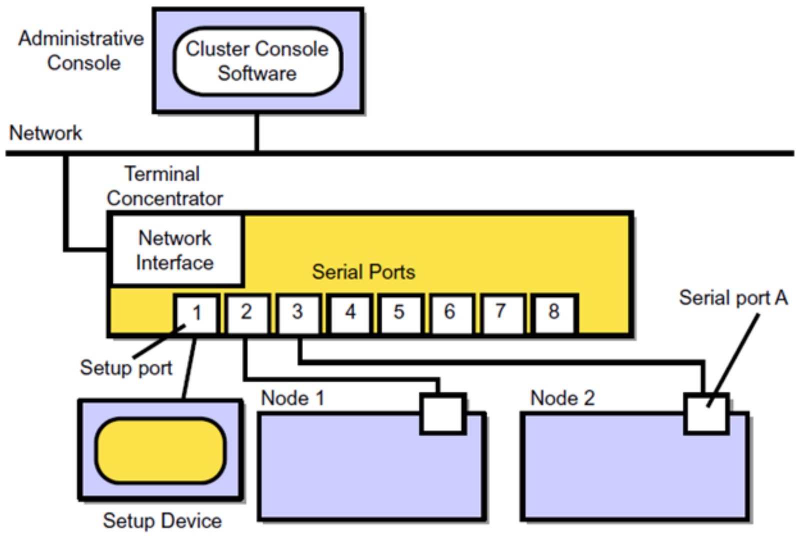
You can disable node recognition of a `BREAK` signal by a hardware keyswitch position (on some nodes), a software keyswitch position (on midrange and high-end servers), or a file setting (on all nodes). For those servers with a hardware keyswitch, turn the key to the third position to power the server on and disable the `BREAK` signal.

For those servers with a software keyswitch, issue the `setkeyswitch` command with the secure option to power the server on and disable the `BREAK` signal.

For all servers, while running Solaris OS, uncomment the line `KEYBOARD_ABORT=alternate` in `/etc/default/kbd` to disable receipt of the normal `BREAK` signal through the serial port. This setting takes effect on boot, or by running the `kbd -i` command as root. The Alternate Break signal is defined by the particular serial port driver that you happen to have on your system. You can use the `prtconf` command to figure out the name of your serial port driver, and then use `man serial-driver` to figure out the sequence. For example, for the `zs` driver, the sequence is carriage return, tilde (`~`), and Control + B: CR ~ CTRL + B. When the Alternate Break sequence is in effect, only serial console devices are affected.

Accessing Serial Port Node Consoles by Using a Terminal Concentrator

- Provides console-level access to each node from a remote workstation anywhere on the network
- The TC translates traffic through to regular serial traffic



One of the popular ways of accessing traditional node consoles is through a terminal concentrator (TC), a device which listens for connections on the network and passes through traffic (un-encapsulating and re-encapsulating all the TCP/IP headers) to the various serial ports.

A TC is also known as a Network Terminal Server (NTS). The figure in the slide shows a terminal concentrator network and serial port interfaces. The node public network interfaces are not shown. Although you can attach the TC to the public net, most security-conscious administrators would attach it to a private management network.

Most TCs enable you to administer TCP pass-through ports on the TC. When you connect with `telnet` to the TC's IP address and pass through port, the TC transfers traffic directly to the appropriate serial port (perhaps with an additional password challenge).

You can choose any type of TC as long as it does not issue BREAK signals on the serial ports when it is powered on, powered off, or reset, or at any other time that might be considered spurious. If your TC cannot meet that requirement, you can still disable recognition of the BREAK signal or enable an alternate abort signal for your node. Some terminal concentrators support Secure Shell. This might influence your choice, if you are concerned about passing TC traffic in the clear on the network.

Alternatives to a Terminal Concentrator for Nodes with a Serial Port Console

- Dumb terminals for each node
- Workstation with two serial ports as a "`tip launchpad`"

Some possible alternatives to the TC include the following:

- Use dumb terminals for each node console. If these are in a secure physical environment, this is certainly the most secure, but least convenient, method.
- Use a workstation that has two serial ports as a tip launchpad, especially for a cluster with only two nodes.

You can attach a workstation on the network exactly as you would place a TC, and attach its serial ports to the node consoles. You then add lines to the `/etc/remote` file of the Solaris OS workstation as follows:

```
node1:\:dv=/dev/term/a:br#9600:el=^C^S^Q^U^D:ie=%$:oe=^D:
node2:\:dv=/dev/term/b:br#9600:el=^C^S^Q^U^D:ie=%$:oe=^D:
```

This enables you to access node consoles by accessing the launchpad workstation and manually entering `tip node1` or `tip node2`. One advantage of using a Solaris OS workstation instead of a TC is that it is easier to tighten the security in the Solaris OS workstation. For example, you could easily disable `telnet` and `rlogin` access and require that administrators access the `tip` launchpad through Secure Shell.

Accessing the Node Console on Servers with Virtual Consoles

- Hardware domain-based systems (SC or SSP)
- V890 and so on (RSC as alternative to serial console)
- Most modern rackable systems (SC running ALOM)
- Oracle VM for SPARC: Virtual NTS through service domain

Many of the servers supported with Solaris Cluster have console access through a network connection to a virtual console device. These include:

- **Hardware domain-based systems:** The console access device is the system controller (SC) or system service processor (SSP).
- **Servers such as Sun Fire V890:** You can choose to have console access through the Remote System Control (RSC) device and software.
- **Modern rack-based servers:** The console access device is a small onboard system controller running Advanced Lights Out Management (ALOM).
- **Oracle VM for SPARC:** Console access to an Oracle VM for SPARC is through a network connection to the service domain, which provides the virtual console service for the Oracle VM for SPARC.

Agenda

- Describing the different methods for accessing a console
- Describing the Oracle Solaris Parallel Console software
- Installing the `pconsole` utility
- Using the `pconsole` utility

Oracle Solaris Parallel Console Software: Overview

The `pconsole` utility:

- Is part of the Oracle Solaris 11 `terminal/pconsole` package
- Provides convenient pop-up windows to nodes
- Opens a central (or master) console window that is used to send input to all nodes at one time
- Has a common window feature that replicates keystrokes for you across multiple windows

The `pconsole` utility is part of the Oracle Solaris 11 `terminal/pconsole` package.

The `pconsole` utility creates a host terminal window for each remote host that you specify on the command line. The utility also opens a central (or master) console window that you can use to send input to all nodes at one time. For additional information, see the `pconsole(1)` man page that is installed with the `terminal/pconsole` package.

Agenda

- Describing the different methods for accessing a console
- Describing the Oracle Solaris Parallel Console software
- Installing the `pconsole` utility
- Using the `pconsole` utility

Installing the `pconsole` Utility

To install the `pconsole` utility:

1. Become superuser on the administrative console.
2. Ensure that `solaris` and `ha-cluster` publishers are valid.

```
# pkg publisher
PUBLISHER                 TYPE     STATUS   URI
solaris                   origin   online   solaris-repository
ha-cluster                origin   online   solaris-repository
```

3. Install the `terminal/pconsole` package.

```
adminconsole# pkg install terminal/pconsole
```


Ensure that a supported version of the Oracle Solaris OS and any Oracle Solaris software updates are installed on the administrative console.

Notes:

- When you install the Oracle Solaris Cluster man page packages on the administrative console, you can view them from the administrative console before you install Oracle Solaris Cluster software on the cluster nodes or on a quorum server.
- Setting the publisher origin to the file repository URI: To enable client systems to get packages from your local file repository, you need to reset the origin for the `solaris` publisher. Execute the following command on each client:

  ```
  # pkg set-publisher -G '*' -g /net/host1/export/repoSolaris11/ solaris
  ```

  - `-G '*'`: Removes all existing origins for the `solaris` publisher
  - `-g`: Adds the URI of the newly created local repository as the new origin for the `solaris` publisher

Installing the `pconsole` Utility

4. (Optional) Install the Oracle Solaris Cluster man page packages.

```
adminconsole# pkg install man_pkgname
```

Package Name	Description
`ha-cluster/system/manual`	Oracle Solaris Cluster framework man pages
`ha-cluster/system/manual/data-services`	Oracle Solaris Cluster data service man pages
`ha-cluster/service/quorum-server/manual`	Oracle Solaris Cluster Quorum Server man pages

ORACLE

Installing the `pconsole` Utility

4. Set the directory paths on the administrative console
   – If you installed the `ha-cluster/system/manual/data-services` package, ensure that the `/opt/SUNWcluster/bin/` directory is in the PATH.
   – If you installed any other man page package, ensure that the `/usr/cluster/bin/` directory is in the PATH.
5. Start the `pconsole` utility.

```
adminconsole# pconsole host[:port] […] &
```

Quiz

Installing and using the `pconsole` software is the only way to access host systems participating in a cluster.

a. True

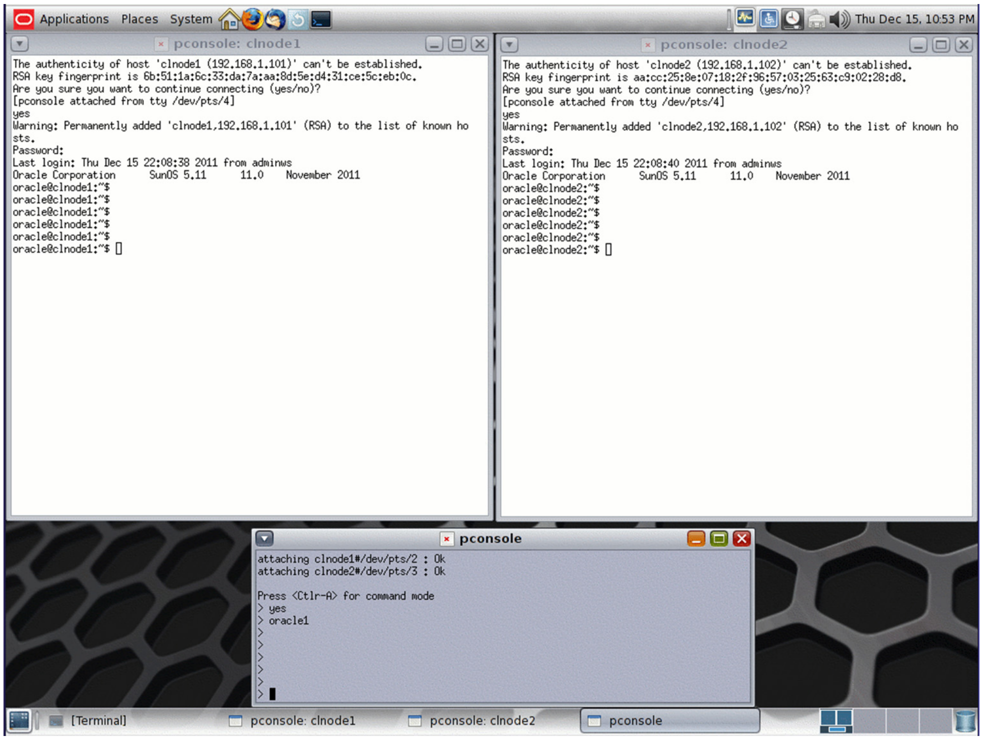
b. False

Answer: b

Agenda

- Describing the different methods for accessing a console
- Describing the Oracle Solaris Parallel Console software
- Installing the `pconsole` utility
- Using the `pconsole` utility

Parallel Console Tools: Look and Feel



All the tools have the same general look and feel. The tool automatically shows one new window for each node, and a small common keystroke window. You can type in each individual window as desired. Input that is directed to the common window is automatically replicated to all the other windows.

Summary

In this lesson, you should have learned how to:

- Describe the different methods for accessing a console
- Give an overview of Oracle Solaris parallel console software
- Install the `pconsole` utility
- Use the `pconsole` utility

Practice 3 Overview: Connecting to the Cluster Node Console

This practice covers the following topics:

- Task 1: Updating host name resolution
- Task 2: Installing the `pconsole` utility
- Task 3: Configuring the `pconsole` utility
- Task 4: Using the `pconsole` utility

4

Preparing for the Oracle Solaris Cluster Installation

Objectives

After completing this lesson, you should be able to:

- Prepare the Oracle Solaris Operating System (OS) environment
- Configure the Oracle Solaris Cluster storage connections
- Describe the quorum votes and quorum devices
- Describe persistent quorum reservations and cluster amnesia
- Describe data fencing
- Configure a cluster interconnect
- Identify public network adapters
- Configure shared physical adapters

Agenda

- Preparing the Oracle Solaris OS environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Preparing the Oracle Solaris OS Environment

- Selecting an Oracle Solaris installation method
- Oracle Solaris Operating System (OS) feature restrictions
- System disk partitions

Selecting an Oracle Solaris Installation Method

Install the Oracle Solaris software by using:

- Local DVD-ROM
- Automated Installer (AI)

You can install Oracle Solaris software from a local DVD-ROM or from a network installation server by using the Automated Installer (AI) installation method. In addition, Oracle Solaris Cluster software provides a custom method for installing both the Oracle Solaris OS and Oracle Solaris Cluster software by using the AI installation method. During AI installation of Oracle Solaris software, you choose to either install the OS with defaults accepted or run an interactive installation of the OS where you can customize the installation for components, such as the boot disk and the ZFS root pool. If you are installing several cluster nodes, consider a network installation.

Oracle Solaris OS Feature Restrictions

Consider the following points when you plan for the Oracle Solaris OS in an Oracle Solaris Cluster configuration:

- Oracle Solaris Zones
  - Install Oracle Solaris Cluster framework software only in the global zone.
- Loopback file system (LOFS)
  - Disable LOFS if NFS and `automountd` daemon are running.
- Power-saving shutdown
  - Not supported in Oracle Solaris Cluster configurations
- Network Auto-Magic (NWAM)
  - Disable NWAM before you configure or run your cluster.

Consider the following points when you plan to use the Oracle Solaris OS in an Oracle Solaris Cluster configuration.

- **Oracle Solaris Zones:** Install Oracle Solaris Cluster framework software only in the global zone.
- **Loopback file system (LOFS):** During cluster creation, LOFS capability is enabled by default. If the cluster meets both of the following conditions, you must disable LOFS to avoid switchover problems or other failures:
  - HA for NFS is configured on a highly available local file system.
  - The `automountd` daemon is running.

  If the cluster meets any one of these conditions, you can safely enable LOFS.

  If you require both LOFS and the `automountd` daemon to be enabled, exclude from the automounter map all files that are part of the highly available local file system that is exported by HA for NFS.

- **Power-saving shutdown:** Automatic power-saving shutdown is not supported in Oracle Solaris Cluster configurations and should not be enabled. See the `poweradm`(1M) man page for more information.
- **Network Auto-Magic (NWAM):** The Oracle Solaris Network Auto-Magic (NWAM) feature activates a single network interface and disables all others. Hence, NWAM cannot coexist with the Oracle Solaris Cluster software, and you must disable it before you configure or run your cluster.

Oracle Solaris OS Feature Restrictions

IP Filter

- Oracle Solaris Cluster relies on IP multipathing (IPMP) for public network monitoring.
- IP Filter configuration must follow the IPMP configuration guidelines.

- **IP Filter:** Oracle Solaris Cluster relies on IP network multipathing (IPMP) for public network monitoring. Any IP Filter configuration must be made in accordance with IPMP configuration guidelines and restrictions concerning IP Filter.

System Disk Partitions

- The space requirements for the `root` (`/`) file system are as follows:
  - Oracle Solaris Cluster software requires less than 40 MB of space in the root (`/`) file system.
  - Each Oracle Solaris Cluster data service might use between 1 MB and 5 MB.
  - Solaris Volume Manager software requires less than 5 MB
  - Log file requires up to 100 MB.
  - `lofi` device for the global-devices namespace requires up to 100 MB.
- `/var` file system requires up to 100 MB.
- `swap` space must be not less than 750 MB.
- Create 20 MB partition on slice 6 for volume manager use.

When you install the Oracle Solaris OS, ensure that you create the required Oracle Solaris Cluster partitions and that all partitions meet minimum space requirements.

- **Root (`/`):** The primary space requirements for the root (`/`) file system are as follows:
  - The Oracle Solaris Cluster software occupies less than 40 MB of space in the root (/) file system.
  - Each Oracle Solaris Cluster data service might use between 1 MB and 5 MB.
  - Solaris Volume Manager software requires less than 5 MB.
  - You need to set aside ample space for log files. Also, more messages might be logged on a clustered node than would be found on a typical stand-alone server. Therefore, allow at least 100 MB for log files.
  - The `lofi` device for the global-devices namespace requires 100 MB of free space. In Oracle Solaris Cluster 4.0, a dedicated partition is no longer used for the global-devices namespace.

- To configure ample additional space and inode capacity, add at least 100MB to the amount of space you would normally allocate for your root (`/`) file system. This space is used for the creation of both block special devices and character special devices used by the volume management software. You especially need to allocate this extra space if a large number of shared disks are in the cluster.

- **`/var` :** The Oracle Solaris Cluster software occupies a negligible amount of space in the `/var` file system at installation time. However, you need to set aside ample space for log files. Also, more messages might be logged on a clustered node than would be found on a typical stand-alone server. Therefore, allow at least 100 MB for the `/var` file system.
- **`swap`:** The combined amount of `swap` space that is allocated for Oracle Solaris and Oracle Solaris Cluster software must be no less than 750 MB. For best results, add at least 512 MB for Oracle Solaris Cluster software to the amount that is required by the Oracle Solaris OS. In addition, allocate any additional `swap` amount that is required by applications that are to run on the Oracle Solaris host.

  Note: If you create an additional `swap` file, do not create the `swap` file on a global device. Use only a local disk as a `swap` device for the host.
- **Volume manager:** Create a 20 MB partition on slice 6 for volume manager use.

Agenda

- Preparing the Oracle Solaris OS environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Oracle Solaris Cluster Storage Connections

- Oracle Solaris Cluster software supports up to 16 nodes.
- A shared storage device can connect to as many nodes as the storage device supports.

Previous versions of the Oracle Solaris Cluster software had strict rules regarding how many nodes were supported in various disk topologies. The only rules in the Oracle Solaris Cluster software regarding the data storage for the cluster are the following:

- Oracle Solaris Cluster software supports up to 16 nodes. Some storage configurations have restrictions on the total number of nodes supported.
- A shared storage device can connect to as many nodes as the storage device supports.
- Shared storage devices do not need to connect to all nodes of the cluster. However, these storage devices must connect to at least two nodes.

Quiz

Which of the following statements are true?

a. A shared storage device cannot be used as a boot device.
b. For the nodes participating in a cluster, you must use the same version of the Solaris OS, including the OS update.
c. In a cluster, you can connect the storage to as many nodes as the storage device supports.
d. Oracle Solaris Cluster can support more than 16 nodes in some topologies.

Answer: a, b, c

Identifying Cluster Storage Topologies

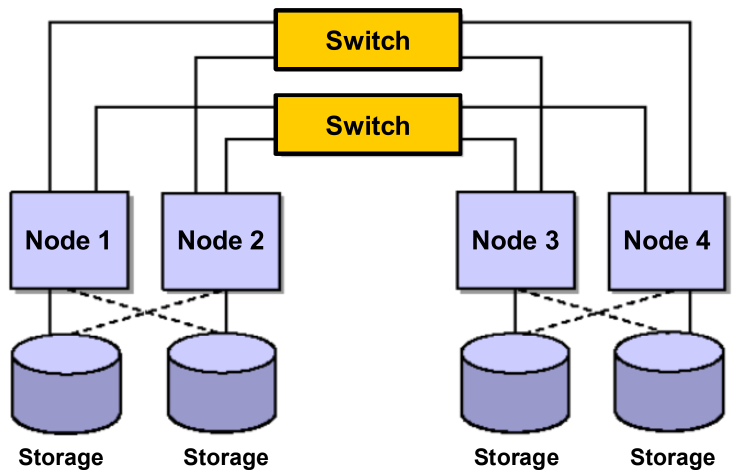
Oracle Solaris Cluster supports the following topologies:

- Clustered pairs
- Pair+N
- N+1
- N*N scalable
- NAS device-only
- Data replication
- Single-node
- Disaster recovery

Cluster topologies describe typical ways in which cluster nodes can be connected to data storage devices. Oracle Solaris Cluster does not require you to configure a cluster by using specific topologies.

Clustered Pairs Topology

- Nodes are configured in pairs and share storage
  - You can have any even number of nodes from 2 to 16
- Suitable for failover data services



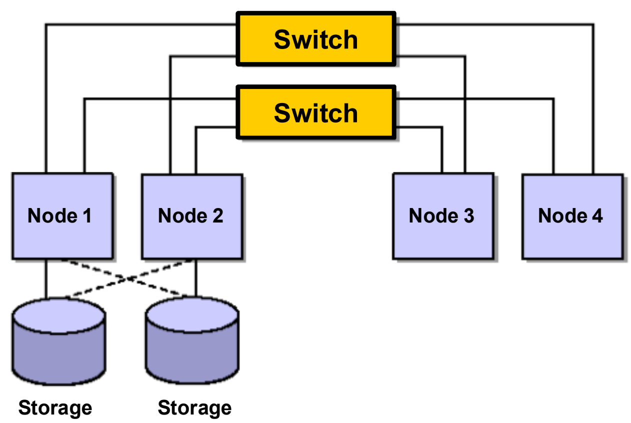
In a clustered pairs topology, two or more pairs of nodes are connected with each pair physically connected to some storage. Because of the global device and global file system infrastructure, this does not restrict where applications can fail over to and run. Still, it is likely that you will configure applications to fail over in pairs of nodes attached to the same storage.

Features of clustered pair configurations:

- Nodes are configured in pairs. You can have any even number of nodes from 2 to 16.
- Each pair of nodes shares storage. Storage is connected to both nodes in the pair.
- All nodes are part of the same cluster. You are likely to design applications that run on the pair of nodes physically connected to the data storage for that application, but you are not restricted to this design.
- Because each pair has its own storage, no one node must have a significantly higher storage capacity than the others.
- This configuration is well suited for failover data services.
- This configuration is well suited if you have a legacy SCSI-array or any disk array that can be attached to no more than two nodes.

Pair+N Topology

- All the shared storage is connected to a single pair of cluster nodes.
- Other cluster nodes must use the cluster interconnect to access shared storage.
- It is suitable for scalable data services.



The Pair+N topology includes a pair of nodes that are directly connected to the shared storage and nodes that must use the cluster interconnect to access shared storage because they have no direct connection themselves.

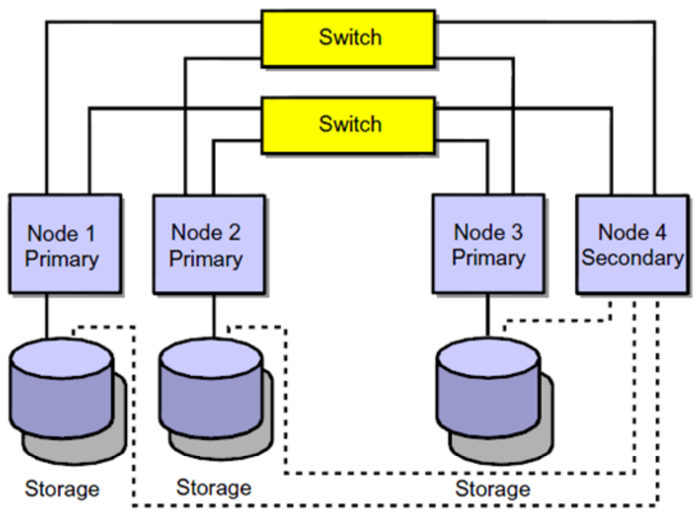
Features of Pair+N configurations:

- All shared storage is connected to a single pair.
- Additional cluster nodes support scalable data services or failover data services with the global device and file system infrastructure.
- A maximum of 16 nodes are supported.
- There are common redundant interconnects between all the nodes.
- The Pair+N configuration is well suited for scalable data services.
- This configuration is well suited if you have a legacy SCSI-array or any disk array that can be attached to no more than two nodes.

The limitations of a Pair+N configuration are that there can be heavy data traffic on the cluster interconnects. You can increase the bandwidth by adding more cluster transports.

N+1 Topology

- Using a single storage backup (secondary) in case of failure of any of the other nodes.
- Secondary node is physically connected to all the multihost storage.



The N+1 topology enables one system to act as the storage backup for every other system in the cluster. All of the secondary paths to the storage devices are connected to the redundant or secondary system, which can be running a normal workload of its own.

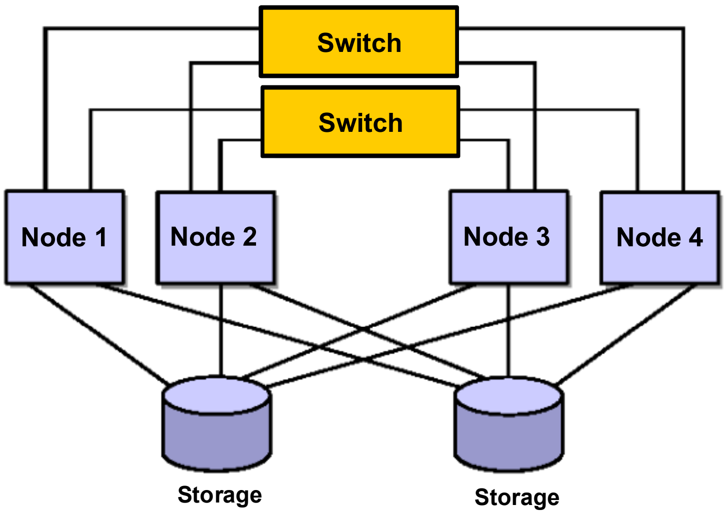
Features of N+1 configurations:

- The secondary node is the only node in the configuration that is physically connected to all the multihost storage.
- The backup node can take over without any performance degradation.
- The backup node is more cost effective because it does not require additional data storage.
- This configuration is best suited for failover data services.
- This configuration is well suited if you have legacy SCSI-array, or any disk array that can only be attached to two nodes.

A limitation of the N+1 configuration is that if there is more than one primary node failure, you can overload the secondary node.

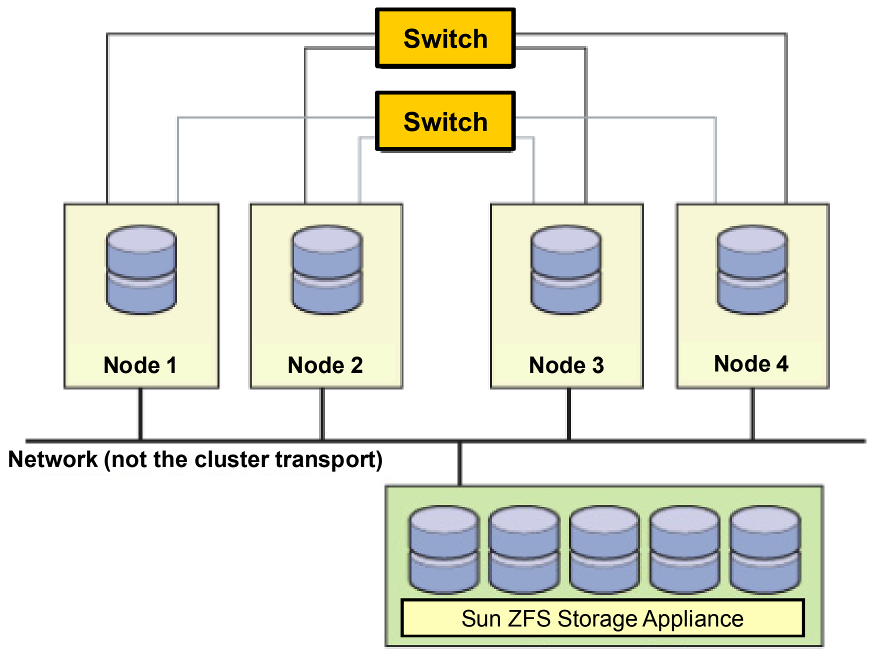
N*N Scalable Topology

- More than two nodes can be physically connected to the same storage.
- It is required for Oracle Real Application Clusters (Oracle RAC) on more than two nodes.



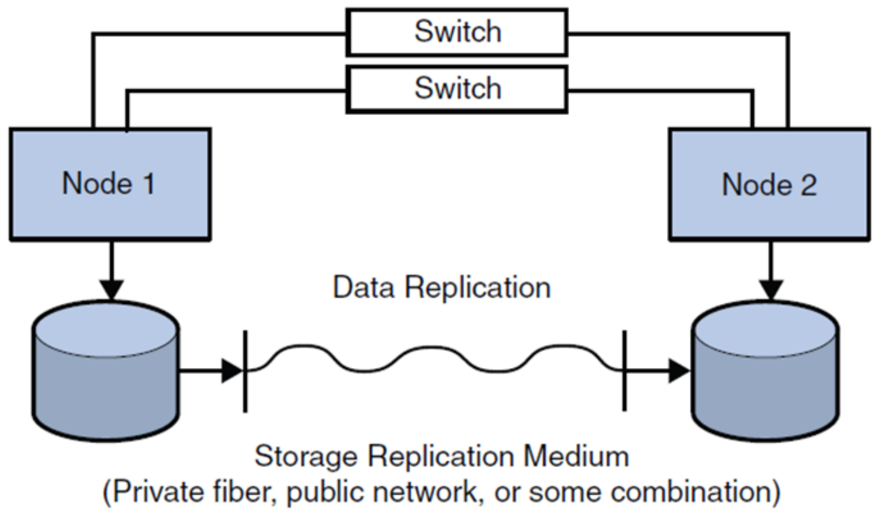
In a scalable, or N*N, topology, more than two nodes can be physically connected to the same storage. This configuration is required for runningOracle Real Application Clusters (Oracle RAC) across more than two nodes. For ordinary, cluster-unaware applications, each particular disk group or diskset in the shared storage still supports physical traffic from only one node at a time. However, having more than two nodes physically connected to the storage adds flexibility and reliability to the cluster.

NAS Device-Only Topology



Data Replication Topology

- Data storage is not physically multiported between nodes.
- Data is replicated between storage attached to the individual nodes by using controller-based replication.



Oracle Solaris Cluster supports a data replication topology. In this topology, data storage is not physically multiported between nodes but, rather, is replicated between storage attached to the individual nodes by using controller-based replication.

The data replication topology is ideal for wider-area clusters where the data replication solution is preferred to the extra connectivity that would be involved to actually connect the storage to nodes that are far apart. This topology would be ideal with the quorum server feature.

Single-Node Cluster Topology

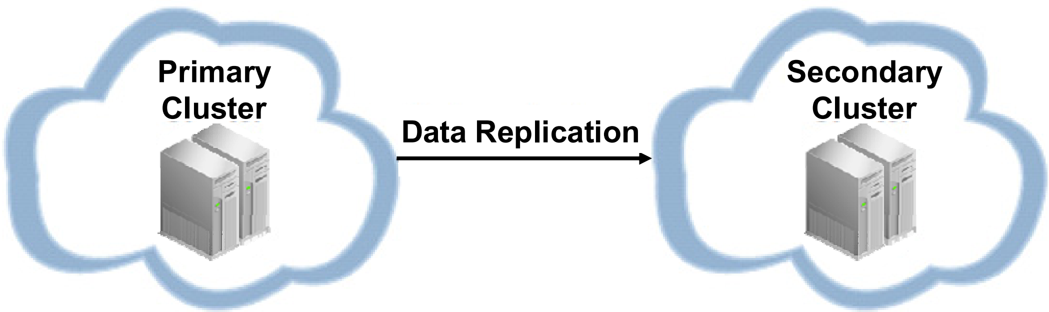
- A complete cluster framework on a single node
- Ideal for agent development and for training
- Could be used to watch applications fail over, even between non-global Oracle Solaris Zones
- Can be useful as a Oracle Solaris Cluster Geographic Edition partner member

In this configuration, one node or domain forms the entire cluster. This configuration allows for a single node to run as a functioning cluster. It offers users the benefits of having application management functionality and application restart functionality. The cluster starts and is fully functional with just one node.

Single-node clusters are ideal for users learning how to manage a cluster, to observe cluster behavior (possibly for agent development purposes), or to begin a cluster with the intention of adding nodes, as time goes on. Oracle Solaris Cluster provides the ability to experience application failovers, even on a single-node cluster. You could have an application that fails over between different nonglobal Oracle Solaris zones on the node.

Single-node clusters can also be useful in the Oracle Solaris Cluster Geographic Edition product, which manages a partnership of two clusters with data replication across a wide area. Each member of such a partnership must be a full Oracle Solaris Cluster installation, and a one-node cluster on either or both ends is acceptable.

Solaris Cluster Geographic Edition Software: A Cluster of Clusters



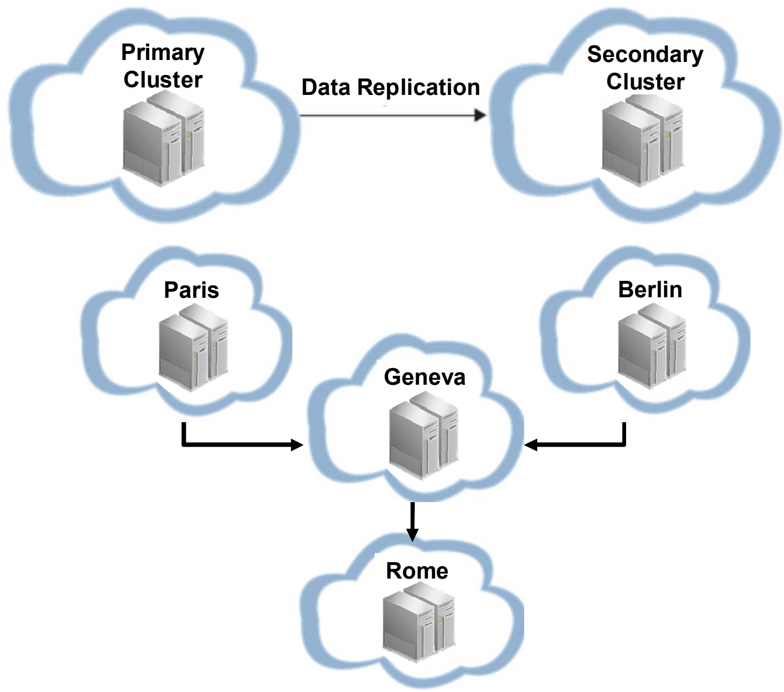
Oracle Solaris Cluster Geographic Edition enables you to implement a disaster recovery scenario by forming a conceptual "cluster of clusters" across a wide area. Application data is then replicated by using data replication.

The Oracle Solaris Cluster Geographic Edition software has the following properties:

- Solaris Cluster Geographic Edition software is configured on top of standard Solaris Cluster software on the participating clusters.
- Exactly two clusters are involved in the relationship shown in the diagram, and are said to form a partnership.
- There is no conceptual limit to the distance between the two clusters.

- Oracle Solaris Cluster Geographic Edition does not currently provide an automatic failover of an application across the two clusters. Instead it provides very simple commands to migrate an application (either nicely or forcefully) across a wide area, while simultaneously performing the correct operations on the data replication framework.
- Oracle Solaris Cluster 4.0 offers reliable protection from disaster for traditional or virtualized workloads on Oracle Solaris 11 through automated application failover and coordination with replication solutions such as StorageTek Availability Suite 4.0, Oracle Data Guard, and a script-based plug-in.

Solaris Cluster Geographic Edition Software: A Cluster of Clusters



It is possible to set up a Solaris Cluster Geographic Edition partnership as a more symmetrical entity. Though each partnership has only two clusters, it is possible for a single cluster to be a member of different partnerships.

The following points reinforce the main concepts of Oracle Solaris Cluster Geographic Edition by comparing its elements to a single-cluster configuration:

- Oracle Solaris Cluster resource groups control manual and automatic migration/failover of applications within a single cluster. Oracle Solaris Cluster Geographic Edition protection groups provide a framework for control of application migration and data replication between remote clusters, but the actual migration/takeover is manual (an easy three-word command).
- Single-cluster configurations do support data replication as an alternative to full storage multiporting between the nodes. This enables single clusters to run in a wider area (campus or metro clusters) without having to connect nodes to storage that is far away. Oracle Solaris Cluster Geographic Edition depends on data replication to provide a disaster-recovery scenario for data and applications that can be an arbitrarily wide distance apart.

Quiz

Match the following cluster topology with its benefits or features:

Pair+N topology

a. Suitable for an application that requires no data storage
b. Well suited for scalable data services
c. Ideal for wider-area clusters
d. Required for running Oracle RAC across more than two nodes
e. Best suited for failover data services

Answer: b

Quiz

Match the following cluster topology with its benefits or features:

N+1 topology

a. Suitable for an application that requires no data storage
b. Well suited for scalable data services
c. Ideal for wider-area clusters
d. Required for running Oracle RAC across more than two nodes
e. Best suited for failover data services

Answer: e

Quiz

Match the following cluster topology with its benefits or features:

Scalable storage (N*N) topology

a. Suitable for an application that requires no data storage
b. Well suited for scalable data services
c. Ideal for wider-area clusters
d. Required for running Oracle RAC across more than two nodes
e. Best suited for failover data services

Answer: d

Quiz

Match the following cluster topology with its benefits or features:

Data-replication topology

a. Suitable for an application that requires no data storage
b. Well suited for scalable data services
c. Ideal for wider-area clusters
d. Required for running Oracle RAC across more than two nodes
e. Best suited for failover data services

Answer: c

Quiz

Match the following cluster topology with its benefits or features:

Nonstorage topology

a. Suitable for an application that requires no data storage
b. Well suited for scalable data services
c. Ideal for wider-area clusters
d. Required for running Oracle RAC across more than two nodes
e. Best suited for failover data services

Answer: a

Agenda

- Preparing the Oracle Solaris OS Environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Need for Quorum Voting

- A quorum device is a shared storage device.
- A quorum server is shared by two or more nodes to contribute votes that are used to establish a quorum.
- Clusters operate only when a quorum of votes is available.

Types of Quorum Devices

Oracle Solaris Cluster software supports multiple kinds of quorum devices:

- Quorum disk
  - Nominated disk in the shared storage of your cluster
- Quorum server
  - Solaris server in the network
- Network attached storage device

Describing Quorum Votes and Quorum Devices

- Each node has exactly one vote.
- Votes are assigned to the following devices:
  - Directly attached shared disks (most usual type)
  - NAS quorum devices
  - Quorum servers
- A majority of votes (greater than one-half) is required to form a cluster or to remain in the cluster

The cluster membership subsystem of the Oracle Solaris Cluster software framework operates on a voting system as follows:

- Each node is assigned exactly one vote.
- Certain devices can be identified as quorum devices and are assigned votes. The following are types of quorum devices:
  - Directly attached multiported disks: Disks are the traditional type of quorum device and have been supported in all versions of Solaris Cluster.
  - NAS quorum devices
  - Quorum servers
- There must be a majority (more than 50 percent of all possible votes present) to form a cluster or remain in a cluster.

Benefits of Quorum Voting

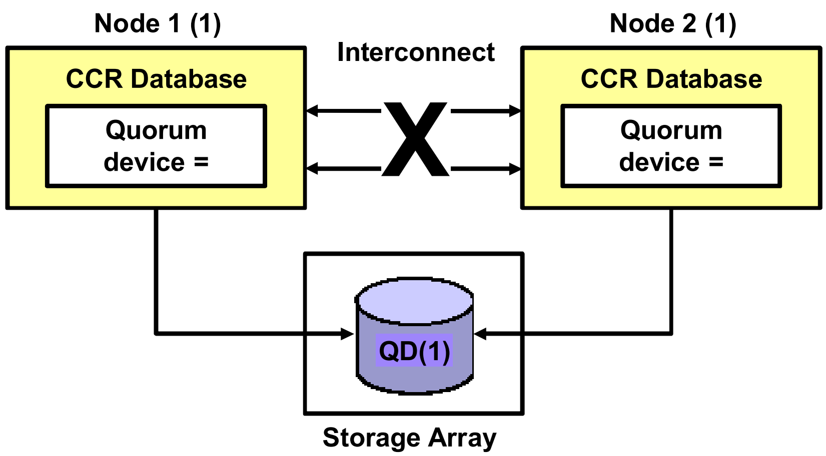
- Failure fencing
- Amnesia prevention

Given the rules for quorum voting, it is clear by looking at a simple two-node cluster why you need extra quorum device votes. If a two-node cluster had only node votes, you must have both nodes booted to run the cluster. This defeats one of the major goals of the cluster, which is to be able to survive node failure. But why have quorum voting at all? If there were no quorum rules, you could run as many nodes in the cluster as were able to boot at any point in time. However, the quorum vote and quorum devices solve the following two major problems:

- Failure fencing
- Amnesia prevention

These are two distinct problems that are solved by the quorum mechanism in the Solaris Cluster software. These problems are discussed in the following slides.

Failure Fencing



If interconnect communication between nodes ceases, either because of a complete interconnect failure or a node crashing, each node must assume that the other is still functional. This is called *split-brain operation* or *split-brain syndrome*. Two separate clusters cannot be allowed to exist because of the potential for data corruption. Each node tries to establish a cluster by gaining another quorum vote. Both nodes attempt to reserve the designated quorum device. The first node to reserve the quorum device establishes a majority and remains as a cluster member. The node that fails the race to reserve the quorum device aborts the Oracle Solaris Cluster software because it does not have a majority of votes.

Amnesia Prevention

- If Node 2 is down, it could “miss” cluster configuration updates.
- To prevent the cluster from forming with a “stale” copy of the cluster configuration repository (CCR), prevent Node 2 from booting as the first node in the cluster after shutting down the cluster.

If it is allowed to happen, a cluster amnesia scenario would involve one or more nodes being able to form a cluster (boot first in the cluster) with a stale copy of the cluster configuration. Consider the following scenario:

1. In a two-node cluster (Node 1 and Node 2), Node 2 is halted for maintenance or crashes.
2. Cluster configuration changes are made on Node 1.
3. Node 1 is shut down.
4. You try to boot Node 2 to form a new cluster. If this is allowed, the cluster would lose the configuration changes.

Quorum Device Rules

- Quorum device must be available to both nodes in a two-node cluster.
- Quorum device information is maintained locally in the CCR.
- Disk quorum device can contain user data.
- Maximum and optimal quorum disk votes are N – 1.
- Quorum device is not required if there are more than two nodes, but it is still recommended.
- A single-disk quorum can be automatically configured by `scinstall` for a two-node cluster.
- Others are configured manually.
- Disk quorum devices are configured (specified) by using DID devices.

Quorum Mathematics and Consequences

- Required number of votes is greater than half of the total possible votes.
- A node that is not able to get the required votes at boot time freezes and waits for more nodes to boot.
- A node falling below the required votes after boot time kernel panics.

When the cluster is running, it is always aware of the following:

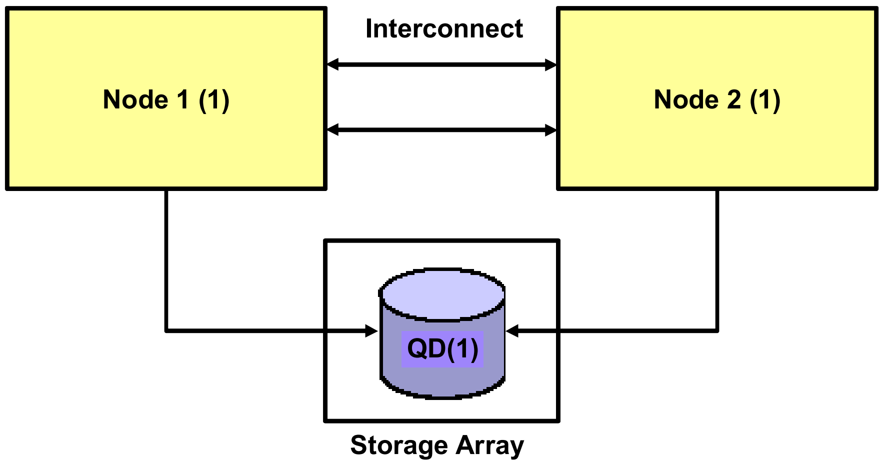
- The total possible quorum votes (number of nodes plus the number of disk quorum votes defined in the cluster)
- The total present quorum votes (number of nodes booted in the cluster plus the number of disk quorum votes physically accessible by those nodes)
- The total needed quorum votes, which is greater than 50 percent of the possible votes

The consequences are the following:

- A node will freeze if it cannot find the needed number of votes at boot time, waiting for other nodes to join to obtain the needed vote count.
- A node will kernel panic if it is booted in the cluster but can no longer find the needed number of votes.

Two-Node Cluster Quorum Devices

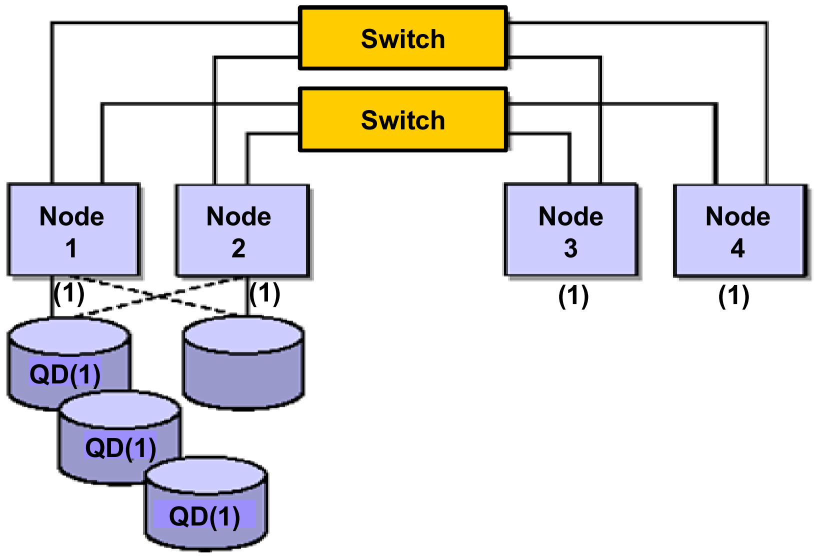
Two-node cluster requires a single quorum device.



A two-node cluster requires a single quorum device, which is typically a quorum disk. The total votes are three. With the quorum disk, a single node can start clustered operation with a majority of votes (two votes, in the example shown in the slide).

Pair+N Quorum Disks

With three quorum devices, Node 1 or Node 2 can form the cluster alone.

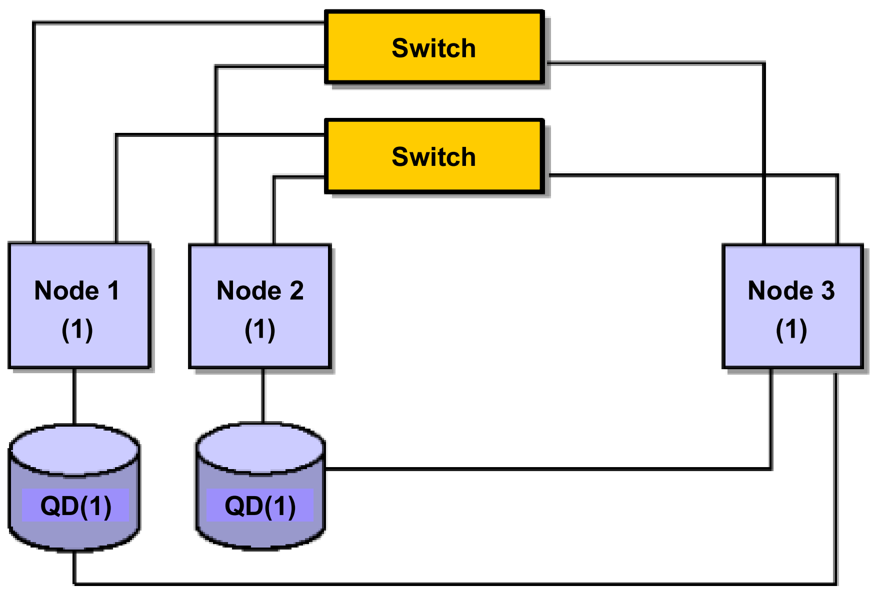


A typical quorum disk configuration in a Pair+2 configuration is shown in the figure. Three quorum disks are used.

The following is true for the Pair+N configuration:

- There are three quorum disks.
- There are seven possible votes.
- A quorum is four votes.
- Nodes 3 and 4 do not have access to any quorum devices.
- Nodes 1 or 2 can start clustered operation by themselves.
- Up to three nodes can fail (Nodes 1, 3, and 4 or Nodes 2, 3, and 4), and clustered operation can continue.

N+1 Quorum Disks



The N+1 configuration requires a different approach. Node 3 is the failover backup for both Node 1 and Node 2.

The following is true for the N+1 configuration:

- There are five possible votes.
- A quorum is three votes.
- If Nodes 1 and 2 fail, Node 3 can continue.

Quiz

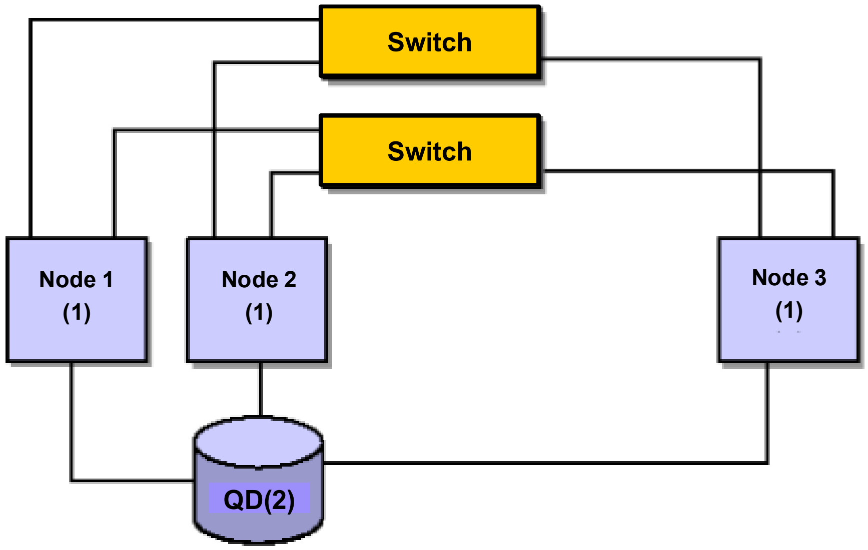
When the cluster is running, it is always aware of which of the following?

a. The total possible quorum votes
b. The total present quorum votes
c. The total needed quorum votes

Answer: a, b, c

Quorum Devices in the Scalable Storage Topology

- Disk votes equal one fewer than the votes from the attached nodes.
- A group reservation mechanism for failure fencing is used.



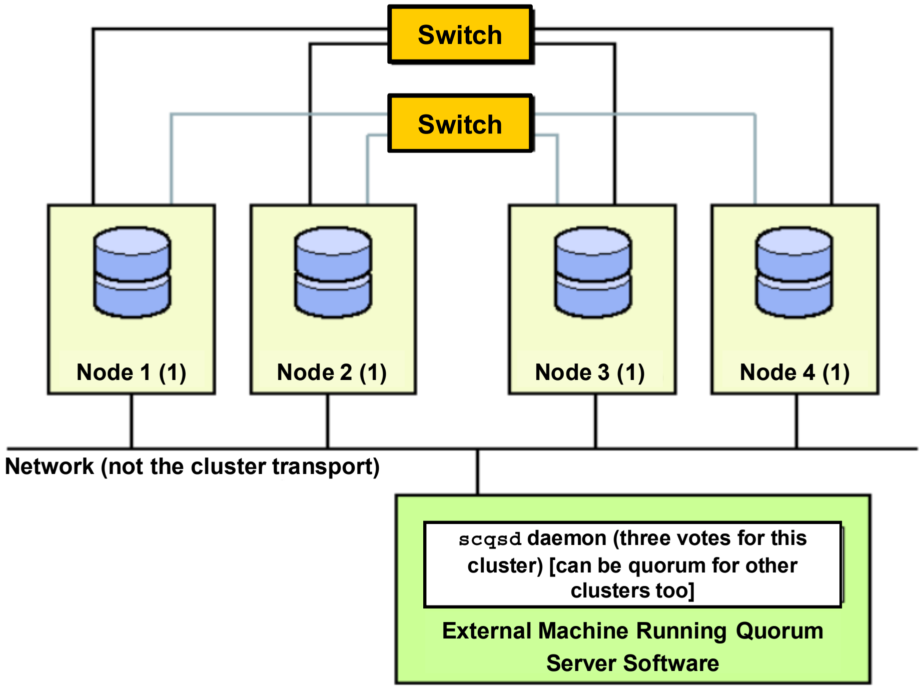
Quorum devices in the scalable storage topology differ significantly from those in any other topology. The following is true for the quorum devices in the scalable storage topology:

- The single quorum device has a vote count equal to the votes of the nodes directly attached to it minus one.

  Note: This rule is universal. In all the previous examples, there were two nodes (with one vote each) directly connected to the quorum device, so that the quorum device had one vote.
- The mathematics and consequences still apply.
- A reservation is performed by using a SCSI-3 persistent group reservation (which is discussed in more detail later in this lesson).
- If, for example, Nodes 1 and 3 can intercommunicate but Node 2 is isolated, Node 1 or Node 3 can reserve the quorum device on behalf of both of them.

  Note: It would seem that in the same race, Node 2 could win and eliminate both Nodes 2 and 3. The topic titled “Intentional Reservation Delays for Partitions with Fewer Than Half of the Nodes,” later in this lesson, shows why this is unlikely.

Quorum Server as Quorum Devices



Oracle Solaris Cluster introduced a new kind of quorum device called a quorum server quorum device. The quorum server software is installed on some machine external to the cluster. A quorum server daemon (`scqsd`) runs on this external machine. The daemon essentially takes the place of a directly connected quorum disk.

Characteristics of the quorum server quorum device:

- The same quorum server daemon can be used as a quorum device for an unlimited number of clusters.
- The quorum server software must be installed separately on the server (external side).
- No additional software is necessary on the cluster side.

- A quorum server is especially useful when there is a great physical distance between quorum nodes. It would be an ideal solution for a cluster that uses the data replication topology.
- A quorum server can be used on any cluster where you prefer the logic of having a single-cluster quorum device with quorum device votes automatically assigned to be one fewer than the node votes.

  For example, with a clustered pairs topology, you might prefer the simplicity of a quorum server quorum device. In that example, any single node could boot into the cluster by itself, if it could access the quorum server. Of course, you might not be able to run clustered applications unless the storage for a particular application is also available, but those relationships can be controlled properly by the application resource dependencies.

Agenda

- Preparing the Oracle Solaris OS environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Preventing Cluster Amnesia with Persistent Reservations

- Oracle Solaris Cluster prevents a node from booting first in the cluster if it was not part of the most recent cluster.
- This prevents amnesia.
- The node that is not part of the most recent cluster may not have the latest CCR.
- This is implemented with quorum-persistent reservations.

Quorum devices in the Oracle Solaris Cluster software environment are used not only as a means of failure fencing but also as a means to prevent cluster amnesia.

Earlier, you reviewed the following scenario:

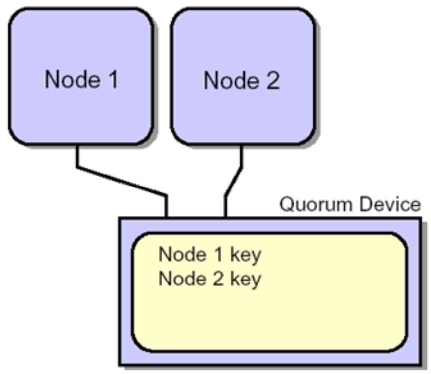
1. In a two-node cluster (Node 1 and Node 2), Node 2 is halted for maintenance.
2. Meanwhile Node 1, which is running fine in the cluster, makes all sorts of cluster configuration changes (new device groups, resource groups).
3. Now Node 1 is shut down.
4. You try to boot Node 2 to form a new cluster.

In this simple scenario, the problem is that if you were allowed to boot Node 2 at the end, it would not have the correct copy of the CCR. Node 2 would have to use the copy that it has (because there is no other copy available) and you would lose the changes to the cluster configuration made in Step 2.

The Oracle Solaris Cluster software quorum involves persistent reservations that prevent Node 2 from booting into the cluster. It is not able to count the quorum device as a vote. Therefore, Node 2 waits until the other node boots to achieve the correct number of quorum votes.

Persistent Reservations and Reservation Keys

- A unique 64-bit key is assigned to each node.
- The key is written to the quorum device when the node is part of a cluster.
  - This set of keys is collectively called the *registered keys*.
- One node is the “reservation holder of record,” but it has more rights than any other node that is registered.



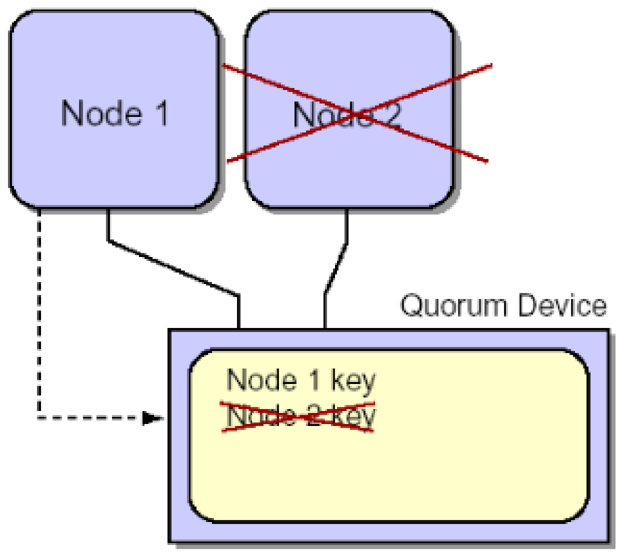
A persistent reservation means that reservation information on a quorum device will survive:

- Even if all nodes connected to the device are reset
- Even after the quorum device itself is powered on and off

Clearly, this involves writing some type of information on the disk itself. The information is called a *reservation key* and is as follows:

- Each node is assigned a unique 64-bit reservation key value.
- Every node that is physically connected to a quorum device has its reservation key physically written onto the device. This set of keys is collectively known as the *registered keys* on the device.
- Exactly one node’s key is recorded on the device as the reservation holder, but this node has no special privileges greater than any other registrant. You can think of the reservation holder as the last node to ever manipulate the keys, but the reservation holder can later be fenced out by another registrant.

Persistent Reservations and Reservation Keys



If Node 1 needs to fence out Node 2 for any reason, it will preempt Node 2's registered key off of the device. If a node's key is preempted from the device, it is fenced from the device. If there is a split brain, each node is racing to preempt the other's key.

Now the rest of the equation is clear. The reservation is persistent, so if a node is booting into the cluster, a node cannot count the quorum device vote unless its reservation key is already registered on the quorum device. Therefore, in the scenario illustrated in the previous paragraph, if Node 1 subsequently goes down so there are no remaining cluster nodes, only Node 1's key remains registered on the device. If Node 2 tries to boot first into the cluster, it will not be able to count the quorum vote, and must wait for Node 1 to boot.

After Node 1 joins the cluster, it can detect Node 2 across the transport and add Node 2's reservation key back to the quorum device so that everything is equal again. A reservation key only gets added back to a quorum device by another node in the cluster whose key is already there.

SCSI-2 and SCSI-3 Reservations

The default policy is `prefer3`.

- For two paths, use SCSI-2.
- For three or more paths, use SCSI-3.
- You can change policy (affecting two-path disks).

Oracle Solaris Cluster supports both SCSI-2 and SCSI-3 disk reservations. The default policy is `prefer3`.

- For disks to which there are exactly two paths, use SCSI-2.
- For disks to which there are more than two paths (for example, any disk with physical connections from more than two nodes), you must use SCSI-3.

The following slides outline the differences between SCSI-2 and SCSI-3 reservations.

SCSI-2 and SCSI-3 Reservations

SCSI-2 reservations and SCSI-2 Persistent Group Reservation Emulation (PGRE):

- Actual SCSI-2 reservation has no persistence.
- SCSI-2 knows nothing about “registered keys.”
- You must add PGR Emulation (PGRE).
  - Reservation keys are emulated in cluster software.
  - Reservation keys are written on private cylinders of disk.
  - Normal SCSI-2 reservation is still used for split-brain “race” (nonpersistent), and then PGRE is used by the winner.

SCSI-2 reservations themselves provide a simple reservation mechanism (the first one to reserve the device fences out the other one), but it is not persistent and does not involve registered keys. In other words, SCSI-2 is sufficient to support the fencing goals in Oracle Solaris Cluster, but does not include the persistence required to implement amnesia prevention.

To implement amnesia prevention by using SCSI-2 quorum devices, Solaris Cluster must make use of Persistent Group Reservation Emulation (PGRE) to implement the reservation keys. PGRE has the following characteristics:

- The persistent reservations are not supported directly by the SCSI-2 command set. Instead, they are emulated by the Solaris Cluster software.
- Reservation keys are written (by the Solaris Cluster software, not directly by the SCSI reservation mechanism) on private cylinders of the disk (cylinders that are not visible in the format command, but are still directly writable by the Solaris OS).

  The reservation keys have no impact on using the disk as a regular data disk, where you will not see the private cylinders.
- The race (for example, in a split-brain scenario) is still decided by a normal SCSI-2 disk reservation. It is not really a race to eliminate the other's key, it is a race to do a simple SCSI-2 reservation. The winner will then use PGRE to eliminate the other's reservation key.

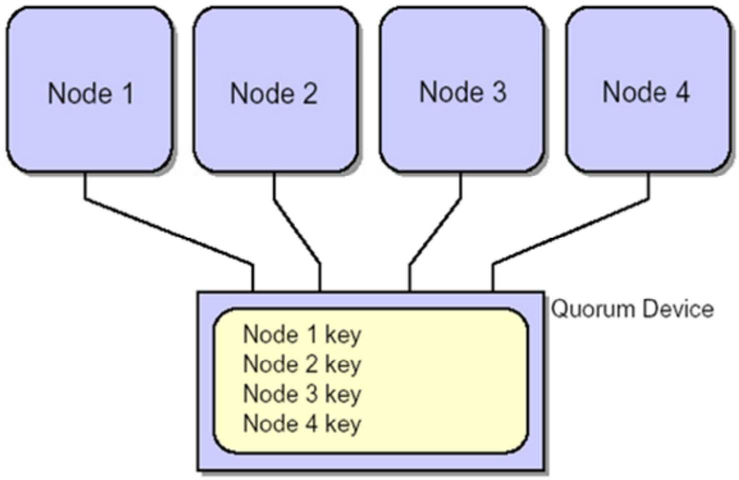
SCSI-3 Persistent Group Reservation (PGR)

- Persistent reservations are implemented directly.
- Key removal is not a separate step from reservation.
- SCSI-3 reservations are generally simpler in the cluster (only one step).
- SCSI-3 must be used when there are more than two paths.

SCSI-3 reservations have the persistent group reservation (PGR) mechanism built in. They have the following characteristics:

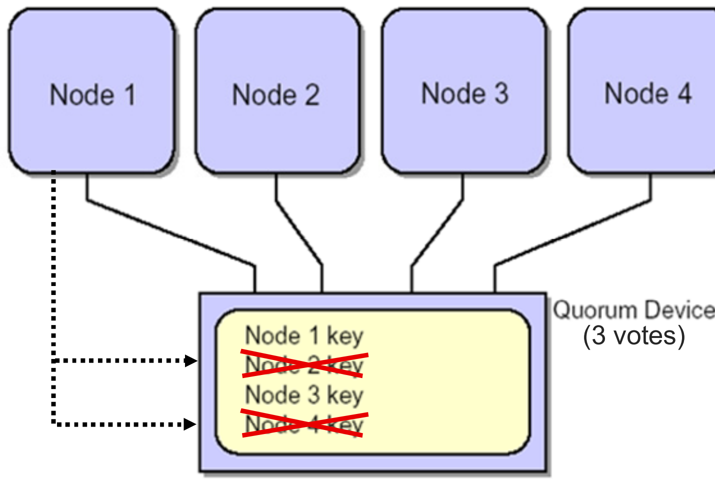
- The persistent reservations are implemented directly by the SCSI-3 command set. Disk firmware itself must be fully SCSI-3 compliant.
- Removing another node’s reservation key is not a separate step from physical reservation of the disk, as it is in SCSI-2. With SCSI-3, the removal of the other node’s key is both the fencing and the amnesia prevention.
- SCSI-3 reservations are generally simpler in the cluster because everything that the cluster needs to do (both fencing and persistent reservations to prevent amnesia) is done directly and simultaneously with the SCSI-3 reservation mechanism.
- With more than two disk paths (that is, any time more than two nodes are connected to a device), Oracle Solaris Cluster must use SCSI-3.

SCSI-3 PGR Scenario with More Than Two Nodes



The figure in the slide shows that four nodes are all physically connected to a quorum drive. Remember that the single quorum drive has three quorum votes.

SCSI-3 PGR Scenario with More Than Two Nodes



Now, imagine that, because of multiple transport failures, there is a partitioning where Nodes 1 and 3 can see each other over the transport and Nodes 2 and 4 can see each other over the transport. In each pair, the node with the lower reservation key tries to eliminate the registered reservation key of the other pair. The SCSI-3 protocol assures that only one pair will remain registered (the operation is atomic).

In the diagram, Node 1 has successfully won the race to eliminate the keys for Nodes 2 and 4. Because Nodes 2 and 4 have their reservation key eliminated, they cannot count the three votes of the quorum device. Because they fall below the needed quorum, they will cause kernel panic.

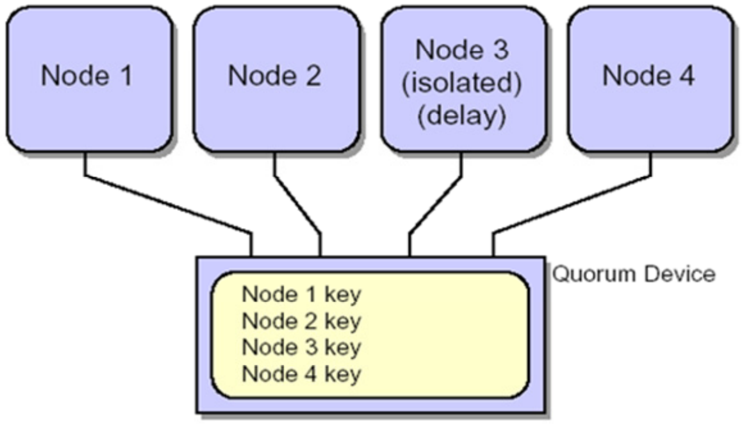
Cluster amnesia is avoided in the same way as in a two-node quorum device. If you now shut down the whole cluster, Node 2 and Node 4 cannot count the quorum device because their reservation key is eliminated. They must wait for either Node 1 or Node 3 to join. One of those nodes can then add back reservation keys for Node 2 and Node 4.

NAS Quorum and Quorum Server Persistent Reservations

- Provides reservation key–based persistent reservations
- Maintains keys persistently on the server side
- Survives rebooting of server side

Both NAS quorum and quorum server provide reservation key–based persistent emulations. Fencing and amnesia prevention is provided in an analogous way to show they are provided with a SCSI-3 quorum device. In both implementations, the keys are maintained in a persistent fashion on the server side; that is, the state of the registration keys recorded with the quorum device survives rebooting of both the cluster nodes and the quorum server device.

Intentional Reservation Delays for Partitions with Fewer Than Half of the Nodes



ORACLE

The diagram in the slide shows the scenario just presented, but three nodes can talk to each other while the fourth is isolated on the cluster transport.

Is there anything to prevent the lone node from eliminating the cluster keys of the other three and making them all kernel panic?

In this configuration, the lone node intentionally delays before racing for the quorum device. The only way it can win is if the other three nodes are really dead, or if each is isolated and delaying the same amount. The delay is implemented when the number of nodes that a node can see on the transport (including itself) is fewer than half the total nodes.

Agenda

- Preparing the Oracle Solaris OS environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Data Fencing

The surviving node fences all shared disks. This eliminates any timing-related danger in taking over data.

- With the `prefer3` policy, data fencing uses SCSI-2 (two-path disks) or SCSI-3.
- You can change the policy to use SCSI-3, even for two-path disks.
- If you use SCSI-2, data fencing is just the reservation (no PGRE).

ORACLE

As an extra precaution, nodes that are eliminated from the cluster because of quorum problems also lose access to all shared data devices. The reason for this is to eliminate a potential timing problem. The node or nodes that remain in the cluster have no idea whether the nodes being eliminated from the cluster are actually still running. If they are running, they will have a kernel panic (after they recognize that they have fallen beneath the required quorum votes). However, the surviving node or nodes cannot wait for the other nodes to kernel panic before taking over the data. The reason that nodes are being eliminated is that there has been a communication failure with them.

To eliminate this potential timing problem, which could otherwise lead to data corruption, before a surviving node or nodes reconfigure the data and applications, the `prefer3` policy fences the eliminated node or nodes from all shared data devices, in the following manner:

- With the `prefer3` policy, SCSI-2 reservation is used for two-path devices and SCSI-3 for devices with more than two paths.
- You can change the policy to use SCSI-3 even if there are only two paths.
- If you do use the default SCSI-2 for a two-path device, data fencing is just the reservation and does not include any PGRE.
- Data fencing is released when a fenced node is able to boot successfully into the cluster again.

Optional Data Fencing

- You can turn off data fencing either per disk or globally. This is:
  - Intended for support of SATA disks
  - Recommended to keep fencing on for disks that support fencing
- Quorum on disk with no fencing:
  - “Software quorum” (race for quorum done without SCSI protocols)
  - Persistent fencing by using PGRE (just like SCSI-2 quorum)

You can disable fencing, either on a disk-by-disk basis or globally in the cluster.

You look at how this can be done in later lessons. It is highly recommended that you keep the fencing on normal SCSI-capable shared disks.

With the new option to disable fencing, Oracle Solaris Cluster can support SATA disks that are incapable of either SCSI-2 or SCSI-3 fencing in any cluster, or disks incapable of SCSI-3 fencing in a cluster where more than two nodes are connected to the storage. Oracle Solaris Cluster can also support access to a storage device from servers outside of the cluster, if fencing is disabled on the device.

Quorum Device on a Disk with No Fencing

Oracle Solaris Cluster can support a quorum device on a disk on which it is doing neither SCSI-2 nor SCSI-3 fencing. Solaris Cluster will implement a “software” reservation process, whereby races for the quorum devices can be decided atomically and reliably without use of any SCSI-2 or SCSI-3 protocols. The persistent reservation for a disk with no fencing works exactly the same way as a disk on which you are doing SCSI-2 fencing (the persistent reservation is emulated by using PGRE).

Quiz

A lone node intentionally delays before racing for the quorum device when the number of nodes that the node can see on the transport (including itself) is:

a. Fewer than half the total nodes
b. More than half the total nodes

Answer: a

Agenda

- Preparing the Oracle Solaris OS environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Configuring a Cluster Interconnect

There are two variations of cluster interconnects:

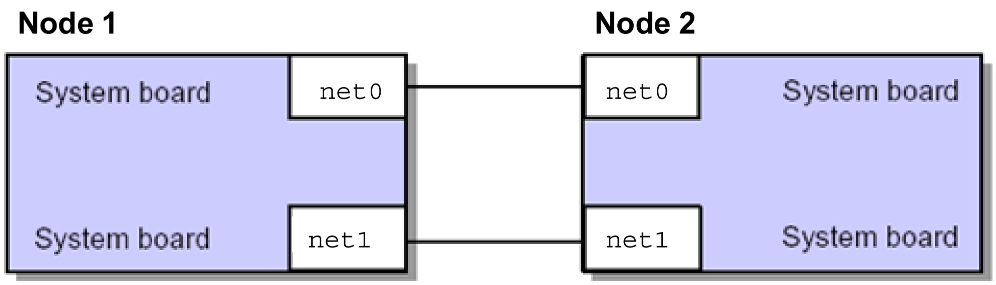
- Point-to-point cluster interconnect
- Switch-based cluster interconnect

ORACLE

Types of Cluster Interconnects

Point-to-point cluster interconnect

- Point-to-point is for a two-node cluster only.
- Provide adapter names when you configure the cluster by using `scinstall`.
- You should usually use auto-discovery.
  - Provide names correctly on first node and auto-discover for others.



There are two variations of cluster interconnects: point-to-point and switch-based.

Point-to-Point Cluster Interconnect

In a two-node cluster, you can directly connect interconnect interfaces by using crossover cables. The figure shows a point-to-point interconnect configuration by using 100BASE-T interfaces.

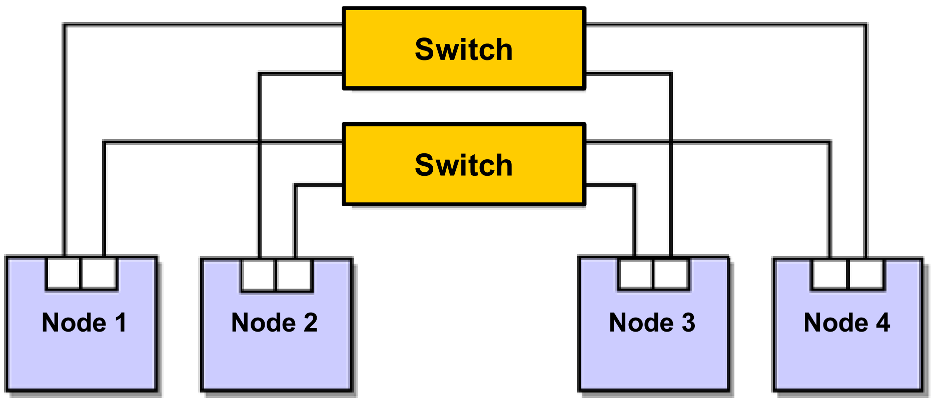
During the Oracle Solaris Cluster software installation, you must provide the names of the end-point interfaces for each cable.

Note: If you provide the wrong interconnect interface names during the initial Solaris Cluster software installation, the first node is installed without errors, but when you try to manually install the second node, the installation hangs. You have to correct the cluster configuration error on the first node and then restart the installation on the second node.

Types of Cluster Interconnects

Switch-based cluster interconnect

- Switches are required for more than two nodes.
- Identify adapters by using `scinstall`.



In cluster configurations with more than two nodes, you must join the interconnect interfaces by using switches. You can also use switches to join two-node cluster interconnects to prepare for the expansion of the number of nodes at a later time. A typical switch-based interconnect is shown in the figure in the slide.

During the Oracle Solaris Cluster software installation, you are asked whether the interconnect system uses switches. If you answer yes, you must provide names for each of the switches.

Note: If you specify more than two nodes during the initial portion of the Solaris Cluster software installation, the use of switches is assumed.

Cluster Transport Interface Addresses and Netmask

- Installation default is the IP range starting with `172.16.0.0`.
  - Recommended: Keep it unless it causes a conflict.
  - It is perfectly fine for multiple clusters to use the same address.
- The `netmask` property defines the extent of the range.
- The `netmask` default is `255.255.240.0`.
  - 12-bit range dedicated for cluster transport
  - `172.16.0.0` – `172.16.15.255`

During the Oracle Solaris Cluster software installation, the cluster interconnects are assigned IP addresses based on a base address of `172.16.0.0`. If necessary, you can override the default address, but this is not recommended. Uniform addresses can be a benefit during problem isolation.

The `netmask` property associated with the entire cluster transport describes, together with the base address, the entire range of addresses associated with the transport. For example, if you used the default base address of `172.16.0.0` and the default netmask of `255.255.240.0`, you would be dedicating a 12-bit range (`255.255.240.0` has 12 zeros at the end) to the cluster transport. This range is from `172.16.0.0` to `172.16.15.255`.

Note: When you set `255.255.240.0` as the cluster transport netmask, you will not see this netmask actually applied to any of the private network adapters. Once again, the cluster uses this netmask to define the entire range that it has access to, and then subdivides the range even further to cover the multiple separate networks that make up the cluster transport.

Choosing the Cluster Transport Netmask

Specify the maximum number of:

- Private networks
- Cluster nodes
- Virtual clusters

While you can choose the cluster transport netmask by hand, the cluster prefers instead that you specify:

- The maximum anticipated number of private networks
- The maximum anticipated number of nodes
- The maximum anticipated number of virtual clusters (**Note:** *Virtual clusters* is another term for Solaris Containers clusters.)

Note: In Oracle Solaris Cluster, if you want to restrict private network addresses with a class C–like space, similar to `192.168.5.0`, you can do it easily, even with relatively large numbers of nodes and subnets and Solaris Container clusters.

Identifying Cluster Transport Interfaces

1. Display information about the physical attributes of datalinks currently on the system.

```
# dladm show-phys
LINK      MEDIA       STATE     SPEED  DUPLEX   DEVICE
net1      Ethernet    unknown   0      unknown  e1000g1
net2      Ethernet    unknown   0      unknown  e1000g2
net0      Ethernet    up        1000   full     e1000g0
net3      Ethernet    unknown   0      unknown  e1000g3
```

2. Display information about datalinks currently on the system.

```
# dladm show-link
LINK      CLASS     MTU     STATE     OVER
net1      phys      1500    unknown   --
net2      phys      1500    unknown   --
net0      phys      1500    up        --
net3      phys      1500    unknown   --
```


Identifying network interfaces is not an easy task.

To accurately determine the logical name of each interface on a system:

1. Display information about the physical attributes of datalinks currently on the system:

   # dladm show-phys

2. Display information about datalinks currently on the system:

   # dladm show-link

Identifying Cluster Transport Interfaces

3. Choose a network interface.
   – Not one that is already configured
   – The network interface might be a private network or (unconfigured) public network, or might be not attached at all.

3. Choose a network interface (at this point, it might be an actual private network, or one ready to be set up as a secondary public network, or when not connected to anything at all).

Identifying Cluster Transport Interfaces

4. Make sure that the selected network interface is not actually on the public network.
   a. In one window, run the `snoop` command for the interface:

```
# ipadm create-ip net1
# snoop -d net1
<..hope to see no output here..>
```

   b. In another window, `ping` the public broadcast address, and make sure no traffic is seen by the selected network interface:

```
# ping public_net_broadcast_address
```

If you do see `snoop` output, you are looking at a public network adapter. Do not continue. Just return to Step 3 and pick a new network interface.

Identifying Cluster Transport Interfaces

5. Configure the IP interface with a valid IP address for testing:

```
# ipadm create-addr -T static -a 192.168.1.1/24 net1/v4static
```

6. Use the corresponding IP on the other node.
7. Try to `ping` across each private network.
8. Delete the interface before installing the cluster. If it is up, the installation will fail.

```
# ipadm delete-ip net1
```

9. Repeat for additional private networks.

ORACLE

5. Now that you know your adapter is not on the public net, check to see whether it is connected on a private net. Make up some unused subnet address just to test out interconnectivity across a private network. Do not use addresses in the existing public subnet space.

```
# ipadm create-addr -T static -a 192.168.1.1/24 net2/v4static
```

6. Perform Steps 4 and 5 to try to guess the matching network interface on the other node. Choose a corresponding IP address (for example, `192.168.1.2`).
7. Test that the nodes can `ping` across each private network, as in the following example:

```
# ping 192.168.1.2
192.168.1.2 is alive
```

8. After you have identified the new network interfaces, delete them. Cluster installation fails if your transport network interfaces are still up from testing.

```
# ipadm delete-ip net1
```

9. Repeat Steps 3 through 8 with network interfaces for the second cluster transport. Repeat again if you are configuring more than two cluster transports.

Agenda

- Preparing the Oracle Solaris OS environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Identifying Public Network Adapters

- The public network must be managed by IPMP.
- Public network configuration is not part of cluster framework installation.
- Make sure that you distinguish the public network from the private transport.
- Use the `snoop` and `ping` commands to see whether public network broadcasts are visible.

You will not be asked about public network configuration when you are installing the cluster.

The public network interfaces must be managed by IPMP, which can be administered either before or after cluster installation.

Because you are identifying your private transport interfaces before cluster installation, it can be useful to identify your public network interfaces at the same time, so as to avoid confusion.

Your primary public network adapter should be the only one currently configured on the public network. You can verify this with the following command:

```
# dladm show-link
# ipadm show-if
```

You can verify your secondary public network adapter, if applicable, by making sure that:

- It is not one of those that you identified to be used as the private transport
- It can `snoop` public network broadcast traffic

```
# ipadm create-ip net2
# snoop -d net2
(other window or node)
# ping -s pubnet_broadcast_addr
```

Agenda

- Preparing the Oracle Solaris OS environment
- Configuring the Oracle Solaris Cluster storage connections
- Describing quorum votes and quorum devices
- Describing persistent quorum reservations and cluster amnesia
- Describing data fencing
- Configuring a cluster interconnect
- Identifying public network adapters
- Configuring shared physical adapters

Configuring Shared Physical Adapters

- Tagged VLAN with *`Vlan_id`* `>=1`
- The instance that you see is:
  `(1000*Vlan_id) + phys-instance`
  - Example: `net1` using the `Vlan` IDs `3` and `5`
    - Appears as `net3001` and `net5001`
- Let `scinstall` configure a private VLAN.

Recall that certain adapters are capable of participating in 802.1q-tagged VLANs, and can be used as both private and public network adapters assuming that the switches are also capable of tagged VLANs. This enables blade architecture servers that have only two physical network adapters to be clustered and to still have redundant public and private networks.

An adapter that is participating in a tagged VLAN configuration is assigned an instance number, `1000*(Vlan_identifer) + physical_instance_number`.

For example, if you have a physical adapter `net1`, and it is participating in a tagged VLAN with ID `3` as its public network personality, and a tagged VLAN with ID `5` as its private network personality, then it appears as if it were two separate adapters, `net3001` and `net5001`.

Configuring the Public Network

To configure a shared adapter's public network personality, all you have to do is configure the adapter instance according to the instance number mathematical formula. VLAN ID `3` is going to be used for the public network identity of what would otherwise be `net1`. When instance `ce3001` is plumbed, the adapter driver understands that it is using tagged VLAN ID `3` on physical instance number `1`.

Allocating a Different VLAN ID for the Private Network

You should never configure the private network ID manually. Instead, you should perform the initial configuration by using the `scinstall` utility. This procedure is documented in the lesson titled "Installing and Configuring the Oracle Solaris Cluster Software." All you need to do is ensure that you have different VLAN IDs for the public and private networks. The `scinstall` utility automatically detects a tagged VLAN-capable adapter and queries for the private VLAN ID.

Summary

In this lesson, you should have learned how to:

- Prepare the Oracle Solaris OS environment
- Configure the Oracle Solaris Cluster storage connections
- Describe the quorum votes and quorum devices
- Describe persistent quorum reservations and cluster amnesia
- Describe data fencing
- Configure a cluster interconnect
- Identify public network adapters
- Configure shared physical adapters

Practice 4 Overview: Preparing for Installation

This practice covers the following topics:

- Task 1: Verifying the Oracle Solaris 11 environment
- Task 2: Identifying a cluster topology
- Task 3: Selecting quorum devices
- Task 4: Verifying the cluster private interconnect configuration
- Task 5: Selecting the public network interfaces

5

Installing and Configuring the Oracle Solaris Cluster Software

Objectives

After completing this lesson, you should be able to:

- Identify cluster install package groups
- Identify the prerequisites for installing the Oracle Solaris Cluster software
- Install the Oracle Solaris Cluster software
- Set the root environment
- Configure the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Understand settings that are automatically configured by `scinstall`
- Perform automatic quorum configuration
- Describe manual quorum selection
- Perform post-installation verification

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the Root Environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Identifying Cluster Install Package Groups

Primary group packages for the Oracle Solaris Cluster 4.0 software are:

- `ha-cluster-full`
- `ha-cluster-framework-full`
- `ha-cluster-data-services-full`
- `ha-cluster-minimal`
- `ha-cluster-framework-minimal`

Identifying Cluster Install Package Groups

Feature	ha-cluster-full	ha-cluster-framework-full	ha-cluster-data-services-full	ha-cluster-minimal	ha-cluster-framework-minimal
Framework	X	X	X	X	X
Agents	X	X	X		
Localization	X	X			
Framework man pages	X	X			
Data service man pages	X		X		
Agent builder	X	X			
Generic data service	X	X	X		

The table shown in the slide lists the primary group packages for the Oracle Solaris Cluster 4.0 software and the principal features that each group package contains. You must install at least the `ha-cluster-framework-minimal` group package.

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the Root Environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Prerequisites for Installing the Oracle Solaris Cluster Software

- Boot disks must be configured according to Oracle Solaris Cluster standards.
- The Oracle Solaris Operating System (OS) and OS Service Repository Updates (SRUs) must be installed.

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the Root Environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Installing the Oracle Solaris Cluster Software

Perform the following tasks before you begin installation:

- Ensure that the Oracle Solaris OS is installed to support Oracle Solaris Cluster software.
- Choose which Oracle Solaris Cluster software packages to install from the package group.

Installing the Oracle Solaris Cluster Software

1. Display a console screen for each node in the cluster if you are using a cluster administrative console.
   – As superuser, use the following command to start the `pconsole` utility:

```
adminconsole# pconsole host[:port] […]  &
```


Perform the following steps to complete the Oracle Solaris cluster installation:

1. If you are using a cluster administrative console, display a console screen for each node in the cluster.
   - If `pconsole` software is installed and configured on your administrative console, use the `pconsole` utility to display the individual console screens.
   - As superuser, use the following command to start the `pconsole` utility:

     ```
     adminconsole# pconsole host[:port] […]  &
     ```

     The `pconsole` utility also opens a master window from which you can send your input to all individual console windows at the same time.
   - If you do not use the `pconsole` utility, connect to the consoles of each node individually.

Installing the Oracle Solaris Cluster Software

2. Restore external access to remote procedure call (RPC) communication.

```
# svccfg
svc:> select network/rpc/bind
svc:/network/rpc/bind> setprop config/local_only=false
svc:/network/rpc/bind> quit
# svcadm refresh network/rpc/bind:default
# svcprop network/rpc/bind:default | grep local_only
```

Note: The output of the last command should show that the `local_only` property is now set to `false`.

2. Restore external access to remote procedure call (RPC) communication.

   During the installation of the Oracle Solaris OS, a restricted network profile that disables external access for certain network services is used. The restricted services include the RPC communication service, which is required for cluster communication.

   Perform the commands shown in the slide to restore external access to RPC communication.

Installing the Oracle Solaris Cluster Software

3. Become superuser on the cluster node to install.
4. Disable Network Auto-Magic (NWAM).
   – To disable NWAM, you enable the `defaultfixed` profile:

```
# netadm enable -p ncp defaultfixed
# netadm list -p ncp defaultfixed
```


3. Become superuser on the cluster node to install.

   Alternatively, if your user account is assigned the System Administrator profile, issue commands as nonroot through a profile shell, or prefix the command with the `pfexec` command.
4. Disable Network Auto-Magic (NWAM).

   NWAM activates a single network interface and disables all others. For this reason, NWAM cannot coexist with the Oracle Solaris Cluster software and you must disable it before you configure or run your cluster. To disable NWAM, you enable the `defaultfixed` profile.

```
# netadm enable -p ncp defaultfixed
# netadm list -p ncp defaultfixed
```

Installing the Oracle Solaris Cluster Software

5. Set up the repository for the Oracle Solaris Cluster software packages.
   - If you are using an ISO image of the software, perform the following steps:
     a. Download the Oracle Solaris Cluster 4.0 ISO image from Oracle Software Delivery Cloud at `http://edelivery.oracle.com/`.
     b. Make the Oracle Solaris Cluster 4.0 ISO image available.

```
# lofiadm -a path-to-iso-image
/dev/lofi/N
# mount -F hsfs /dev/lofi/N /mnt
```

     c. Set the location of the Oracle Solaris Cluster 4.0 package repository.

```
# pkg set-publisher -g file:///mnt/repo ha-cluster
```


5. Set the repository for the Oracle Solaris Cluster software packages.

   If you are using an ISO image of the software, perform the following steps:

   a. Download the Oracle Solaris Cluster 4.0 ISO image from Oracle Software Delivery Cloud at `http://edelivery.oracle.com/`.

      Oracle Solaris cluster software is part of the Oracle Solaris Product Pack. Follow online instructions to complete selection of the media pack and download the software. A valid Oracle license is required to access Oracle Software Delivery Cloud.

   b. Make the Oracle Solaris Cluster 4.0 ISO image available.

```
# lofiadm -a path-to-iso-image
/dev/lofi/N
# mount -F hsfs /dev/lofi/N /mnt
```

      Where `-a path-to-iso-image`, specifies the full path and file name of the ISO image

   c. Set the location of the Oracle Solaris Cluster 4.0 package repository.

```
# pkg set-publisher -g file:///mnt/repo ha-cluster
```

Installing the Oracle Solaris Cluster Software

5. Set up the repository for the Oracle Solaris Cluster software packages.
   - If the cluster nodes have direct access to the Internet, perform the following steps:
     a. Go to `http://pkg-register.oracle.com`.
     b. Choose `Oracle Solaris Cluster software`.
     c. Accept the license.
     d. Request a new certificate by choosing `Oracle Solaris Cluster software` and submitting a request.
     e. Download the key and certificate files and install them as described in the returned certification page.
     f. Configure the `ha-cluster` publisher with the downloaded SSL keys, and set the location of the Oracle Solaris Cluster 4.0 repository.

5. Set up the repository for the Oracle Solaris Cluster software packages.

   If the cluster nodes have direct access to the Internet, perform the following steps:

   a. Go to `http://pkg-register.oracle.com`.
   b. Choose `Oracle Solaris Cluster` software.
   c. Accept the license.
   d. Request a new certificate by choosing `Oracle Solaris Cluster software` and submitting a request.
   e. The certification page is displayed with download buttons for the key and the certificate.
   f. Download the key and certificate files and install them as described in the returned certification page.
   g. Configure the `ha-cluster` publisher with the downloaded SSL keys and set the location of the Oracle Solaris Cluster 4.0 repository.

In the following example, the repository name is `https://pkg.oracle.com/`*`repository-location`*`/`.

```
# pkg set-publisher \
-k /var/pkg/ssl/Oracle_Solaris_Cluster_4.0.key.pem \
-c /var/pkg/ssl/Oracle_Solaris_Cluster_4.0.certificate.pem \
-O https://pkg.oracle.com/repository-location/ ha-cluster
```

Where:

- `-k /var/pkg/ssl/Oracle_Solaris_Cluster_4.0.key.pem`
  specifies the full path to the downloaded SSL key file
- `-c /var/pkg/ssl/Oracle_Solaris_Cluster_4.0.certificate.pem`
  specifies the full path to the downloaded certificate file
- `-O https://pkg.oracle.com/repository-location/`
  specifies the URL to the Oracle Solaris Cluster 4.0 package repository
  Note: For more information, see the `pkg`(1) man page.

Installing the Oracle Solaris Cluster Software

6. Ensure that the `solaris` and `ha-cluster` publishers are valid.

```
# pkg publisher
PUBLISHER              TYPE     STATUS   URI
solaris                origin   online   solaris-repository
ha-cluster             origin   online   ha-cluster-repository
```

7. Install the Oracle Solaris Cluster 4.0 software.

```
# /usr/bin/pkg install package
```

8. Verify that the package installed successfully.

```
# pkg info -r package
```


6. Ensure that the `solaris` and `ha-cluster` publishers are valid as shown in the slide.
   Note: For information about setting the `solaris` publisher, see http://www.oracle.com/technetwork/indexes/documentation/index.html#CCOSPrepo_sharenfs2.
7. Install the Oracle Solaris Cluster 4.0 software.
   `# /usr/bin/pkg install` *`package`*
8. Verify that the package is installed successfully.
   `# pkg info -r` *`package`*

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the Root Environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Set the Root Environment

Perform this procedure on each node in the global cluster:

1. Become superuser on a cluster node.
2. Add `/usr/cluster/bin` to the `PATH`.
3. (Optional) Set the same root password on each node, if you have not already done so.

Perform the procedure on each node in the global cluster as in the slide.

Note: Always make `/usr/cluster/bin` the first entry in the `PATH`. This placement ensures that Oracle Solaris Cluster commands take precedence over any other binaries that have the same name, thus avoiding unexpected behavior.

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software by using IPS
- Setting the Root Environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Configuring the Oracle Solaris Cluster Software

- Setting the `installmode` flag
- Automatic quorum configuration
- Automatic reset of `installmode` without quorum devices
- Configuration information required to run `scinstall`
- Variations of interactive `scinstall`
- Configuring the entire cluster at once
- Typical versus custom installation

Configuring the Oracle Solaris Cluster Software

The Oracle Solaris Cluster configuration is performed by using:

- `scinstall` utility
- Automated Installer (AI) method

The Solaris Cluster configuration is performed by one of the two following methods:

- **Using the `scinstall` utility interactively:** This is the most common method of configuring Solaris Cluster, and the only one that is described in detail in this lesson.
- **Automated Installer:** Set up an Automated Installer (AI) install server. Then use the `scinstall` AI option to install the software on each node and establish the cluster.

Setting the `installmode` Flag

Understanding the `installmode` flag:

- As nodes get configured into the cluster:
  - The first node (node ID 1) has a quorum vote of 1
  - All other nodes have a quorum vote of 0
- This enables nodes to reboot into the cluster without losing the cluster quorum.
- If the first node rebooted at this stage, all other nodes would panic.

As you configure Oracle Solaris Cluster software on cluster nodes and reboot the nodes into the cluster, a special flag called the `installmode` flag is set in the cluster CCR. When this flag is set, the following happens:

- The first node installed (node ID 1) has a quorum vote of one.
- All other nodes have a quorum vote of zero.

This enables you to complete the rebooting of the second node into the cluster while maintaining the quorum mathematics rules. If the second node had a vote (making a total of two in the cluster), the first node would kernel panic when the second node was rebooted after the cluster software was installed because the first node would lose operational quorum.

One important side effect of the `installmode` flag is that you must be careful not to reboot the first node (node ID 1) until you can choose quorum devices and eliminate (reset) the `installmode` flag. If you accidentally reboot the first node, all the other nodes will kernel panic because they have zero votes out of a possible total of one.

If the installation is a single-node cluster, the `installmode` flag is not set. Post-installation steps to choose a quorum device and reset the `installmode` flag are unnecessary.

Automatic Quorum Configuration

In a two-node cluster configuration:

- You have the option to have cluster select and configure the quorum device for you
- The shared disk with the lowest DID number will be chosen
- The `installmode` flag will be automatically reset after the quorum device is configured
- Disable automatic configuration if you want to:
  - Choose quorum device yourself
  - Use NAS device
  - Use quorum server

On a two-node cluster only, you have the option of having the `scinstall` utility insert a script that automatically chooses your quorum device as the second node boots into the cluster. The defaults will always be to accept the option.

The quorum device chosen will be the first dual-ported disk or LUN (the one with the lowest DID number).

If you choose to allow automatic quorum configuration, the `installmode` flag is automatically reset after the quorum device is automatically configured.

You can disable the two-node cluster automatic quorum configuration if you want to:

- Choose the quorum device yourself
- Use a NAS device as a quorum device
- Use the quorum server as a quorum device

Automatic Reset of `installmode` Without Quorum Devices

In a cluster with more than two nodes only, the `scinstall` script:

- Automatically resets the `installmode` flag
- Does not choose the quorum device

In clusters with more than two nodes, `scinstall` inserts a script to automatically reset the `installmode` flag. It will not automatically configure a quorum device. If you want a quorum device, you still have to do that manually after the installation. By resetting `installmode`, each node is assigned its proper single quorum vote.

Configuration Information Required to Run `scinstall`

The configuration information required to run `scinstall` includes:

- The name of the cluster and the names of all nodes
- Cluster transport IP network number and netmask
- Cluster transport adapters and switches
- Whether you want DES authentication as nodes join cluster
  - Default (UNIX) authentication has certain dangers.

The following information is required by `scinstall` and should be prepared in advance:

The name of the cluster and the names of all nodes: The cluster name is just a name agreed upon by the nodes, it is not a name that resolves to an IP address.

In Oracle Solaris Cluster Geographic Edition, the cluster name is used as a logical IP name. So, in Oracle Solaris Cluster, it might be a good idea to choose a cluster name that does not conflict with any existing host name.

The nodes in the cluster must be able to resolve each other's host names. If for some reason this is true but the names are not in each node's `/etc/inet/hosts` file (the names could have been resolved through a name server), the `/etc/inet/hosts` file is automatically modified to include these names.

Cluster transport IP network number and netmask: As described in the lesson titled "Exploring Node Console Connectivity and the Cluster Console Software," the default cluster transport IP address range begins with `172.16.0.0` and the netmask is `255.255.240.0`. You should keep the default if it causes no conflict with anything else visible on any other network. It is perfectly fine for multiple clusters to use the same addresses on their cluster transports, because these addresses are not visible anywhere outside the cluster.

Note: The netmask refers to the range of IP addresses that are reserved for all possible cluster transport addresses. This will not match the actual netmask that you will see configured on the transport adapters if you check by using `ifconfig -a`.

If you must specify a different IP address range for the transport, you can do so. Rather than being asked initially for a specific netmask, you will be asked for the anticipated maximum number of nodes and physical private networks. A suggested netmask is then calculated for you.

Cluster transport adapters and switches: You must be prepared to identify transport adapters on at least the first node on which you run `scinstall`. On other nodes you normally let `scinstall` use its auto-discovery feature to automatically determine the transport adapters.

You can define a two-node cluster topology as using switches or just using point-to-point cables. This does not even need to match the actual topology; the software really has no way of telling. It is just the definitions in the Cluster Configuration Repository (CCR), and whichever way you define it will be the way it is presented when you view the cluster configuration by using command-line commands or the graphical web-based administration tool.

Names that you provide for switches are arbitrary, and are used only to match up transport connections between the various nodes.

Port names for specific switch connections are arbitrary except for SCI switches. For SCI, the port name must match the switch port number to which an actual cable is connected.

Using DES authentication for nodes to authenticate with each other as they join the cluster: By default, nodes are authenticated by using standard UNIX authentication as they join the cluster. A reverse IP address lookup is done for a node trying to join the cluster, and if the resulting name is in the list of nodes that you typed in as nodes for this cluster, it is allowed to add itself to the cluster.

The reasons for considering more stringent authentication are the following:

- Nodes that are adding themselves to the cluster communicate across the public network.
- A bogus node adding itself to the cluster can add bogus information to the CCR or copy out the CCR.

DES authentication, also known as *secure remote procedure call* (secure RPC) authentication, is a much stronger authentication that cannot be spoofed by something simple like spoofing IP addresses.

Quiz

Which flag enables you to complete the rebooting of the second node into the cluster while maintaining the quorum mathematics rules?

a. `scinstall`
b. `quorumvote`
c. `installmode`

Answer: c

Variations in Interactive `scinstall`

Variations in interactive `scinstall` include doing the following:

- Configure the entire cluster at once:
  - Typical installation
  - Custom installation
- Configure cluster nodes one at a time:
  - Typical installation
  - Custom installation

Configuring Entire Cluster at Once

- The node that you are driving from becomes the last to join.
- Drive from the node that you want to have highest node ID.
- List the other nodes in reverse order.
- Communication by using RPC is already installed in the package.
- There is no need for `ssh` or `rsh`.

If you choose the option to configure the entire cluster, you run `scinstall` on only one node. You should be aware of the following behavior:

- The node that you are driving from will be the last node to join the cluster, because it needs to configure and reboot all the other nodes first.
- If you care about which node IDs are assigned to the nodes, you should drive from the node that you want to be the highest node ID, and list the other nodes in reverse order.
- The Oracle Solaris Cluster software packages must already be installed on all nodes. Therefore, you do not need remote shell access (neither `rsh` nor `ssh`) between the nodes. The remote configuration is performed by using RPC installed by the Solaris Cluster packages. If you are concerned about authentication, you can use DES authentication.

Configuring Cluster Nodes One at a Time

- Completely configure one node.
- That node boots into the cluster and becomes the “sponsor node.”
- Configure other nodes one at a time if you want predictable node ID.

If you choose this method, you run `scinstall` separately on each node.

You must complete `scinstall` and reboot into the cluster on the first node. This becomes the sponsoring node for the remaining nodes.

If you have more than two nodes, you can run `scinstall` simultaneously on all but the first node, but it might be hard to predict which node gets assigned which node ID. If you care, you should just run `scinstall` on the remaining nodes one at a time, and wait for each node to boot into the cluster before starting the next one.

Typical Versus Custom Installation

Typical installation assumes:

- Default network address and netmask
- Adapter auto-discovery in "all-at-once" method
- Switches named switch1 and switch2
- Standard system authentication

Both the all-at-once and one-at-a-time methods have typical and custom configuration options (to make a total of four variations).

The typical configuration mode assumes the following responses:

- It uses network address `172.16.0.0` with netmask `255.255.240.0` for the cluster interconnect.
- It assumes that you want to perform autodiscovery of cluster transport adapters on the other nodes with the all-at-once method. (On the one-node-at-a-time method, it asks whether you want to use autodiscovery in both typical and custom modes.)
- It uses the names switch1 and switch2 for the two transport switches, and assumes the use of switches even for a two-node cluster.
- It assumes that you want to use standard system authentication (not DES authentication) for new nodes configuring themselves into the cluster.

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the Root Environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Configuring by Using All-at-Once and Typical Modes: Example (1/12)

```
# /usr/cluster/bin/scinstall
*** Main Menu ***

Please select from one of the following (*) options:

  * 1) Create a new cluster or add a cluster node
  * 2) Print release information for this cluster node

  * ?) Help with menu options
  * q) Quit

Option:  1
```


The example shows the full dialog for the Oracle Solaris cluster installation that requires the least information, the all-at-once, Typical mode installation. The example is from a two-node cluster, where the default is to let `scinstall` set up a script that automates configuration of the quorum device. In the example, `scinstall` is running on the node named `clnode1`.

From the `Main Menu`, select option `1`, `Create a new cluster or add a cluster node`.

Configuring by Using All-at-Once and Typical Modes: Example (2/12)

```
*** New Cluster and Cluster Node Menu ***

    Please select from any one of the following options:

        1) Create a new cluster
        2) Create just the first node of a new cluster on this machine
        3) Add this machine as a node in an existing cluster

        ?) Help with menu options
        q) Return to the Main Menu

    Option:  1
```


From the `New Cluster and Cluster Node Menu`, select option `1`, `Create a new cluster`.

Configuring by Using All-at-Once and Typical Modes: Example (3/12)

```
*** Create a New Cluster ***


    This option creates and configures a new cluster.

    Press Control-D at any time to return to the Main Menu.


    Do you want to continue (yes/no) [yes]?  yes

    Checking the value of property "local_only" of service
    svc:/network/rpc/bind ...
    Property "local_only" of service svc:/network/rpc/bind is already
    correctly set to "false" on this node.

Press Enter to continue:
```


Specify `yes` when asked if you want to continue.

The `scinstall` utility now checks whether the value of `local_only` property of `network/rpc/bind` service is correctly set to `false`.

Press the Enter key to continue.

Configuring by Using All-at-Once and Typical Modes: Example (4/12)

```
>>> Typical or Custom Mode <<<
    This tool supports two modes of operation, Typical mode and Custom
    mode. For most clusters, you can use Typical mode. However, you
    might need to select the Custom mode option if not all of the
    Typical mode defaults can be applied to your cluster.

    For more information about the differences between Typical and
    Custom modes, select the Help option from the menu.

    Please select from one of the following options:

        1) Typical
        2) Custom
        ?) Help
        q) Return to the Main Menu

    Option [1]:  1
```


From the Typical or Custom Mode menu, select option `1`, `Typical`.

Configuring by Using All-at-Once and Typical Modes: Example (5/12)

```
>>> Cluster Name <<<

    Each cluster has a name assigned to it. The name can be made up of
    any characters other than whitespace. Each cluster name should be
    unique within the namespace of your enterprise.

    What is the name of the cluster you want to establish?  Cluster1
```


Specify a cluster name, such as `cluster1`.

Configuring by Using All-at-Once and Typical Modes: Example (6/12)

```
>>> Cluster Nodes <<<
    This Oracle Solaris Cluster release supports a total of up to 16
    nodes.

    List the names of the other nodes planned for the initial cluster
    configuration. List one node name per line. When finished, type
    Control-D:

    Node name (Control-D to finish):  clnode2
    Node name (Control-D to finish):  ^D

    This is the complete list of nodes:

        clnode1
        clnode2

    Is it correct (yes/no) [yes]?  yes
```


Specify the name of the other cluster node, such as `clnode2`.

Press Control-D to finish adding the cluster nodes to display the complete list of cluster nodes added and specify `yes` to confirm the listing of the cluster nodes.

Configuring by Using All-at-Once and Typical Modes: Example (7/12)

```
>>> Cluster Transport Adapters and Cables <<<

    You must identify the cluster transport adapters which attach this
    node to the private cluster interconnect.

    Select the first cluster transport adapter:

        1) net1
        2) net2
        3) net3
        4) Other

    Option:  1

    Searching for any unexpected network traffic on "net1" ... done
    Verification completed. No traffic was detected over a 10 second
    sample period.
```


Select the cluster transport adapters that will be used as the cluster private interconnect. In this example, `net1` and `net3` are selected as the cluster private interconnect.

Configuring by Using All-at-Once and Typical Modes: Example (8/12)

```
Select the second cluster transport adapter:

        1) net1
        2) net2
        3) net3
        4) Other

    Option:  3

    Searching for any unexpected network traffic on "net3" ... done
    Verification completed. No traffic was detected over a 10 second
    sample period.

    Plumbing network address 172.16.0.0 on adapter net1 >> NOT
    DUPLICATE ... done
    Plumbing network address 172.16.0.0 on adapter net3 >> NOT
    DUPLICATE ... done
```

Configuring by Using All-at-Once and Typical Modes: Example (9/12)

```
>>> Quorum Configuration <<<

    Every two-node cluster requires at least one quorum device. By
    default, scinstall selects and configures a shared disk quorum
    device for you.
    This screen allows you to disable the automatic selection and
    configuration of a quorum device.

    You have chosen to turn on the global fencing. If your shared
    storage devices do not support SCSI, such as Serial Advanced
    Technology Attachment (SATA) disks, or if your shared disks do not
    support SCSI-2, you must disable this feature.

    If you disable automatic quorum device selection now, or if you
    intend to use a quorum device that is not a shared disk, you must
    instead use clsetup(1M) to manually configure quorum once both
    nodes have joined the cluster for the first time.

    Do you want to disable automatic quorum device selection (yes/no)
    [no]?  no
```


Specify `no` when asked whether you want to disable the automatic quorum device selection.

Configuring by Using All-at-Once and Typical Modes: Example (10/12)

```
...
Is it okay to create the new cluster (yes/no) [yes]?  yes

During the cluster creation process, cluster check is run on each
of the new cluster nodes. If cluster check detects problems, you
can either interrupt the process or check the log files after the
cluster has been established.

Interrupt cluster creation for cluster check errors (yes/no) [no]?
no
```


Specify `yes` when asked whether it is okay to create the new cluster.

Specify `no` when asked to interrupt cluster creation for cluster check errors.

The cluster configuration begins. Somewhere toward the end of the cluster configuration, the cluster node reboots.

Configuring by Using All-at-Once and Typical Modes: Example (11/12)

```
Cluster Creation
    Log file - /var/cluster/logs/install/scinstall.log.5197

    Configuring global device using lofi on clnode2: done

    Starting discovery of the cluster transport configuration.

    The following connections were discovered:

       clnode1:net1  switch1  clnode2:net1
         clnode1:net3  switch2  clnode2:net3

    Completed discovery of the cluster transport configuration.
     Started cluster check on "clnode1".
     Started cluster check on "clnode2".

     cluster check failed for "clnode1".
     cluster check failed for "clnode2".

The cluster check command failed on both of the nodes.
```

Configuring by Using All-at-Once and Typical Modes: Example (12/12)

```
    .........
Refer to the log file for details.
The name of the log file is
/var/cluster/logs/install/scinstall.log.5197.


   Configuring "clnode2" ... done
    Rebooting "clnode2" ... done

    Configuring "clnode1" ... done
    Rebooting "clnode1" ...

Log file - /var/cluster/logs/install/scinstall.log.5197


Rebooting ...

Connection to clnode1 closed by remote host.
Connection to clnode1 closed.
oracle@adminws:~$
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (1/24)

```
clnode1:/# /usr/cluster/bin/scinstall
*** Main Menu ***

    Please select from one of the following (*) options:

      * 1) Create a new cluster or add a cluster node
        2) Configure a cluster to be JumpStarted from this install
           server
        3) Manage a dual-partition upgrade
        4) Upgrade this cluster node
        5) Print release information for this cluster node

      * ?) Help with menu options
      * q) Quit

    Option: 1
```


The following is an example of using the one-at-a-time node configuration. The dialog is shown for `clnode1`, the first node in the cluster. You cannot install other cluster nodes until this node is rebooted into the cluster and can then be the sponsor node for the other nodes.

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (2/24)

```
*** New Cluster and Cluster Node Menu ***

    Please select from any one of the following options:

        1) Create a new cluster
        2) Create just the first node of a new cluster on this machine
        3) Add this machine as a node in an existing cluster

        ?) Help with menu options
        q) Return to the Main Menu

    Option: 2
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (3/24)

First node introduction

```
*** Establish Just the First Node of a New Cluster ***
This option is used to establish a new cluster using this
machine as the first node in that cluster.

Before you select this option, the Oracle Solaris Cluster
framework software must already be installed. Use the Oracle
Solaris Cluster installer to install Oracle Solaris Cluster
software.

Press Control-d at any time to return to the Main Menu.

Do you want to continue (yes/no) [yes]? yes
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (4/24)

Typical or Custom Mode

```
>>> Typical or Custom Mode <<<

This tool supports two modes of operation, Typical mode and
Custom.

For most clusters, you can use Typical mode. However, you
might need to select the Custom mode option if not all of the
Typical defaults can be applied to your cluster.
...
    Please select from one of the following options:
        1) Typical
        2) Custom
        ?) Help
        q) Return to the Main Menu
    Option [1]:2
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (5/24)

Cluster name

```
>>> Cluster Name <<<

Each cluster has a name assigned to it. The name can be made up of any
characters other than whitespace.

Each cluster name should be unique within the namespace of your
enterprise.

What is the name of the cluster you want to establish? cluster1
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (6/24)

Option for cluster check

```
>>> Check <<<

This step enables you to run cluster check to verify that certain basic
hardware and software pre-configuration requirements have been met.

If cluster check detects potential problems with configuring this
machine as a cluster node, a report of violated checks is prepared and
available for display on the screen.

Do you want to run cluster check (yes/no) [yes]? No
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (7/24)

Authenticating nodes with DES

```
>>> Authenticating Requests to Add Nodes <<<

After the first node establishes itself as a single node cluster, other
nodes attempting to add themselves to the cluster configuration must be
found on the list of nodes you just provided. You can modify this list
by using claccess(1CL) or other tools after the cluster has been
established.

By default, nodes are not securely authenticated as they attempt to add
themselves to the cluster configuration. This is generally considered
adequate, because nodes which are not physically connected to the
private cluster interconnect will never be able to actually join the
cluster. However, DES authentication is available. If DES
authentication is selected, you must configure all necessary encryption
keys before any node will be allowed to join the cluster (see
keyserv(1M), publickey(4)).

Do you need to use DES authentication (yes/no) [no]? no
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (8/24)

Choosing whether to define switches and switch names

```
>>> Point-to-Point Cables <<<
The two nodes of a two-node cluster may use a directly-connected
interconnect. That is, no cluster switches are configured. However,
when there are greater than two nodes, this interactive form of
scinstall assumes that there will be exactly one switch for each
private network.

Does this two-node cluster use switches (yes/no) [yes]? yes
>>> Cluster Switches <<<

All cluster transport adapters in this cluster must be cabled to a
"switch." And, each adapter on a given node must be cabled to a
different switch. Interactive scinstall requires that you identify one
switch for each private network in the cluster.

What is the name of the first switch in the cluster [switch1]? <CR>

What is the name of the second switch in the cluster [switch2]? <CR>
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (9/24)

```
>>> Cluster Transport Adapters and Cables <<<

You must configure the cluster transport adapters for each node in the
cluster. These are  the adapters which attach to the private cluster
interconnect.
Select the first cluster transport adapter:
      1) net1
      2) net2
      3) net3
      4) Other
      Option: 2
Will this be a dedicated cluster transport adapter (yes/no) [yes]?  yes
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (10/24)

```
Adapter “net2” is an Ethernet adapter.
Searching for any unexpected network traffic on “net1” ... Done

Verification completed. No traffic was detected over a 10 second sample
period.
The “dlpi” transport type will be set for this cluster.
Name of the switch to which “net1” is connected [switch1]? <CR>

Each adapter is cabled to a particular port on a switch. And, each port
is assigned a name. You can explicitly assign a name to each port. Or,
for Ethernet and Infiniband switches, you can choose to allow scinstall
to assign a default name for you. The default port name assignment sets
the name to the node number of the node hosting the transport adapter
at the other end of the cable.
Use the default port name for the “net2” connection (yes/no) [yes]? yes
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (11/24)

```
Select the second cluster transport adapter:
    1) net1
    2) net2
    3) net3
    4) Other
    Option: 3
Will this be a dedicated cluster transport adapter (yes/no) [yes]? Yes

Adapter “net3” is an Ethernet adapter.
Searching for any unexpected network traffic on “net3” ... Done

Verification completed. No traffic was detected over a 10 second sample
period.
Name of the switch to which “net3” is connected [switch2]? <CR>
Use the default port name for the “net3” connection (yes/no) [yes]? yes
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (12/24)

```
>>> Network Address for the Cluster Transport <<<

The cluster transport uses a default network address of 172.16.0.0. If
this IP address is already in use elsewhere within your enterprise,
specify another address from the range of recommended private addresses
(see RFC 1918 for details).

The default netmask is 255.255.240.0. You can select another netmask,
as long as it minimally masks all bits that are given in the network
address.
The default private netmask and network address result in an IP address
range that supports a cluster with a maximum of 64 nodes, 10 private
networks and 0 virtual clusters.

Is it okay to accept the default network address (yes/no) [yes]?yes

Is it okay to accept the default netmask (yes/no) [yes]? no
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (13/24)

```
The combination of private netmask and network address will dictate
the maximum number of both nodes and private networks, and virtual
clusters that can be supported by a cluster. Given your private
network address, this program will generate a range of recommended
private netmasks that are based on the maximum number of nodes and
private networks and the virtual clusters that you anticipate for
this cluster.
In specifying the anticipated maximum number of nodes and private
networks and virtual clusters for this cluster, it is important
that you give serious consideration to future growth potential.
While both the private netmask and network address can be changed
later, the tools for making such changes require that all nodes in
the cluster be booted in noncluster mode.

Maximum number of nodes anticipated for future growth [64]?  4
Maximum number of private networks anticipated for future growth
[10]?  4
Maximum number of virtual clusters expected [12]?  2
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (14/24)

```
Specify a netmask of 255.255.255.192 to meet anticipated future
requirements of 4 cluster nodes and 4 private networks.

To accommodate more growth, specify a netmask of 255.255.255.0 to
support up to 8 cluster nodes and 8 private networks.

What netmask do you want to use [255.255.255.192]? 255.255.255.192

Plumbing network address 172.16.0.0 on adapter ce2>> NOT DUPLICATE ...
done

Plumbing network address 172.16.0.0 on adapter ce3>> NOT DUPLICATE ...
done
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (15/24)

```
>>> Global Devices File System <<<

Each node in the cluster must have a local file system mounted on
global/.devices/node@<nodeID> before it can successfully participate as
a cluster member. Because the "nodeID" is not assigned until
scinstall is run, scinstall will set this up for you.
You must supply the name of either an already-mounted file system or
raw disk partition which scinstall can use to create the global devices
file system. This file system or partition should be at least 512 MB in
size.
Alternatively, you can use a loopback file (lofi), with a new file
system, and mount it on /global/.devices/node@<nodeid>.
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (16/24)

```
If an already-mounted file system is used, the file system must be
empty. If a raw disk partition is used, a new file system will be
created for you.

If the lofi method is used, scinstall creates a new 100 MB filesystem
from a lofi device by using the file /.globaldevices. The lofi method
is typically preferred, since it does not require the allocation of a
dedicated disk slice.

The default is to use /globaldevices.

Is it okay to use this default (yes/no) [yes]? no
Is it okay to use the lofi method (yes/no) [yes]? yes
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (17/24)

```
>>> Set Global Fencing <<<

Fencing is a mechanism that a cluster uses to protect data
integrity when the cluster interconnect between nodes is lost. By
default, fencing is turned on for global fencing, and each disk
uses the global fencing setting. This screen enables you to turn
off the global fencing.

...

If you choose to turn off global fencing now, after your cluster
starts you can still use the cluster(1CL) command to turn on global
fencing.

Do you want to turn off global fencing (yes/no) [no]? no
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (18/24)

Automatic quorum configuration (two-node cluster)

```
>>> Quorum Configuration <<<
Every two-node cluster requires at least one quorum device. By default,
scinstall selects and configures a shared disk quorum device for you.

This screen enables you to disable the automatic selection and
configuration of a quorum device.
You have chosen to turn on the global fencing. If your shared storage
devices do not support SCSI, such as Serial Advanced Technology
Attachment (SATA) disks, or if your shared disks do not support SCSI-2,
you must disable this feature.

If you disable automatic quorum device selection now, or if you intend
to use a quorum device that is not a shared disk, you must instead use
clsetup(1M) to manually configure quorum after both nodes have joined
the cluster for the first time.

Do you want to disable automatic quorum device selection (yes/no) [no]?
no
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (19/24)

Automatic reboot

```
>>> Automatic Reboot <<<

After scinstall has successfully initialized the Oracle Solaris Cluster
software for this machine, the machine must be rebooted. After the
reboot, this machine is established as the first node in the new
cluster.

Do you want scinstall to reboot for you (yes/no) [yes]? Yes
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (20/24)

Option confirmation

```
>>> Confirmation <<<
Your responses indicate the following options to scinstall:
Your responses indicate the following options to scinstall:
   scinstall -i \
     -C cluster1 \
     -F \
     -G lofi \
     -T node=clnode1,node=clnode2,authtype=sys \
     -w netaddr=172.16.0.0,netmask=255.255.255.192,maxnodes=4,
        maxprivatenets=4,numvirtualclusters=2 \
     -A trtype=dlpi,name=net2 -A trtype=dlpi,name=net3 \
     -B type=switch,name=switch1 -B type=switch,name=switch2 \
     -m endpoint=:net2,endpoint=switch1 \
     -m endpoint=:net3,endpoint=switch2 \
     -P task=quorum,state=INIT
```

ORACLE

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (21/24)

Option confirmation

```
                                    ...
The explanation for these options can be found in the Oracle Solaris
Cluster Installation Guide. The first option -i is config. without
pkgadd (the packages have already been installed at this point). -C is
the cluster name. -F is the first node in the cluster. -G is the option
for global devices. -T is the authentication option. -w are the private
network options. -A displays the adapter options. -B shows the switch
options. -m specifies the transport cable connections. -P is the post-
configuration option (for selecting quorum).

Are these the options you want to use (yes/no) [yes]? yes

Do you want to continue with the this configuration step (yes/no)
 [yes]? Yes
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (22/24)

Configuration messages

```
                                    ...
Initializing cluster name to “cluster1" ... done
Initializing authentication options ... done
Initializing configuration for adapter “net2" ... done
Initializing configuration for adapter “net3" ... done
Initializing configuration for switch "switch1" ... done
Initializing configuration for switch "switch2" ... done
Initializing configuration for cable ... done
Initializing configuration for cable ... done
Initializing private network address options ... done
```

ORACLE

Configuring Using One-at-a-Time and Custom Modes: Example (First Node) (23/24)

Configuration messages

```
Setting the node ID for "clnode1" ... done (id=1)

Verifying that NTP is configured ... done
Initializing NTP configuration ... done

Adding cluster node entries to /etc/inet/hosts ... done

Configuring IP multipathing groups ...done
```

Configuring by Using One-at-a-Time and Custom Modes: Example (First Node) (24/24)

Configuration messages

```
Verifying that power management is NOT configured ... done
Unconfiguring power management ... done
/etc/power.conf has been renamed to /etc/power.conf.041309025345
Power management is incompatible with the HA goals of the cluster.

Please do not attempt to re-configure power management.
Ensure that the EEPROM parameter “local-mac-address?” is set to “true”
... done
Ensure network routing is disabled ... done
Log file - /var/cluster/logs/install/scinstall.log.1065

Rebooting ...
```

Quiz

Which of the following statements are true about configuring an entire cluster at once?

a. The node that you are driving from becomes the first to join the cluster.
b. Drive from the node that you want to have the highest node ID.
c. List other nodes in reverse order.

ORACLE

Answer: b, c

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (1/13)

```
Clnode2 :/# scinstall

*** Main Menu ***
Please select from one of the following (*) options:
    * 1) Create a new cluster or add a cluster node
      2) Configure a cluster to be JumpStarted from this install server
      3) Manage a dual-partition upgrade
      4) Upgrade this cluster node
      5) Print release information for this cluster node
    * ?) Help with menu options
    * q) Quit
    Option: 1
```


In the one-at-a-time method, after the first node has rebooted into the cluster, you can configure the remaining node or nodes. Here, there is almost no difference between the Typical and Custom modes, except that the Typical mode does not ask about the global devices file system. (The installer assumes that the placeholder is `/globaldevices`.) Here, you have no choice about the automatic quorum selection or the authentication mechanism, because it was already chosen on the first node.

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (2/13)

```
*** New Cluster and Cluster Node Menu ***

Please select from any one of the following options:

    1) Create a new cluster
    2) Create just the first node of a new cluster on this machine
    3) Add this machine as a node in an existing cluster

    ?) Help with menu options
    q) Return to the Main Menu

    Option: 3
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (3/13)

Additional node configuration

```
*** Add a Node to an Existing Cluster ***

This option is used to add this machine as a node in an already
established cluster. If this is a new cluster, there may only be a
single node which has established itself in the new cluster.

Before you select this option, the Oracle Solaris Cluster framework
software must already be installed.

Press Control-d at any time to return to the Main Menu.

Do you want to continue (yes/no) [yes]? yes
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (4/13)

Typical or Custom mode

```
>>> Typical or Custom Mode <<<

This tool supports two modes of operation, Typical and Custom Modes.
For most clusters, you can use Typical mode. However, you might need to
select the Custom mode option if not all of the Typical defaults can be
applied to your cluster.
For more information about the differences between Typical and Custom
modes, select the Help option from the menu.
Please select from one of the following options:
        1) Typical
        2) Custom
        ?) Help
        q) Return to the Main Menu
        Option [1]: 1
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (5/13)

Sponsoring node

```
>>> Sponsoring Node <<<

For any machine to join a cluster, it must identify a node in that
cluster willing to "sponsor" its membership in the cluster. When
configuring a new cluster, this "sponsor" node is typically the first
node used to build the new cluster. However, if the cluster is already
established, the "sponsoring" node can be any node in that cluster.

Already established clusters can keep a list of hosts which are able to
configure themselves as new cluster members. This machine should be in
the join list of any cluster which it tries to join. If the list does
not include this machine, you may need to add it by using claccess(1CL)
or other tools.

And, if the target cluster uses DES to authenticate new machines
attempting to configure themselves as new cluster members, the
necessary encryption keys must be configured before any attempt to
join.

What is the name of the sponsoring node? clnode1
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (6/13)

Cluster name

```
>>> Cluster Name <<<

Each cluster has a name assigned to it. When adding a node to the
cluster, you must identify the name of the cluster you are attempting
to join. A sanity check is performed to verify that the “sponsoring“
node is a member of that cluster.

What is the name of the cluster you want to join? cluster1

Attempting to contact “clnode1” ... done

Cluster name “cluster1” is correct.

Press Enter to continue:
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (7/13)

Option for cluster check

```
>>> Check <<<

This step enables you to run cluster check to verify that certain basic
hardware and software pre-configuration requirements have been met. If
cluster check detects potential problems with configuring this machine
as a cluster node, a report of violated checks is prepared and
available for display on the screen.

Do you want to run cluster check (yes/no) [yes]? no
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (8/13)

Cluster transport autodiscovery option

```
>>> Autodiscovery of Cluster Transport <<<

If you are using Ethernet or Infiniband adapters as the cluster
transport adapters, autodiscovery is the best method for configuring
the cluster transport.

Do you want to use autodiscovery (yes/no) [yes]? yes
    Probing ........

The following connections were discovered:
    clnode1:net2  switch1  clnode2:net2
    clnode1:net3  switch2  clnode2:net3

Is it okay to add these connections to the configuration (yes/no)
[yes]? yes
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (9/13)

Letting you use `lofi` device for `/global/.devices/node@#`

```
/globaldevices is not a directory or file system mount point.
Cannot use "/globaldevices."

Do you want to use a lofi device instead and continue the installation
(yes/no) [yes]? Yes
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (10/13)

Automatic reboot

```
>>> Automatic Reboot <<<

After scinstall has successfully initialized the Oracle Solaris Cluster
software for this machine, the machine must be rebooted. The reboot
will cause this machine to join the cluster for the first time.

Do you want scinstall to reboot for you (yes/no) [yes]? yes
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (11/13)

Option confirmation

```
>>> Confirmation <<<

Your responses indicate the following options to scinstall:

   scinstall -i \
      -C cluster1 \
      -N clnode1 \
      -G lofi \
      -A trtype=dlpi,name=net2 -A trtype=dlpi,name=net3 \
      -m endpoint=:net2,endpoint=switch1 \
      -m endpoint=:net3,endpoint=switch2

Are these the options you want to use (yes/no) [yes]? yes

Do you want to continue with this configuration step (yes/no) [yes]?
yes
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (12/13)

Configuration messages

```
Adding node "clnode2" to the cluster configuration ... done
Adding adapter "net2" to the cluster configuration ... done
Adding adapter "net3" to the cluster configuration ... done
Adding cable to the cluster configuration ... done
Adding cable to the cluster configuration ... done

Copying the config from "clnode1" ... done

Copying the postconfig file from "clnode1" if it exists ... done

Setting the node ID for "clnode2" ... done (id=2)

Verifying the major number for the "did" driver with "clnode1" ... done

Verifying that NTP is configured ... done
Initializing NTP configuration ... Done
```

Configuring Additional Nodes When Using the One-at-a-Time Method: Example (13/13)

Configuration messages

```
...
Adding cluster node entries to /etc/inet/hosts ... done

Configuring IP multipathing groups ...done

Verifying that power management is NOT configured ... done
Unconfiguring power management ... done
/etc/power.conf has been renamed to /etc/power.conf.041309030943
Power management is incompatible with the HA goals of the cluster.
Please do not attempt to re-configure power management.

Ensure that the EEPROM parameter "local-mac-address?" is set to "true"
... done
Ensure network routing is disabled ... done

Updating file ("ntp.conf.cluster") on node clnode1 ... done
Updating file ("hosts") on node clnode1 ... done

Log file - /var/cluster/logs/install/scinstall.log.1091

Rebooting ...
```

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the root environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Settings Automatically Configured by `scinstall`

Oracle Solaris OS files and settings that are automatically configured by `scinstall` are:

- Changes to the `/etc/inet/hosts` file
- Modifying the `/etc/nsswitch.conf` file
- Changes to the `/etc/inet/ntp.conf.sc` file
- Modifying the `local-mac-address?` EEPROM variable

`scinstall` automatically configures the following files and settings on each cluster node:

- `/etc/inet/hosts file`
- `/etc/nsswitch.conf file`
- `/etc/inet/ntp.conf file`
- `local-mac-address?`
  Setting in electrically erasable programmable read-only memory (EEPROM) (SPARC only)

Settings Automatically Configured by `scinstall`

- Changes to the `/etc/inet/hosts` file
  - Adds other host names to each host
  - The names must have been resolvable.
- Modifying the `/etc/nsswitch.conf` file
  - Puts `files` first for every line
  - Puts `cluster` keyword for `hosts`, `ipnodes`, and `netmasks`
- Changes to the `/etc/inet/ntp.conf.sc` file
  - `peer clusternode1-priv prefer`
  - `peer clusternode2-priv`
- Modifying the `local-mac-address?` EEPROM variable

The `scinstall` utility automatically adds all the cluster names and IP addresses to each node's hosts file if it was not there already. (All the names already had to be resolvable, through some name service, for `scinstall` to work at all.)

Changes to the `/etc/nsswitch.conf` File

- It makes sure the files keyword precedes every other name service for every entry in the file.
- It adds the `cluster` keyword for the `hosts` and `netmasks` keywords. This keyword modifies the standard Oracle Solaris OS resolution libraries so that they can resolve the cluster transport host names and netmasks directly from the CCR. The default transport host names (associated with IP addresses on the `clprivnet0` adapter) are `clusternode1-priv`, `clusternode2-priv`, and so on. These names can be used by any utility or application as normal resolvable names without having to be entered in any other name service.

Creating the `/etc/inet/ntp.conf.sc` File

This file contains a Network Time Protocol Configuration which, if used, has all nodes synchronize their time clocks against each other (and only against each other). This file is automatically modified to contain only lines for cluster nodes defined during `scinstall`, and, therefore, should not need to be modified manually. For a two-node cluster, it includes the following lines:

```
peer clusternode1-priv prefer
peer clusternode2-priv
```

Modifying the `local-mac-address?` EEPROM Variable

On SPARC systems, this EEPROM variable is set to `true` so that each network adapter is given a unique Media Access Control (MAC) address (that is, Ethernet address for Ethernet adapters) to support IPMP.

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the root environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- **Performing automatic quorum configuration**
- Describing manual quorum selection
- Performing post-installation verification

Automatic Quorum Configuration and `installmode` Resetting

```
Apr 13 03:17:39 theo cl_runtime: NOTICE: CMM: Cluster members:
clnode1,clnode2.
Apr 13 03:17:39 theo cl_runtime: NOTICE: CMM: node reconfiguration
#4 completed.
Apr 13 03:17:39 theo cl_runtime: NOTICE: CMM: Votecount changed
from 0 to 1 for node clnode2.
Apr 13 03:17:39 theo cl_runtime: NOTICE: CMM: Cluster members:
clnode1,clnode2.
Apr 13 03:17:39 theo cl_runtime: NOTICE: CMM: node reconfiguration
#5 complete
```


On a two-node cluster on which you chose to allow automatic quorum configuration, the quorum device is chosen (the lowest possible DID device number) as the second node boots into the cluster for the first time.

If your cluster has more than two nodes, no quorum device is selected automatically, but the `installmode` flag is automatically reset as the last node boots into the cluster.

In the Oracle Solaris 11 OS, as the last node boots into the cluster, you get the login prompt on the last node booting into the cluster before the quorum auto-configuration runs. This is because the boot environment is controlled by the SMF of the Oracle Solaris 11 OS, which runs boot services in parallel and gives you the login prompt before many of the services are complete. The auto-configuration of the quorum device does not complete until a minute or so later. Do not attempt to configure the quorum device by hand, because the auto-configuration eventually runs to completion.

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the root environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Manual Quorum Selection

```
# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  clnode1:/dev/rdsk/c0t0d0
d2                  clnode1:/dev/rdsk/c0t1d0
d3                  clnode1:/dev/rdsk/c0t6d0
d4                  clnode1:/dev/rdsk/c1t0d0
d4                  clnode2:/dev/rdsk/c1t0d0
…
d18                 clnode1:/dev/rdsk/c2t10d0
d18                 clnode2:/dev/rdsk/c2t10d0
d19                 clnode1:/dev/rdsk/c2t11d0
d19                 clnode2:/dev/rdsk/c2t11d0
d20                 clnode2:/dev/rdsk/c0t0d0
d21                 clnode2:/dev/rdsk/c0t1d0
d22                 clnode2:/dev/rdsk/c0t6d0
```


You must choose a quorum device or quorum devices manually in the following circumstances:

- Two-node cluster where you disabled automatic quorum selection
- Any cluster of more than two nodes where a quorum device is desired

Verifying DID Devices

If you are going to be manually choosing quorum devices that are physically attached disks or LUNs, you must know the DID device number for the quorum device or devices that you want to choose.

The `cldevice` (`cldev`) command shows the DID numbers assigned to the disks in the cluster. The most succinct option that shows the mapping between DID numbers and all the corresponding disk paths is `cldev list -v`.

You must know the DID device number for the quorum device that you choose in the next step. You can choose any multiported disk.

Note: The local disks (single-ported) appear at the beginning and end of the output and cannot be chosen as quorum devices.

```
# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  clnode1:/dev/rdsk/c0t0d0
d2                  clnode1:/dev/rdsk/c0t1d0
d3                  clnode1:/dev/rdsk/c0t6d0
d4                  clnode1:/dev/rdsk/c1t0d0
d4                  clnode2:/dev/rdsk/c1t0d0
d5                  clnode1:/dev/rdsk/c1t1d0
d5                  clnode2:/dev/rdsk/c1t1d0
d6                  clnode1:/dev/rdsk/c1t2d0
d6                  clnode2:/dev/rdsk/c1t2d0
d7                  clnode1:/dev/rdsk/c1t3d0
d7                  clnode2:/dev/rdsk/c1t3d0
d8                  clnode1:/dev/rdsk/c1t8d0
d8                  clnode2:/dev/rdsk/c1t8d0
d9                  clnode1:/dev/rdsk/c1t9d0
d9                  clnode2:/dev/rdsk/c1t9d0
d10                 clnode1:/dev/rdsk/c1t10d0
d10                 clnode2:/dev/rdsk/c1t10d0
d11                 clnode1:/dev/rdsk/c1t11d0
d11                 clnode2:/dev/rdsk/c1t11d0
d12                 clnode1:/dev/rdsk/c2t0d0
d12                 clnode2:/dev/rdsk/c2t0d0
d13                 clnode1:/dev/rdsk/c2t1d0
d13                 clnode2:/dev/rdsk/c2t1d0
d14                 clnode1:/dev/rdsk/c2t2d0
d14                 clnode2:/dev/rdsk/c2t2d0
d15                 clnode1:/dev/rdsk/c2t3d0
d15                 clnode2:/dev/rdsk/c2t3d0
d16                 clnode1:/dev/rdsk/c2t8d0
d16                 clnode2:/dev/rdsk/c2t8d0
d17                 clnode1:/dev/rdsk/c2t9d0
d17                 clnode2:/dev/rdsk/c2t9d0
d18                 clnode1:/dev/rdsk/c2t10d0
d18                 clnode2:/dev/rdsk/c2t10d0
d19                 clnode1:/dev/rdsk/c2t11d0
d19                 clnode2:/dev/rdsk/c2t11d0
d20                 clnode2:/dev/rdsk/c0t0d0
d21                 clnode2:/dev/rdsk/c0t1d0
d22                 clnode2:/dev/rdsk/c0t6d0
```

Manual Quorum Selection

Choosing a quorum device (clusters with more than two nodes)

```
# /usr/cluster/bin/clsetup

*** Main Menu ***
Please select from one of the following options:

    1) Quorum
    2) Resource groups
    3) Data Services
    4) Cluster interconnect
    5) Device groups and volumes
    6) Private hostnames
    7) New nodes
    8) Other cluster properties
    ?) Help with menu options
    q) Quit
    Option: 1
```


In a cluster of more than two nodes, the `installmode` flag is always automatically reset, but the quorum device or devices are never automatically selected.

You should use `clsetup` to choose quorum devices, but the initial screens look a little different because the `installmode` flag is already reset.

Manual Quorum Selection

Choosing a quorum device (clusters with more than two nodes)

```
*** Quorum Menu ***

Please select from one of the following options:

    1) Add a quorum device
    2) Remove a quorum device

    ?) Help
    q) Return to the Main Menu

    Option: 1
```


From here, the dialog looks similar to the previous example, except that the `installmode` is already reset. Therefore, after adding your quorum devices, you just return to the main menu.

Manual Quorum Selection

Choosing quorum and resetting the `installmode` attribute (two-node cluster)

```
# /usr/cluster/bin/clsetup
>>> Initial Cluster Setup <<<

This program has detected that the cluster "installmode" attribute is
still enabled. As such, certain initial cluster setup steps will be
performed at this time. This includes adding any necessary quorum
devices, then resetting both the quorum vote counts and the
"installmode" property.

Please do not proceed if any additional nodes have yet to join the
cluster.

Is it okay to continue (yes/no) [yes]?  yes

Do you want to add any quorum devices (yes/no) [yes]? yes
```

ORACLE

Before a new cluster can operate normally, you must reset the `installmode` attribute on all nodes. On a two-node cluster where automatic quorum selection was disabled, the `installmode` will still be set on the cluster. You must choose a quorum device as a prerequisite to resetting `installmode`.

Choosing quorum by using the `clsetup` utility: The `clsetup` utility is a menu-driven interface, which, when the `installmode` flag is reset, turns into a general menu-driven alternative to low-level cluster commands.

The `clsetup` utility recognizes whether the `installmode` flag is still set, and will not present any of its normal menus until you reset it. For a two-node cluster, this involves choosing a single quorum device first.

Manual Quorum Selection

Choosing quorum by using the `clsetup` utility

```
Following are supported Quorum Devices types in Oracle Solaris Cluster.
Please refer to Oracle Solaris Cluster documentation for detailed
information on these supported quorum device topologies.

What is the type of device you want to use?

    1) Directly attached shared disk
    2) Network Attached Storage (NAS) from Network Appliance
    3) Quorum Server
    q)

    Option: 1
```

Manual Quorum Selection

Choosing quorum by using the `clsetup` utility

```
>>> Add a SCSI Quorum Disk <<<

A SCSI quorum device is considered to be any Oracle Solaris Cluster-
supported attached storage which is connected to two or more nodes of
the cluster. Dual-ported SCSI-2 disks may be used as quorum devices in
two-node clusters. However, clusters with more than two nodes require
that SCSI-3 PGR disks be used for all disks with more than two node-to-
disk paths.

You can use a disk containing user data or one that is a member of a
device group as a quorum device.

For more information on supported quorum device topologies, see the
Oracle Solaris Cluster documentation.

Is it okay to continue (yes/no) [yes]? Yes
Which global device do you want to use (d<N>)?  D4
Is it okay to proceed with the update (yes/no) [yes]?  yes
```

ORACLE

Manual Quorum Selection

Choosing quorum by using the `clsetup` utility

```
clquorum add d4

Command completed successfully.


Press Enter to continue:

Do you want to add another quorum device (yes/no) [yes]?  no

After the "installmode" property has been reset, this program will skip
"Initial Cluster Setup" each time it is run again in the future.
However, quorum devices can always be added to the cluster by using the
regular menu options. Resetting this property fully activates quorum
settings and is necessary for the normal and safe operation of the
cluster.
```

Agenda

- Identifying cluster install package groups
- Prerequisites for installing the Oracle Solaris Cluster software
- Installing the Oracle Solaris Cluster software
- Setting the root environment
- Configuring the Oracle Solaris Cluster software
- Sample cluster configuration scenarios
- Settings that are automatically configured by `scinstall`
- Performing automatic quorum configuration
- Describing manual quorum selection
- Performing post-installation verification

Performing Post-Installation Verification

- Cluster status
- Cluster configuration information
- `status` subcommands:
  - Nodes
  - Devices
  - Quorum votes
  - Device groups
  - Resource groups and resources
  - Interconnect

When you have completed the Oracle Solaris Cluster software installation on all nodes, verify the following information:

- General cluster status
- Cluster configuration information

Verifying General Cluster Status

The status subcommand of the cluster utilities shows the current status of various cluster components, such as:

- Nodes
- Devices
- Quorum votes (including node and device quorum votes)
- Device groups
- Resource groups and related resources
- Cluster interconnect status

Note: The cluster command-line interface (CLI) commands are described in detail starting in the next lesson and continuing on a per-topic basis as you configure storage and applications into the cluster in the following lessons.

Performing Post-Installation Verification

```
root@clnode1:~# cluster status -t quorum
root@clnode1:~# clquorum status

=== Cluster Quorum ===
--- Quorum Votes Summary from (latest node reconfiguration) ---
            Needed   Present   Possible
            ------   -------   --------
            2        3         3

--- Quorum Votes by Node (current status) ---
Node Name        Present       Possible       Status
---------        -------       --------       ------
clnode2          1             1              Online
clnode1          1             1              Online

--- Quorum Votes by Device (current status) ---
Device Name        Present      Possible      Status
-----------        -------      --------      ------
d1                 1            1             Online
```

ORACLE

The following two commands give identical output, and show the cluster membership and quorum vote information:

```
# cluster status -t quuorum
# clquorum status
```

Performing Post-Installation Verification

```
root@clnode1:~# cluster status -t interconnect
root@clnode1:~# clinterconnect status

=== Cluster Transport Paths ===

Endpoint1                Endpoint2                Status
---------                ---------                ------
clnode2:net3             clnode1:net3             Path online
clnode2:net1             clnode1:net1             Path online
```


The following two commands are identical, and show the status of the private networks that make up the cluster interconnect (cluster transport):

```
# cluster status -t interconnect
# clinterconnect status
```

Performing Post-Installation Verification

```
root@clnode1:~# cluster show

=== Cluster ===

Cluster Name:                                         cluster1
  clusterid:                                            0x4F066DD6
  installmode:                                          disabled
  heartbeat_timeout:                                    10000
  heartbeat_quantum:                                    1000
  private_netaddr:                                      172.16.0.0
  private_netmask:                                      255.255.240.0
  max_nodes:                                            64
  max_privatenets:                                      10
  num_zoneclusters:                                     12
  udp_session_timeout:                                  480
  concentrate_load:                                     False
  global_fencing:                                       prefer3
  Node List:                                            clnode2, clnode1
```

ORACLE

Cluster configuration is displayed in general by using the `list`, `list -v`, `show`, and `show -v` subcommands of the various cluster utilities.

The following command shows the configuration of everything. If you added a `-t` global at the end of the command, it would list only the cluster global properties that appear in the first section of output.

```
# cluster show
```

Performing Post-Installation Verification

```
=== Host Access Control ===

  Cluster name:                                       cluster1
    Allowed hosts:                                       None
    Authentication Protocol:                             sys

  === Cluster Nodes ===

  Node Name:                                          clnode2
    Node ID:                                             1
    Enabled:                                             yes
    privatehostname:                                     clusternode1-priv
    reboot_on_path_failure:                              disabled
    globalzoneshares:                                    1
    defaultpsetmin:                                      1
    quorum_vote:                                         1
    quorum_defaultvote:                                  1
    quorum_resv_key:                                     0x4F066DD600000001
    Transport Adapter List:                              net1, net3
```

Performing Post-Installation Verification

```
Node Name:                                    clnode1
   Node ID:                                       2
   Enabled:                                       yes
   privatehostname:                               clusternode2-priv
   reboot_on_path_failure:                        disabled
   globalzoneshares:                              1
   defaultpsetmin:                                1
   quorum_vote:                                   1
   quorum_defaultvote:                            1
   quorum_resv_key:                               0x4F066DD600000002
   Transport Adapter List:                        net1, net3
```

ORACLE

Performing Post-Installation Verification

```
=== Transport Cables ===
 Transport Cable:                              clnode2:net1,switch1@1
   Endpoint1:                                     clnode2:net1
   Endpoint2:                                     switch1@1
   State:                                         Enabled

 Transport Cable:                              clnode2:net3,switch2@1
   Endpoint1:                                     clnode2:net3
   Endpoint2:                                     switch2@1
   State:                                         Enabled

 Transport Cable:                              clnode1:net1,switch1@2
   Endpoint1:                                     clnode1:net1
   Endpoint2:                                     switch1@2
   State:                                         Enabled

 Transport Cable:                              clnode1:net3,switch2@2
   Endpoint1:                                     clnode1:net3
   Endpoint2:                                     switch2@2
   State:                                         Enabled
```

Performing Post-Installation Verification

```
=== Transport Switches ===

  Transport Switch:                              switch1
    State:                                          Enabled
    Type:                                           switch
    Port Names:                                     1 2
    Port State(1):                                  Enabled
    Port State(2):                                  Enabled

  Transport Switch:                              switch2
    State:                                          Enabled
    Type:                                           switch
    Port Names:                                     1 2
    Port State(1):                                  Enabled
    Port State(2):                                  Enabled
```

Performing Post-Installation Verification

```
=== Quorum Devices ===

 Quorum Device Name:                              d1
   Enabled:                                          yes
   Votes:                                            1
   Global Name:                                      /dev/did/rdsk/d1s2
   Type:                                             shared_disk
   Access Mode:                                      scsi3
   Hosts (enabled):                                  clnode2, clnode1

 === Device Groups ===
.
.
.
```

Summary

In this lesson, you should have learned how to:

- Identify cluster install package groups
- Describe the prerequisites for installing the Oracle Solaris Cluster software
- Install the Oracle Solaris Cluster software
- Set the root environment
- Configure the Oracle Solaris Cluster software
- Describe cluster configuration scenarios
- Describe settings that are automatically configured by `scinstall`
- Perform automatic quorum configuration
- Describe manual quorum selection
- Perform post-installation verification

Practice 5 Overview: Installing and Configuring the Oracle Solaris Cluster Software

This practice covers the following topics:

- Task 1: Establishing Connection to the Cluster Nodes
- Task 2: Preparing the Cluster Node Environment
- Task 3: Installing the Oracle Solaris Cluster Packages
- Task 4: Configuring a New Cluster – All Nodes at Once
- Task 5: Configuring a New Cluster – One Node at a Time
- Task 6: Verifying the Cluster Configuration and Status

6

Performing Cluster Administration

Objectives

After completing this lesson, you should be able to:

- Identify the cluster daemons
- Use cluster commands
- Use RBAC with Oracle Solaris Cluster
- Administer cluster global properties
- Administer cluster nodes
- Administer quorum
- Administer disk path monitoring
- Administer SCSI protocol settings for storage devices
- Administer interconnect components
- Use the `clsetup` command
- Perform cluster operations
- Modify private network settings while in non-cluster mode

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Identifying Cluster Daemons

```
# ps -ef | grep clust
 root     8     0    0 03:36:21 ?           0:09 cluster
    root   935     1    0 03:38:48 ?           0:00
/usr/cluster/lib/sc/ifconfig_proxy_serverd
    root   911   910    0 03:38:47 ?           0:00 /usr/cluster/lib/sc/clexecd
    root   624     1    0 03:37:40 ?           0:00 /usr/cluster/lib/sc/qd_userd
    root   894     1    0 03:38:46 ?           0:00 /usr/cluster/lib/sc/cl_execd
    root   940     1    0 03:38:48 ?           0:00 /usr/cluster/lib/sc/rtreg_proxy_serverd
    root  1236     1    0 03:39:01 ?           0:00 /usr/cluster/lib/sc/cl_eventlogd
    root   895   894    0 03:38:46 ?           0:00 /usr/cluster/lib/sc/cl_execd
    root   910     1    0 03:38:47 ?           0:00 /usr/cluster/lib/sc/clexecd
    root   889     1    0 03:38:46 ?           0:00 /usr/cluster/lib/sc/clevent_listenerd
    root   901     1    0 03:38:47 ?           0:00 /usr/cluster/lib/sc/failfastd
    root   916     1    0 03:38:47 ?           0:00 /usr/cluster/lib/sc/pmmd
    root  1120     1    0 03:38:56 ?           0:00 /usr/cluster/lib/sc/scqdmd
    root  1129     1    0 03:38:56 ?           0:00 /usr/cluster/lib/sc/cl_eventd
    root  1171     1    0 03:38:58 ?           0:00 /usr/cluster/lib/sc/rpc.fed
    root  1212     1    0 03:39:00 ?           0:00 /usr/cluster/lib/sc/pnm_mod_serverd
    root  1124     1    0 03:38:56 ?           0:00 /usr/cluster/lib/sc/sc_zonesd
    root  1141     1    0 03:38:57 ?           0:00 /usr/cluster/lib/sc/rpc.pmfd
    root  1223     1    0 03:39:00 ?           0:00 /usr/cluster/lib/sc/rgmd
    root  1163     1    0 03:38:57 ?           0:00 /usr/cluster/lib/sc/cl_ccrad
    root  1165     1    0 03:38:57 ?           0:00 /usr/cluster/lib/sc/scdpmd
    root  1194     1    0 03:38:59 ?           0:00 /usr/cluster/lib/sc/scprivipd
    root  1179     1    0 03:38:58 ?           0:00 /usr/cluster/bin/cl_pnmd
    root  1183     1    0 03:38:58 ?           0:00 /usr/cluster/lib/sc/cznetd
    root  1225   916    0 03:39:00 ?           0:00 /usr/cluster/lib/sc/rgmd -z global
    root  1280     1    0 03:39:02 ?           0:00 /usr/cluster/lib/sc/syncsa_serverd
```


When a cluster node is fully booted into a cluster, several cluster daemons are added to the traditional Oracle Solaris Operating System (OS).

None of these daemons require any manual maintenance, regardless of which version of Oracle Solaris OS you are running. Behind the scenes, the Oracle Solaris 11 OS uses Service Management Facility (SMF) to launch daemons. Therefore, at boot time, you might see a console login prompt before many of these daemons are launched. SMF itself can restart some daemons.

The output shown in the slide is taken from a cluster node running Oracle Solaris 11 OS.

- `cluster`: This is a system process (created by the kernel) to encapsulate the kernel threads that make up the core kernel range of operations.

  There is no way to kill this process (even with a `KILL` signal) because it is always in the kernel.
- `failfastd`: This daemon is the `failfast` proxy server. Other daemons that require services of the `failfast` device driver register with `failfastd`. The `failfastd` daemon allows the kernel to panic if certain essential daemons have failed.
- `clexecd`: This is used by cluster kernel threads to execute user commands (such as the `run_reserve` and `dofsck` commands). It is also used to run cluster commands remotely (like the `cluster shutdown` command).

  This daemon registers with `failfastd` so that the `failfast` device driver will panic the kernel if this daemon is killed and not restarted in 30 seconds.
- `cl_eventd`: This daemon registers and forwards cluster events (such as nodes entering and leaving the cluster). There is also a protocol whereby user applications can register themselves to receive cluster events.

  The daemon is automatically respawned if it is killed.
- `qd_userd`: This daemon serves as a proxy whenever any quorum device activity requires execution of some command in userland.

  If you kill this daemon, you must restart it manually.
- `scqdmd`: This daemon monitors the health of the quorum device.

  The daemon is automatically re-spawned if it is killed.
- `rgmd`: This is the resource group manager, which manages the state of all cluster-unaware applications.

  The `failfast` driver panics the kernel if this daemon is killed and not restarted in 30 seconds.
- `rpc.fed`: This is the fork-and-exec daemon, which handles requests from `rgmd` to spawn methods for specific data services.

  The `failfast` driver panics the kernel if this daemon is killed and not restarted in 30 seconds.
- `rpc.pmfd`: This is the process monitoring facility. It is used as a general mechanism to initiate restarts and failure action scripts for some cluster framework daemons (in Solaris 9 OS), and for most application daemons and application fault monitors.

  The `failfast` driver panics the kernel if this daemon is stopped and not restarted in 30 seconds.
- `cl_pnmd`: This is the public network management daemon, which manages network status information received from the local IPMP daemon running on each node and facilitates application failovers caused by complete public network failures on nodes.

  It is automatically restarted if it is stopped.

- cl_eventlogd: This daemon logs cluster events into a binary log file.

  It is automatically restarted if it is stopped.
- cl_ccrad: This daemon provides access from user management applications to the CCR.

  It is automatically restarted if it is stopped.
- scdpmd: This daemon monitors the status of disk paths, so that they can be reported in the output of the cldev status command.

  It is automatically restarted if it is stopped.
- sc_zonesd: This daemon monitors the state of Oracle Solaris 10 HA zones so that applications designed to failover between zones can react appropriately to zone booting and failure.

  The failfast driver panics the kernel if this daemon is stopped and not restarted in 30 seconds.
- scprivipd: This daemon provisions IP addresses on the clprivnet0 interface, on behalf of zones.

  It is automatically restarted if it is stopped.
- pnm_mod_serverd, ifconfig_proxy_serverd: These daemons run in the global zones to provide required network services to zones in an Oracle Solaris zones cluster.

  They are automatically restarted if stopped.
- rtreg_proxy_serverd: This daemon runs in the global zone to provide resource type registration services to zones of an Oracle Solaris zones cluster.

  It is automatically restarted if stopped.
- pmmd: This daemon provides cross-cluster starting, restarting, and monitoring for some essential cluster processes such as rgmd that need to be aware of process status on different nodes.

  The failfast driver panics the kernel if this daemon is stopped and not restarted in 30 seconds.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Using Cluster Commands

- The command's name relates to the object that is administered.
- Every command requires a subcommand (action).
- Every command has a common syntax style:
  - The object in question is the last item on the command line.
  - You can use + (wildcard) in place of the object name.
  - For display (list, show, status):
    - + is the default.
    - You can use the name to limit output to only objects of that name.

The Oracle Solaris Cluster command-line commands have an object-oriented nature:

- The name of the command is related to the cluster object you are trying to display or administer.
- Every command requires a subcommand that indicates what it is you actually want to do.
- Every command has a common style of syntax in which:
  - When using subcommands to operate on (or delete) a specific object, you give the name of the object as the last item on the command line, or you give a + to indicate a wildcard. The wildcard might still be limited by other command-line arguments (specific group or subtypes, if appropriate).
  - When using subcommands to display configuration or status, the default is to show all objects (of that category), but you can give an optional last argument to show only a specific object

Commands for Basic Cluster Administration

- `clnode`
- `clquorum` (`clq`)
- `clinterconnect` (`clintr`)
- `cldevice` (`cldev`)
- `clnasdevice` (`clnas`)
- `cluster`

The following commands relate to configuration, status, or administration of a cluster. Every command has a full name. Some commands have an abbreviated name (long and short names are hard linked to the same executable):

- `clnode`: Configuration, status, and settings for nodes
- `clquorum` (`clq`): Configuration, status, settings, adding, and deleting quorum devices (includes node quorum information)
- `clinterconnect` (`clintr`): Configuration, status, settings, adding, and deleting private networks
- `cldevice` (`cldev`): Configuration, status, and settings for individual devices (disk, CD-ROM, and tape)
- `cluster`:
  - Administering cluster global settings
  - Showing configuration and status of everything; can be limited by other arguments to certain types or groups of information

Additional Cluster Commands

- `cldevicegroup` (`cldg`)
- `clresourcegroup` (`clrg`)
- `clresource` (`clrs`)
- `clreslogicalhostname` (`clrslh`)
- `clressharedaddress` (`clrssa`)
- `clresourcetype` (`clrt`)

The following commands relate to administration of device groups and cluster application resources. In subsequent lessons, you learn more about these commands.

- `cldevicegroup` (`cldg`): Device group configuration, status, settings, and adding and deleting device groups (including VxVM and Solaris Volume Manager device groups)
- `clresourcegroup` (`clrg`): Application resource group configuration, status, settings, and adding and deleting application resource groups
- `clresource` (`clrs`): Resource configuration, status, settings, and adding and deleting individual resources in application resource groups
- `clreslogicalhostname` (`clrslh`) and `clressharedaddress` (`clrssa`): IP resource configuration, status, settings, and adding and deleting IP resources in application resource groups (These commands simplify tasks that can also be accomplished with `clresource`.)
- `clresourcetype` (`clrt`): Resources for Oracle Solaris Cluster data services

Cluster Command Self-Documentation (1/2)

```
# clquorum
clquorum:  (C961689) Not enough arguments.
clquorum:  (C101856) Usage error.

Usage:    clquorum <subcommand> [<options>] [+ | <devicename> ...]
          clquorum [<subcommand>]  -? | --help
          clquorum  -V | --version

Manage cluster quorum

SUBCOMMANDS:

add            Add a quorum device to the cluster configuration
disable        Put quorum devices into maintenance state
enable         Take quorum devices out of maintenance state
export         Export quorum configuration
list           List quorum devices
remove         Remove a quorum device from the cluster configuration
reset          Reset the quorum configuration
show           Show quorum devices and their properties
status         Display the status of the cluster quorum
```


While all the cluster commands have excellent man pages, they are also self-documenting because, if you run a command without any subcommand, the usage message always lists the possible subcommands.

Cluster Command Self-Documentation (2/2)

```
# clquorum add
clquorum:  (C456543) You must specify the name of the quorum device to add.
clquorum:  (C101856) Usage error.

Usage:    clquorum add [<options>] <devicename>
          clquorum add -a [-v]
          clquorum add -i <configfile> [<options>] + | <devicename> ...

Add a quorum device to the cluster configuration

OPTIONS:

  -?    Display this help text.

  -a    Automatically add a Shared Disk quorum device for 2-node cluster.

  -i {- | <clconfiguration>}
        Specify XML configuration as input.

  -p <name>=<value>
        Specify the properties.

  -t <type>
        Specify the device type.

  -v    Verbose output.
```


If you run a subcommand that requires arguments, the usage message gives you more information about the particular subcommand. It does not give you all the information about the names of properties that you might need to set (for that, you have to go to the `man` pages).

Quiz

You want to manage *quorum devices* (configuration, status, settings, and adding and deleting). Which cluster command would you use?

a. `clnode`

b. `clq`

c. `clintr`

d. `cldev`

e. `cluster`

Answer: b

Quiz

You want to manage *private networks* (configuration, status, settings, and adding and deleting). Which cluster command would you use?

a. `clnode`
b. `clq`
c. `clintr`
d. `cldev`
e. `cluster`

Answer: c

Quiz

You want to manage cluster global settings. Which cluster command would you use?

a. `clnode`

b. `clq`

c. `clintr`

d. `cldev`

e. `cluster`

Answer: e

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Oracle Solaris Cluster RBAC Profiles

There are three authorization levels. You can assign these levels directly to a user or role, or through a profile:

- `solaris.cluster.read`: Authorization for list, show, and other read operations
- `solaris.cluster.admin`: Authorization to change the state of a cluster object
- `solaris.cluster.modify`: Authorization to change properties of a cluster object

For selected Oracle Solaris Cluster commands and options that you issue at the command line, use Role-Based Access Control (RBAC) for authorization. Oracle Solaris Cluster has a simplified RBAC structure that can enable you to assign cluster administrative privileges to non-root users or roles. Oracle Solaris Cluster commands and options that require RBAC authorization will require one or more of the following authorization levels:

- `solaris.cluster.read`
- `solaris.cluster.admin`
- `solaris.cluster.modify`

Oracle Solaris Cluster RBAC rights profiles apply to both voting nodes in a global cluster.

`solaris.cluster.read`: Gives the ability to do any status, list, or show subcommands. By default, every user has this authorization because it is in the Basic Oracle Solaris User profile. This can be assigned directly to a user.

`solaris.cluster.modify`: Gives the ability to run add, create, delete, remove, and related subcommands. Can be assigned directly to a role (allowed users must assume the role, and then they get the authorization).

`solaris.cluster.admin`: Gives the ability to run switch, online, offline, enable, and disable subcommands. Can be assigned to a rights profile that is then given to a user or a role.

Creating an RBAC Role

Using the command line, create an RBAC role by performing the following steps:

1. Assume a `solaris.cluster.admin` RBAC authorization role.
2. Create a role:
   – By using the `roleadd` command
   – By editing the `user_attr` file to add a user with `type=role`
   – By using the `roleadd` and `rolemod` commands
3. Start and stop the name service cache daemon.

To create and assign an RBAC role with an Oracle Solaris Cluster Management Rights Profile, perform the following steps:

1. Become a superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization.
2. Select one of the following methods for creating a role:
   - For roles in the local scope, do either of the following:
     – Use the `roleadd` command to specify a new local role and its attributes.
     – Edit the `user_attr` file to add a user with `type=role`. Note that you should use this method only for emergencies.
   - For roles in a name service, use the `roleadd` and `rolemod` commands to specify the new role and its attributes. The `roleadd` and `rolemod` commands require authentication by superuser or a role that is capable of creating other roles. Note that you can apply the `roleadd` command to all name services.
3. Start and stop the name service cache daemon. New roles do not take effect until the name service cache daemon is restarted. As root, type the following text:

```
# /etc/init.d/nscd stop
# /etc/init.d/nscd start
```

Modifying a Non-Root User's RBAC Properties

Using either the user accounts tool or the command line, modify a user's RBAC properties by performing the following steps:

1. Assume a `solaris.cluster.modify` RBAC authorization role.
2. Modify RBAC properties with one of the following commands, as appropriate:
   – Using the `usermod` command
   – Editing the `user_attr` file
   – Using the `roleadd` and `rolemod` commands
3. Start and stop the name service cache daemon.

To modify a non-root user's RBAC properties, perform the following steps:

1. Become a superuser or assume a role that provides `solaris.cluster.modify` RBAC authorization.
2. Select one of the following methods for modifying the RBAC properties:
   - To change user properties that are assigned to a user who is defined in the local scope or in an LDAP repository, you can use the `usermod` command.
   - To change the authorizations, roles, or rights profiles that are assigned to a user who is defined in the local scope, edit the `user_attr` file.
     Note: Use this method for emergencies only.
   - To manage roles locally or in a name service such as an LDAP repository, use the `roleadd` or `rolemod` commands. These commands require authentication as superuser or as a role that is capable of changing user files. You can apply these commands to all name services.
     Note: The Forced Privilege and Stop Rights profiles that ship with Oracle Solaris 11 cannot be modified.
3. Start and stop the name service cache daemon.

Quiz

Select the three authorizations that pertain to Oracle Solaris Cluster:

a. `solaris.cluster.create`

b. `solaris.cluster.read`

c. `solaris.cluster.modify`

d. `solaris.cluster.add`

e. `solaris.cluster.admin`

Answer: b, c, e

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Viewing and Administering Cluster Global Properties

```
# cluster show -t global

=== Cluster ===

Cluster Name:                                   cluster1
  clusterid:                                       0x4EFCF337
  installmode:                                     disabled
  heartbeat_timeout:                               10000
  heartbeat_quantum:                               1000
  private_netaddr:                                 192.16.0.0
  private_netmask:                                 255.255.255.0
  max_nodes:                                       6
  max_privatenets:                                 10
  num_zoneclusters:                                12
  udp_session_timeout:                             480
  concentrate_load:                                False
  global_fencing:                                  prefer3
  Node List:                                       clnode1, clnode2
```


In the example shown in the slide, where the name of the cluster is `cluster1`, the `cluster show -t global` command shows only the cluster global properties or global default SCSI protocol settings.

Renaming the Cluster

```
# cluster rename -c blackcat
# cluster show -t global

Cluster ===

Cluster Name:                                   blackcat
  clusterid:                                       0x4EFCF337
  installmode:                                     disabled
  heartbeat_timeout:                               10000
  heartbeat_quantum:                               1000
  private_netaddr:                                 172.16.0.0
.
.
# cluster rename -c cluster1
```

ORACLE

You can rename the cluster by using the cluster rename command. The cluster name is not particularly important and is not required as an argument in any other commands.

Setting Other Cluster Properties

- `heartbeat_quantum` (in milliseconds)
- `heartbeat_timeout` (in milliseconds)
  - `heartbeat_timeout` must be at least five times bigger than `heartbeat_quantum`.

```
# cluster set -p heartbeat_timeout=4000
cluster:  heartbeat timeout 4000 out of range 5000 - 60000.
# cluster set -p heartbeat_quantum=800
# cluster set -p heartbeat_timeout=4000
# cluster show -t global
Cluster ===
Cluster Name:                                   cluster1
  clusterid:                                       0x4EFCF337
  installmode:                                     disabled
  heartbeat_timeout:                               4000
  heartbeat_quantum:                               800
  private_netaddr:                                 172.16.0.0
  private_netmask:                                 255.255.255.0
  max_nodes:                                       6
  max_privatenets:                                 10
```


`heartbeat_quantum` controls the timing of cluster heartbeats on the private network (in milliseconds).

`heartbeat_timeout` describes the number of milliseconds of missing heartbeat required by a node to declare a single interconnect dead or to declare the other node(s) dead and start reconfiguration.

You usually do not change these values, although it is possible to make them bigger (if your nodes are very far apart) or smaller (if for some reason you are unsatisfied with the 10-second timeout).

The cluster enforces that `heartbeat_timeout` is at least five times as big as `heartbeat_quantum`, as illustrated in the slide.

Note: Modifying the `private_netaddr` and `private_netmask` properties is a special case in that it is done only when the entire cluster is down and all nodes are booted in non-cluster mode. This is covered later in the lesson.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

ORACLE

Viewing and Administering Nodes

Use the `clnode` command to view node status and configuration.

```
# clnode status
Cluster Nodes ===

--- Node Status ---

Node Name                                           Status
---------                                           ------
clnode1                                             Online
clnode2                                             Online
```


The `clnode` command can be used to show status and configuration of nodes. Although it shows a variety of data for each node in the cluster, there is only a limited amount of information that you can actually change with the `clnode` command.

Viewing node status and configuration: The `status` and `show` subcommands show, by default, all nodes. You can also show a single node by giving its name as the last command-line argument.

Modifying Node Information

- Most fields cannot be modified.
- `privatehostname` can be modified.

```
# clnode set -p privatehostname=clnode1-priv clnode1
# clnode show clnode1
=== Cluster Nodes ===

Node Name:                                      clnode1
  Node ID:                                        1
  Enabled:                                        yes
  privatehostname:                                clusternode1-priv
  reboot_on_path_failure:                         disabled
  globalzoneshares:                               1
  defaultpsetmin:                                 1
  quorum_vote:                                    1
  quorum_defaultvote:                             1
  quorum_resv_key:                                0x4EFCF33700000001
  Transport Adapter List:                         net1, net3
…
# getent hosts clnode1
192.168.1.102   clnode2
# clnode set -p privatehostname=clnode1-priv clnode1
```


Most of the information shown by `clnode` cannot be modified by the `clnode` command. Some can be modified by other commands (`clinterconnect` for adding and deleting transport adapters, for example).

The `reboot_on_path_failure` command is described later in the lesson. You can run the `clnode` command instead of the `clsetup` utility to change the `privatehostname`.

You can set the `privatehostname` to whatever you want. This name automatically resolves to the IP address associated with the node's `clprivnet0` adapter. This is the single private IP address whose traffic is automatically distributed across all physical private networks.

Note: If it seems that the OS can resolve private hostnames that no longer exist (because you changed them), it is because of the OS name-caching daemon (`nscd`). You can use `nscd -i hosts` to clear this cache.

Viewing Software Release Information on a Node

```
# clnode show-rev -v
Oracle Solaris Cluster 4.0.0 0.22.1 for Solaris 11 i386
ha-cluster/data-service/apache                                   :4.0.0-0.22
ha-cluster/data-service/dhcp                                     :4.0.0-0.22
ha-cluster/data-service/dns                                      :4.0.0-0.22
...
# cat /etc/cluster/release
     Oracle Solaris Cluster 4.0.0 0.22.1 for Solaris 11 i386
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights
                              reserved.
```


You can use `clnode show-rev -v` to see installed cluster package release information on a node or on all nodes. This is useful information to have when talking to technical support personnel.

You can also directly examine the `/etc/cluster/release` file to get quick information about the release of the cluster software installed on a particular node.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Viewing and Administering Quorum (1/2)

```
# clq status

=== Cluster Quorum ===

--- Quorum Votes Summary from (latest node reconfiguration) ---

            Needed   Present   Possible
            ------   -------   --------
            2        3         3

--- Quorum Votes by Node (current status) ---

Node Name       Present       Possible       Status
---------       -------       --------       ------
clnode1         1             1              Online
clnode2         1             1              Online


--- Quorum Votes by Device (current status) ---

Device Name       Present      Possible      Status
-----------       -------      --------      ------
d2                1            1             Online
```


The `clquorum` (`clq`) command is used to view the configuration and status of quorum devices and node vote count information and to add and delete quorum devices.

Viewing quorum status and configuration: The `status`, `list`, and `show` suboptions show the status and configuration of quorum devices and node-related quorum information. You can reduce the amount of information by adding a type-restriction option (`-t` *`shared_disk`*, for example), or by adding the name of a particular quorum device or node as the very last argument.

Viewing and Administering Quorum (2/2)

```
# clq show d2

=== Quorum Devices ===

Quorum Device Name:                              d2
  Enabled:                                          yes
  Votes:                                            1
  Global Name:                                      /dev/did/rdsk/d2s2
  Type:                                             shared_disk
  Access Mode:                                      scsi3
  Hosts (enabled):                                  clnode1, clnode2
```


To view the quorum configuration, use `clquorum show` or `clq show`.

Adding and Removing (and Replacing) Quorum Devices

```
# clq add d5
# clq remove d2
# clq status -t shared_disk

Cluster Quorum ===

--- Quorum Votes by Device ---

Device Name        Present      Possible      Status
-----------        -------      --------      ------
d5                  1             1             Online
```


There is no specific command to replace or repair a quorum device. You just add a new one and remove the old one.

A two-node cluster, which absolutely requires a quorum device, requires that you perform repairs in that order (add and then remove). If you have more than two nodes, you can perform the operations in any order.

Installing a Quorum Server (Outside the Cluster)

- Outside the cluster, install the `ha-cluster/service/quorum-server` package.
- The quorum server instance is configured automatically on port 9000.
- You can serve as many clusters as you like.

When you install the `ha-cluster/service/quorum-server` package on a machine outside the cluster, a quorum server instance is configured automatically on the default quorum server port (9000). This single instance can serve as the quorum device for as many clusters as you like.

Adding a Quorum Server Device to a Cluster (1/2)

```
# clq add -t quorum_server -p qshost=clustergw -p port=9000 d2
# clq remove d5

# clq status
Cluster Quorum ===
--- Quorum Votes Summary ---
            Needed   Present   Possible
            ------   -------   --------
            2        3         3
--- Quorum Votes by Node ---
Node Name       Present       Possible       Status
---------       -------       --------       ------
clnode1          1             1              Online
clnode2          1             1              Online
--- Quorum Votes by Device ---
Device Name     Present       Possible       Status
-----------     -------       --------       ------
d2               1             1              Online
```


On the cluster side, you need to specifically add a quorum server device to serve the cluster. This is likely to be your only quorum device (after you remove other quorum devices, if necessary) because it always has the number of votes equal to one fewer than the number of node votes.

On the cluster side, you assign a random ASCII device name to the quorum server device (in this example, `qservydude`).

```
# clq add -t quorum_server -p qshost=clustergw \
-p port=9000 qservydude

# clq remove d5
```

Adding a Quorum Server Device to a Cluster (2/2)

```
# clq show d2

=== Quorum Devices ===

Quorum Device Name:                              d2
  Enabled:                                          yes
  Votes:                                            1
  Global Name:                                      /dev/did/rdsk/d2s2
  Type:                                             shared_disk
  Access Mode:                                      scsi3
  Hosts (enabled):                                  clnode1, clnode2
```

Quiz

Statement 1: For a two-node cluster, if you need to change the quorum, you must first add a new quorum and then remove the existing one.

Statement 2: For clusters that have more than two nodes, you can first delete an existing quorum and then add a new quorum.

Which one of the following is correct about statements 1 and 2?

a. 1 is true and 2 is false.
b. Both 1 and 2 are true.
c. 1 is false and 2 is true.

Answer: b

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Viewing and Administering Disk Paths and Settings

The `cldevice` (`cldev`) command performs the following:

- Displays disk path information and the mapping between DID device names and `c#t#d#`
- Displays path status information (Paths are monitored by `scdpmd`.)
- Enables you to change disk monitoring settings
- Enables you to change properties affecting choice of SCSI-2 or SCSI-3 for disks with exactly two paths

The `cldevice` (`cldev`) command can:

- Display disk path information (which nodes are connected to which disks) and the mapping between DID device names and `c#t#d#`
- Display disk path status information (Disk paths are monitored by a `scdpmd` daemon.)
- Enable you to change disk monitoring settings:
  - By default, all paths are monitored.
  - You see at least one reason you may like to turn off monitoring for non-shared disks.
- Enable you to change fencing properties

Displaying Disk Paths

```
# cldev list -v
DID Device          Full Device Path
----------          ----------------
d1                  clnode1:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E2520002d0
d1                  clnode2:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E2520002d0
d2                  clnode1:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E25F0003d0
d2                  clnode2:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E25F0003d0
d3                  clnode1:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E28C0004d0
d3                  clnode2:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E28C0004d0
d4                  clnode1:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E2950005d0
d4                  clnode2:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E2950005d0
.
. //Omitted for brevity
.
d9                  clnode1:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E776000Ad0
d9                  clnode2:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E776000Ad0
d10                 clnode1:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E7E1000Bd0
d10                 clnode2:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E7E1000Bd0
d11                 clnode1:/dev/rdsk/c3t0d0
d12                 clnode1:/dev/rdsk/c3t1d0
d13                 clnode2:/dev/rdsk/c3t0d0
d14                 clnode2:/dev/rdsk/c3t1d0
```


The `cldev list -v` command provides the best summary of node-to-disk paths and the corresponding DID device numbers.

Displaying Disk Path Status

Paths are probed periodically.

```
# cldev status
Device Instance              Node               Status
---------------              ----               ------
/dev/did/rdsk/d1             clnode1            Ok
                             clnode2            Ok

/dev/did/rdsk/d10            clnode1            Ok
                             clnode2            Ok

/dev/did/rdsk/d11            clnode1            Ok

/dev/did/rdsk/d13            clnode2            Ok

/dev/did/rdsk/d2             clnode1            Ok
                             clnode2            Ok

...
# cldev status -s Fail

Device Instance               Node              Status
---------------               ----              ------
```


By default, the `scdpmd` daemon probes disk paths periodically (once every few minutes).

Disk path status changes are logged into `/var/adm/messages` with the `syslogd` `LOG_INFO` facility level. All failures are logged by using the `LOG_ERR` facility level.

The `cldevice` status command shows the status of disk paths as last recorded by the daemon. That is, you can pull out a disk or sever a disk path and the status might still be reported as `Ok` for a couple of minutes, until the next time the daemon probes the paths.

`# cldev status -s fail` is used to print faulted disk paths for the entire cluster.

Changing Disk Path Monitoring Settings

```
# cldev monitor all
# cldev unmonitor d8
# cldev unmonitor -n clnode2 d9
# cldev status -s Unmonitored

=== Cluster DID Devices ===

Device Instance               Node              Status
---------------               ----              ------
/dev/did/rdsk/d8              clnode1           Unmonitored
                              clnode2           Unmonitored
```


By default, all paths to all disks are monitored.

You can unmonitor specific paths, or re-monitor them if you have previously unmonitored them. By default, you affect the path from all connected nodes to the DID device that you mention. You can limit your action to a path from a specific node by using the `-n` option. For example, the commands shown in the slide unmonitor all paths to disk device `d8`, and specifically the path from node `clnode2` to disk device `d9`.

As is the convention for all cluster commands, you can use `+` as the wildcard. So `cldev monitor +` turns monitoring back on for all devices, regardless of their previous state.

Unmonitoring All Non-Shared Devices and Enabling `reboot_on_path_failure`

- The `reboot_on_path_failure` property will make your node reboot if:
  - All monitored paths fail
  - Some disks (at least one) have a path to another node
- If you want to use it, you should unmonitor local disks.

```
# cldev unmonitor d1 d2 d4
# clnode set -p reboot_on_path_failure=enabled clnode1
# cldev unmonitor d20 d21
# clnode set -p reboot_on_path_failure=enabled clnode2
```


If you set the `reboot_on_path_failure` property to `enabled` (the default is `disabled`), a node automatically reboots if all the monitored disk paths from that node are broken and there is another node that still has working paths to at least one of the shared devices.

This is a way to ensure HA even when there are multiple simultaneous disk path failures. Typically, with a single-path failure (to an entire array, for example), everything would still be fine because your data would always be mirrored across multiple controllers, or multipathed, or both. But if a particular node loses access to all disk paths, you might decide (by setting the property) that it is best to just have the node reboot, so that at least any clustered applications running there can fail over to other nodes.

It does not make sense to enable `reboot_on_path_failure` if you are still monitoring non-shared (local) disks. The whole point is to reboot if you lose contact with all multiported storage, but the actual feature will not reboot a node unless it detects failure on all monitored storage. Therefore, if you want to enable this feature, you must unmonitor any local disks.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Viewing Settings Related to SCSI-2 and SCSI-3 Disk Reservations

- Global setting:

```
# cluster show | grep global_fencing
global_fencing:                                    prefer3
```

- Individual disk setting (default is to follow global setting):

```
# cldev show d7
DID Device Name:                                 /dev/did/rdsk/d7
  Full Device Path:
    clnode2:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E6300008d0
  Full Device Path:
    clnode1:/dev/rdsk/c0t600144F0B3CAAC8300004EF9E6300008d0
  Replication:                                      none
  default_fencing:                                  global
```


A global setting controls what form of SCSI reservations, if any, are used with disks. The default value, as demonstrated in the command example, is `prefer3`. With this value, disks with exactly two paths are fenced with SCSI-2 reservations (add SCSI-2 PGRE when used as a quorum device). Disks with more than two paths are fenced with SCSI-3 reservations, which already implement the persistence needed for quorum devices. Nothing more is needed.

Each individual disk has its own fencing property. The default value for every disk is `global`, which means that fencing for that disk follows the `global` property.

Modifying Properties to Use SCSI-3 Reservations for Disks with Two Paths

```
# clq show d2

=== Quorum Devices ===

Quorum Device Name:                              d2
  Enabled:                                          yes
  Votes:                                            1
  Global Name:                                      /dev/did/rdsk/d2s2
  Type:                                             shared_disk
  Access Mode:                                      scsi3
  Hosts (enabled):                                  clnode1, clnode2
```

ORACLE

The per-disk policy for the existing quorum device has been set to `pathcount` so that it does not follow the global default of `prefer3`. For other disks, the individual `default_fencing` property remains with the value `global` and the cluster immediately uses SCSI-3 the next time they need to be fenced.

Getting Quorum Device to Use SCSI-3 Policy (1/2)

1. Remove the disk as a quorum device (but you may have to add another quorum device first if this is a two-node cluster).
2. Change the per-disk `default_fencing` policy back to the value `global`.
3. Add the disk back as a quorum device (at which point you might want to remove any other “placeholder” quorum device).

Getting Quorum Device to Use SCSI-3 Policy (2/2)

```
# clq add d5
# clq remove d4
# cldev set -p default_fencing=global d4
Updating shared devices on node 1
Updating shared devices on node 2
# clq add d4
# clq remove d5
# clq show d4
Quorum Devices ===
Quorum Device Name:                    d4
  Enabled:                             yes
  Votes:                               1
  Global Name:                         /dev/did/rdsk/d4s2
  Type:                                shared_disk
  Access Mode:                         scsi3
  Hosts (enabled):                     clnode1, clnode2
```


The example shown in the slide demonstrates the procedure.

Eliminating SCSI Fencing for Particular Disk Devices

Set per-disk fencing (`default_fencing`):

- `nofencing`
- `nofencing-noscrub`

```
# cldev set -p default_fencing=nofencing d5
Updating shared devices on node 1
Updating shared devices on node 2
```


You can turn off SCSI fencing for particular disk devices. This feature enables the use of Serial Advanced Technology Attachment (SATA) devices as shared storage devices. These devices do not support SCSI fencing of any sort.

Note: Elimination of fencing would also enable you to attach devices that support SCSI-2 but do not support SCSI-3 directly to more than two nodes. This, however, is not the intention of the feature. It was specifically invented to support SATA devices.

It is not recommended in any way that you eliminate fencing for devices that support fencing.

The per-disk `default_fencing` property has values that specify that you do not want fencing for that disk:

- `nofencing`: Turns off fencing after scrubbing the disk of any existing reservation keys
- `nofencing-noscrub`: Turns off fencing without any scrubbing

The example in the slide shows the elimination of fencing for a particular disk device.

Eliminating SCSI Fencing Globally

```
# cluster set -p global_fencing=nofencing
Warning: Device instance d4 is a quorum device - fencing protocol remains
    SCSI-3 for the device.
Updating shared devices on node 1
Updating shared devices on node 2
# clq add d5
# clq remove d4
# cldev show d4
DID Device Instances ===
DID Device Name: /dev/did/rdsk/d4
Full Device Path: clnode1:/dev/rdsk/c1t0d0
Full Device Path: clnode2:/dev/rdsk/c1t0d0
Replication: none
default_fencing: scsi3
# cldev set -p default_fencing=global d4
Updating shared devices on node 1
Updating shared devices on node 2
# clq add d4
# clq remove d5
```


It is likely that if none of your shared disk devices support fencing (all are SATA drives, for example), you would have to turn off fencing globally during `scinstall` time.

In the unlikely case that you need to turn off fencing on all devices after Oracle Solaris Cluster is already running, you can use the new values of the cluster-wide `global_fencing` property. The values are the same as those for the per-disk fencing property, listed previously.

The example in the slide shows the elimination of all disk fencing globally. As in the earlier examples of switching from `scsi2` to `scsi3` fencing, a disk that is already a quorum device requires special manipulation.

Software Quorum for Disks with No SCSI Fencing

```
# clq show d4
=== Quorum Devices ===

Quorum Device Name:                               d4
  Enabled:                                           yes
  Votes:                                             1
  Global Name:                                       /dev/did/rdsk/d6s2
  Type:                                              shared_disk
  Access Mode:                                       sq_disk
  Hosts (enabled):                                   clnode1, clnode2
```

ORACLE

As shown in the previous section, you are enabled to add a disk on which SCSI fencing has been eliminated as a quorum device.

Oracle Solaris Cluster will quietly implement its own software quorum mechanism to reliably and atomically simulate the SCSI-2 or SCSI-3 “race” for the quorum device. The persistent reservations will be implemented by using PGRE, exactly as when a SCSI-2 device is used as a quorum device.

The example in the slide shows verification that a quorum device is using the software quorum mechanism, because its SCSI fencing has been eliminated.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Viewing and Administering Interconnect Components

```
# clintr status

=== Cluster Transport Paths ===

Endpoint1               Endpoint2               Status
---------               ---------               ------
clnode1:net3            clnode2:net3            Path online
clnode1:net1            clnode2:net1            Path online
```


The `clinterconnect` (`clintr`) command enables you to display the configuration and status of the private networks that make up the cluster transport. In addition, it enables you to configure new private networks and/or remove private network components without having to reboot any cluster nodes.

Viewing interconnect status: The `clintr` status command shows the status of all private network paths between all pairs of nodes.

No particular software administration is required to repair a broken interconnect path. If a cable breaks, for example, the cluster immediately reports a path failure. If you replace the cable, the path immediately goes back online.

Adding New Private Networks

- If defined without a switch (two-node only):
  - Define the two adapter endpoints
  - Define the cable between the endpoints
- If defined with a switch:
  - Define the adapter endpoints
  - Define the switch endpoints (any name without a node)
  - Define the cables between each adapter and the new switch

You can cable a new private network and get it defined in the cluster without any reboots or interruption to any existing service.

The private network definitions in the cluster configuration repository are somewhat complex. You must perform one of the following sets of actions:

- For a private network defined without a switch (two-node cluster only):
  - Define the two adapter endpoints (Oracle Solaris 11 vanity naming feature names all interfaces as `net<N>`).
  - Define the cable between the two endpoints.
- For a private network defined with a switch:
  - Define the adapter endpoints for each node.
  - Define the switch endpoint (cluster assumes any endpoint not in the form of *`node:adapter`* is a switch).
  - Define cables between each adapter and the switch.

Adding New Private Networks (Two-Node Cluster with Switch)

```
# clintr add clnode1:net4
# clintr add clnode2:net4
# clintr add switch3
# clintr add clnode1:net4,switch3
# clintr add clnode2:net4,switch3

# clintr status

- - Cluster Transport Paths - -

                    Endpoint            Endpoint            Status
                    --------            --------            ------
Transport path: clnode1:net3        clnode2:net3        Path online
Transport path: clnode1:net2        clnode2:net2        Path online
Transport path: clnode1:net4        clnode2:net4        Path online
```


For example, the commands shown in the slide define a new private network for a two-node cluster. The definitions define a switch. This does not mean that the switch needs to physically exist. It is just a definition in the cluster.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Using the `clsetup` Command

```
# clsetup

  *** Main Menu ***

    Please select from one of the following options:

        1) Quorum
        2) Resource groups
        3) Data Services
        4) Cluster interconnect
        5) Device groups and volumes
        6) Private hostnames
        7) New nodes
        8) Other cluster tasks

        ?) Help with menu options
        q) Quit
```

ORACLE

The `clsetup` command is a menu-driven utility meant to guide you through many common (but not all) administrative operations. The `clsetup` command leads you through a series of menus, and at the end calls the lower-level administrative commands for you.

In general, `clsetup` always shows the lower-level commands as it runs them, so it has educational value as well.

Comparing Low-level Command and `clsetup` Usage

```
*** Cluster Interconnect Menu ***
Please select from one of the following options:
1) Add a transport cable
2) Add a transport adapter to a node
3) Add a transport switch
4) Remove a transport cable
5) Remove a transport adapter from a node
6) Remove a transport switch
7) Enable a transport cable
8) Disable a transport cable
?) Help
q) Return to the Main Menu
Option:
```

- You still need to know the nature and ordering of tasks.
- Example: Disable cables, remove cables, remove switch, remove adapters.

Even if you use the `clsetup` utility, you still need to know how things are done in the Oracle Solaris Cluster software environment.

For example, if you go to the cluster interconnect submenu, you see the output shown in the slide.

For example, to permanently delete an entire private network, you must perform the following tasks in the correct order:

1. Disable the cable or cables that define the transport.
   This is a single operation for a crossover-cable definition, or multiple operations if a switch is defined (one cable per node connected to the switch).
2. Remove the definition(s) of the transport cable or cables.
3. Remove the definition of the switch.
4. Remove the definitions of the adapters (one per node).

Nothing bad happens if you try to do things in the wrong order; you are just informed that you missed a step. This would be the same with the command line or with `clsetup`.

The `clsetup` command saves you from needing to remember the commands, subcommands, and options.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Controlling Clusters

- Starting and stopping nodes
  - You can boot straight into the cluster by default.
  - Use the regular `init` or `shutdown` commands to halt an individual node.
  - You can use `cluster shutdown` to shut down all nodes.
- Shutting down a cluster
  - After production starts, never shut the cluster.

Basic cluster control includes starting and stopping clustered operation on one or more nodes and booting nodes in non-cluster mode.

Starting and stopping cluster nodes: The Oracle Solaris Cluster software starts automatically during a system boot operation. Use the standard `init` or `shutdown` commands to shut down a single node. Use the `cluster shutdown` command to shut down all nodes in the cluster.

Before shutting down an individual node, you should switch resource groups to the next preferred node and then run `shutdown` or `init` on that node.

Note: After an initial Oracle Solaris Cluster software installation, there are no configured resource groups with which to be concerned.

Shutting down a cluster: You can shut down the entire cluster with the `cluster shutdown` command from any active cluster node. When your cluster is in production and running clustered applications, you should never need to do this. The whole purpose of the Oracle Solaris Cluster environment is that you should always be able to keep at least one node running.

```
clnode1:/# cluster shutdown -y -g0


Apr  9 03:59:49 clnode1 cl_runtime: NOTICE: CMM: node
   reconfiguration #20 completed.
Apr  9 03:59:49 clnode1 cl_runtime: NOTICE: CMM: Quorum device
   /dev/did/rdsk/d4s2: owner set to node 1.
clnode1:/# /etc/rc0.d/K05stoprgm: Calling clzc halt -n clnode1 +
/etc/rc0.d/K05stoprgm: Calling scswitch -S (evacuate)
/etc/rc0.d/K05stoprgm: disabling failfasts
Apr  9 03:59:58 clnode1 syseventd[413]: SIGHUP caught - reloading
   modules
Apr  9 03:59:59 clnode1 syseventd[413]: Daemon restarted
svc.startd: The system is coming down.  Please wait.
svc.startd: 150 system services are now being stopped.
Apr  9 04:00:05 clnode1 cl_eventlogd[3143]: Going down on signal
   15.
Apr  9 04:00:37 clnode1 syslogd: going down on signal 15
Apr  9 04:00:37 rpc.metad: Terminated
svc.startd: The system is down.
syncing file systems...NOTICE: clcomm: Path clnode1:net2 -
   clnode2:net2 being drained
NOTICE: clcomm: Path clnode1:net3 - clnode2:net3 being drained
NOTICE: CMM: Node clnode2 (nodeid = 2) is down.
NOTICE: CMM: Cluster members: clnode1.
TCP_IOC_ABORT_CONN: local = 000.000.000.000:0, remote =
   172.016.004.002:0, start = -2, end = 6
TCP_IOC_ABORT_CONN: aborted 0 connection
NOTICE: CMM: node reconfiguration #22 completed.
 done
NOTICE: CMM: Quorum device /dev/did/rdsk/d4s2: owner set to node 1.
WARNING: CMM: Node being shut down.
Program terminated
{0} ok
```

Booting Nodes into Non-Cluster Mode (1/3)

- For emergency maintenance (for nodes remaining in this cluster, this node still seems dead)
- For SPARC:

```
{1} ok boot -x
Resetting ...
Rebooting with command: boot -x
...
Hostname: clnode1
Reading ZFS config: done.
Mounting ZFS filesystems: (5/5)
clnode1 console login: root
Password:
# cluster status
cluster: (C152734) This node is not in cluster mode.
```


Occasionally, you might want to boot a node without it joining the cluster. This might be to debug some sort of problem preventing a node from joining a cluster, or to perform maintenance. For example, you upgrade the cluster software itself when a node is booted into non-cluster mode. Other nodes might still be up running your clustered applications.

To other nodes that are still booted into the cluster, a node that is booted into non-cluster node looks like it has failed completely. It cannot be reached across the cluster transport.

To boot a node to non-cluster mode, you supply the `-x` boot option, which is passed through to the kernel.

Booting a SPARC platform machine with the `-x` command: For a SPARC-based machine, booting is simple (as shown in the slide).

Booting Nodes into Non-Cluster Mode (2/3)

For x86:

- With the normal Oracle Solaris 11 OS highlighted, press the `e` key to edit the boot parameters.
- Highlight the line that begins with `kernel`. Then press `e` again to edit that specific line and add the `-x`.

```
GNU GRUB  version 0.97  (639K lower / 3668928K upper memory)


 +-----------------------------------------------------------------+
 | Oracle Solaris 11 11/11                                          |
 | solaris-backup-1                                                 |
 +-----------------------------------------------------------------+
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.
The highlighted entry will be booted automatically in 1 second.
```

ORACLE

For an x86 machine, booting is a little more complicated.

As the machine boots, you see the menu shown in the slide, with the normal OS highlighted.

Booting Nodes into Non-Cluster Mode (3/3)

```
GNU GRUB  version 0.97  (639K lower / 3668928K upper memory)
 +-------------------------------------------------------------------------+
 | bootfs rpool/ROOT/solaris                                                |
 | kernel$ /platform/i86pc/kernel/amd64/unix-B $ZFS-BOOTFS                  |
 | module$ /platform/i86pc/amd64/boot_archive                               |
 +-------------------------------------------------------------------------+
      Use the ^ and v keys to select which entry is highlighted.
      Press 'b' to boot, 'e' to edit the selected command in the
      boot sequence, 'c' for a command-line, 'o' to open a new line
      after ('O' for before) the selected line, 'd' to remove the
      selected line, or escape to go back to the main menu.


grub edit> kernel /platform/i86pc/amd64/boot_archive -x
```


With the normal Solaris 11 OS highlighted, press the `e` key to edit the boot parameters. You see the screen as displayed in the slide.

Use the arrows to highlight the line that begins with `kernel`, and then press `e` again to edit that specific line and add the `-x`.

Placing Nodes into Maintenance State

```
# clq disable clnode1
The clquorum status command shows that the possible vote for clnode2 is
    now set to 0.
# clq status
Cluster Quorum ===
--- Quorum Votes Summary ---
            Needed   Present   Possible
            ------   -------   --------
            1        1         1
--- Quorum Votes by Node ---
Node Name       Present       Possible       Status
---------       -------       --------       ------
clnode1         0             0              Offline
clnode2         1             1              Online
--- Quorum Votes by Device ---
Device Name       Present      Possible      Status
-----------       -------      --------      ------
d4                0            0             Offline
```


If you anticipate that a node will be down for an extended period, you can place the node into maintenance state from an active cluster node. This operation is done to affect vote counts of a node that is already down. The maintenance state disables the node's quorum vote. You cannot place an active cluster member into maintenance state.

A typical command is as follows:

clnode2:/# **clq disable clnode1**

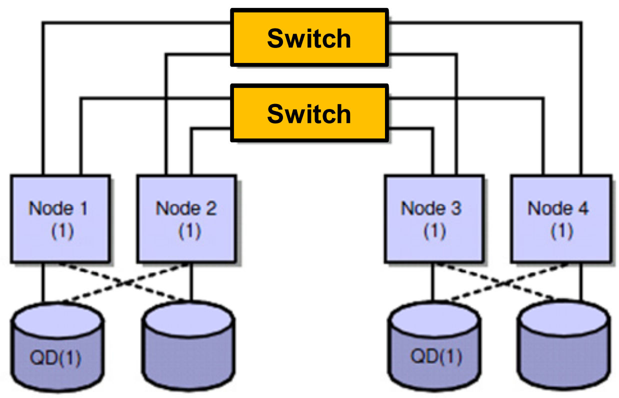
The `clquorum status` command shows that the possible vote for `clnode2` is now set to `0`.

In addition, the vote count for any dual-ported quorum device physically attached to the node is also set to `0`.

You can reset the maintenance state for a node by rebooting the node into the cluster. The node and any dual-ported quorum devices regain their votes.

Maintenance Mode: Example

- You should place nodes in the maintenance state.
  - For example, if you lose a node (for example, Node 4), placing Node 4 into maintenance mode will enable you to lose Node 3 without a panic.
- After you lose Node 3, you can put it into maintenance.



To see the value of placing a node in maintenance state, consider the following topology:

Suppose that there is no way that you can add a quorum device between nodes 2 and 3 as described in an earlier lesson (maybe you have no more available storage controllers).

Now suppose that node 4 has come down. At that point in time, the cluster survives, but you can tell that if you lose node 3, you will lose the whole cluster, because you will have only three of the total possible six quorum votes.

If you put node 4 into maintenance mode, you would eliminate its quorum vote, and the quorum vote of the shared quorum device between nodes three and four. There would temporarily be a total possible value of four quorum votes, making the required number of votes three. This would enable you to survive the death of Node 3.

Agenda

- Identifying the cluster daemons
- Using cluster commands
- Using RBAC with Oracle Solaris Cluster
- Administering cluster global properties
- Administering cluster nodes
- Administering quorum
- Administering disk path monitoring
- Administering SCSI protocol settings for storage devices
- Administering interconnect components
- Using the `clsetup` command
- Controlling cluster operations
- Modifying private network settings while in non-cluster mode

Modifying Private Network Address and Netmask (1/5)

- You must accomplish this process while all nodes are in multi-user, non-cluster mode.
- Run `clsetup` from one node only.
  - It automatically propagates to the other nodes.

The final task presented in this lesson is unique because it must be accomplished while all nodes are in multi-user, non-cluster mode.

In this mode, if you run `clsetup`, it recognizes that the only possible task is to change the private network information, and it guides you through this task. You can choose a different network number, and you can give a different anticipated maximum number of nodes and subnets and use a different suggested netmask.

You run this from one node only, and it automatically propagates to the other nodes. Nodes can communicate by using the same Remote Procedure Calls (RPC) that are used to communicate during `scinstall`.

Refer to the examples shown in the next few slides.

Modifying Private Network Address and Netmask (2/5)

```
# clsetup

  *** Main Menu ***

    Select from one of the following options:

      1) Change Network Addressing and Ranges for the  Cluster Transport
      2) Show Network Addressing and Ranges for the Cluster Transport

      ?) Help with menu options
      q) Quit

    Option:  1
```

Modifying Private Network Address and Netmask (3/5)

```
>>> Change Network Addressing and Ranges for the Cluster Transport <<<

Network addressing for the cluster transport is currently configured as
     follows:

Private Network ===

private_netaddr:                            172.16.0.0
  private_netmask:                          255.255.240.0
  max_nodes:                                64
  max_privatenets:                          10
  num_zoneclusters:                         12
```

Modifying Private Network Address and Netmask (4/5)

```
Do you want to change this configuration (yes/no) [yes]?  yes

The default network address for the cluster transport is 172.16.0.0.

Do you want to use the default (yes/no) [yes]?  no

What network address do you want to use?  192.168.5.0
.
.
.
Maximum number of nodes anticipated for future growth [2]? 4

Maximum number of private networks anticipated for future growth [2]?4
```

Modifying Private Network Address and Netmask (5/5)

```
Specify a netmask of 255.255.255.0 to meet anticipated future requirements
of 4 cluster nodes and 4 private networks.

What netmask do you want to use [255.255.255.0]? <CR>

Is it okay to proceed with the update (yes/no) [yes]?  yes

cluster set-netprops -p private_netaddr=192.168.5.0 -p
private_netmask=255.255.255.0 -p max_nodes=8 -p max_privatenets=4

Command completed successfully.

Press Enter to continue:
```

ORACLE

After you reboot into the cluster, your new private network information is automatically applied by the cluster. For cluster-aware applications, the same `clprivnet0` private hostnames (by default, `clusternode1-priv` and so on) now resolve to the new addresses.

Summary

In this lesson, you should have learned how to:

- Identify the cluster daemons
- Use cluster commands
- Use RBAC with Oracle Solaris Cluster
- Administer cluster global properties
- Administer cluster nodes
- Administer quorum
- Administer disk path monitoring
- Administer SCSI protocol settings for storage devices
- Administer interconnect components
- Use the `clsetup` command
- Perform cluster operations
- Modify private network settings while in non-cluster mode

Practice 6 Overview: Performing Basic Cluster Administration

This practice covers the following topics:

- Task 1: Verifying basic cluster configuration and status
- Task 2: Reassigning a quorum device
- Task 3: Configuring Oracle Solaris Cluster quorum server software
- Task 4: Adding a quorum server quorum device
- Task 5: Preventing cluster amnesia
- Task 6: Changing the Cluster Private IP address range