Oracle® Fusion Middleware

High Availability Guide

12c (12.2.1.4.0)

E95104-06

October 2023

Oracle Fusion Middleware High Availability Guide, 12c (12.2.1.4.0)

E95104-06

Primary Author: Oracle Corporation

Contents

Preface

Part I Introduction to High Availability

1 Introduction and Roadmap

2 High Availability Concepts

6 Scaling Out a Topology (Machine Scale Out)

7 Using Dynamic Clusters

8 JMS and JTA High Availability

9 Administration Server High Availability

Part III Component Procedures

Preface

This preface contains these sections:

Audience

The document is intended for administrators, developers, and others whose role is to deploy and manage Oracle Fusion Middleware with high availability requirements.

Purpose of this Guide

The purpose of this guide is to serve as a reference document to set up a highly available environment. Use this guide in conjunction with your product's installation and administration guides.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

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

Related Documents

Refer to the Oracle Fusion Middleware Library for additional information.

- See About Key Oracle Fusion Middleware Concepts in *Understanding Oracle Fusion Middleware* for information on the common terms and concepts in an Oracle Fusion Middleware environment.

- See Getting Started Managing Oracle Fusion Middleware in *Administering Oracle Fusion Middleware* for information on managing your Oracle Fusion Middleware environment after installation and configuration is complete.
- *Oracle Fusion Middleware Tuning Performance Guide*
- *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*
- For release-related information, see Fusion Middleware Release Notes.

For definitions of unfamiliar terms found in this and other books, see the Glossary.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Part I

Introduction to High Availability

Part I contains the following topics:

- Introduction and Roadmap
- High Availability Concepts
- Whole Server Migration

1 Introduction and Roadmap

Read this section for information on how and why to use this document and high availability environments.

- How to Use This Document
  Use this document to reference information on high availability concepts and tasks as you set up a highly available environment.
- Setting up a Highly Available Environment
  A highly available environment requires preparation and planning before you configure it.
- New and Changed Features in This Release
  Oracle Fusion Middleware 12*c* (12.2.1.4.0) includes new and changed concepts and features.
- What is High Availability?
  High availability is the ability of a system or device to be available when it is needed.
- High Availability Solutions
  You can categorize high availability solutions into local **high availability solutions** that provide high availability in a single data center deployment, and **disaster recovery solutions**.
- About the Oracle Fusion Middleware Standard HA Topology
  Oracle recommends a standard high availability topology that has elements that this topic describes.

How to Use This Document

Use this document to reference information on high availability concepts and tasks as you set up a highly available environment.

Before you use this document, you must have a standard installation topology (SIT) set up for your product. This is the required starting point for setting up high availability. See the topics Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology and Roadmap for Installing and Configuring the Standard Installation Topology to set up the SIT.

Setting up a Highly Available Environment

A highly available environment requires preparation and planning before you configure it.

Table 1-1 describes tasks to set up a highly available environment and additional resources for information.

Table 1-1 Setting up a Highly Available Environment

Task	Description	For more information
Performing administrative tasks and preparing your environment	Common tasks to perform on a newly-created domain.	See Administering and Preparing your WebLogic Domain for High Availability in your product installation guide.
Planning your WebLogic Server Installation	Covers understanding your topology and determining the distribution, components, and features you need.	See Preparing for an Oracle Fusion Middleware Installation in *Planning an Installation of Oracle Fusion Middleware*.
Installing the WebLogic Server Software	Describes how to start the installation process and go through installation screens.	See Installing the Oracle Fusion Middleware Infrastructure Software in your product installation guide.
Configuring a domain	Creating and configuring a domain	See Configuring your Oracle Fusion Middleware Infrastructure Domain in your product installation guide.
Managing Oracle Fusion Middleware	Includes how to: start and stop, change ports, deploy applications, and back up and recover Oracle Fusion Middleware.	See *Oracle Fusion Middleware Administrator's Guide*.
Monitoring and optimizing performance in the Oracle Fusion Middleware environment.	For components that affect performance, use multiple components for optimal performance, and design applications for performance.	See *Oracle Fusion Middleware Tuning Performance Guide*.
Setting up a product-specific enterprise deployment	Oracle best practices blueprints based on proven Oracle high availability and security technologies and recommendations for a product-specific enterprise deployment.	See your product's Enterprise Deployment Guide.
Administering the product environment	To deploy, manage, monitor, and configure applications using the product.	See your product's Administrator's Guide.
Configuring Node Manager	Use Node Manager to start, shut down, and restart the Administration Server and Managed Servers from a remote location. It is an essential tool for a high availability environment.	See *Administering Node Manager for Oracle WebLogic Server*.

New and Changed Features in This Release

Oracle Fusion Middleware 12*c* (12.2.1.4.0) includes new and changed concepts and features.

- Support for WebCenter. See https://docs.oracle.com/en/middleware/webcenter/index.html.

Note:

For a comprehensive list of new and deprecated:

- **WebLogic Server features** in this release, see *Oracle Fusion Middleware What's New in Oracle WebLogic Server.*
- **Terms** in this release, see New and Deprecated Terminology for 12c in *Understanding Oracle Fusion Middleware Concepts*.

What is High Availability?

High availability is the ability of a system or device to be available when it is needed.

A high availability architecture ensures that users can access a system without loss of service. Deploying a high availability system minimizes the time when the system is down, or unavailable, and maximizes the time when it is running, or available.

High availability comes from redundant systems and components. You can categorize high availability solutions by their level of redundancy into **active-active** solutions and **active-passive** solutions.

- Active-Active High Availability Solutions
  An **active-active solution** deploys two or more active servers to improve scalability and provide high availability.
- Active-Passive High Availability Solutions
  An **active-passive solution** deploys one active instance that handles requests and one passive instance that is on standby.

Active-Active High Availability Solutions

An **active-active solution** deploys two or more active servers to improve scalability and provide high availability.

In active-active deployments, all instances handle requests concurrently. Oracle recommends active-active solutions for all single-site middleware deployments.

Active-Passive High Availability Solutions

An **active-passive solution** deploys one active instance that handles requests and one passive instance that is on standby.

If the active node fails, the standby node activates and the middle-tier components continue servicing clients from that node. All middle-tier components fail over to the new active node. No middle-tier components run on a failed node after failover.

Oracle supports active-passive deployments for all components.

High Availability Solutions

You can categorize high availability solutions into local **high availability solutions** that provide high availability in a single data center deployment, and **disaster recovery solutions**.

- Local high availability solutions can protect against process, node, and media failures, as well as human errors, ensuring availability in a single data center deployment.
- Disaster recovery solutions are usually geographically distributed deployments that protect applications from disasters such as floods or regional network outages. You can protect against physical disasters that affect an entire data center by deploying geographically-distributed disaster recovery solutions. For more on disaster recovery for Oracle Fusion Middleware components, see Overview of Disaster Recovery in *Oracle Fusion Middleware Disaster Recovery Guide*.

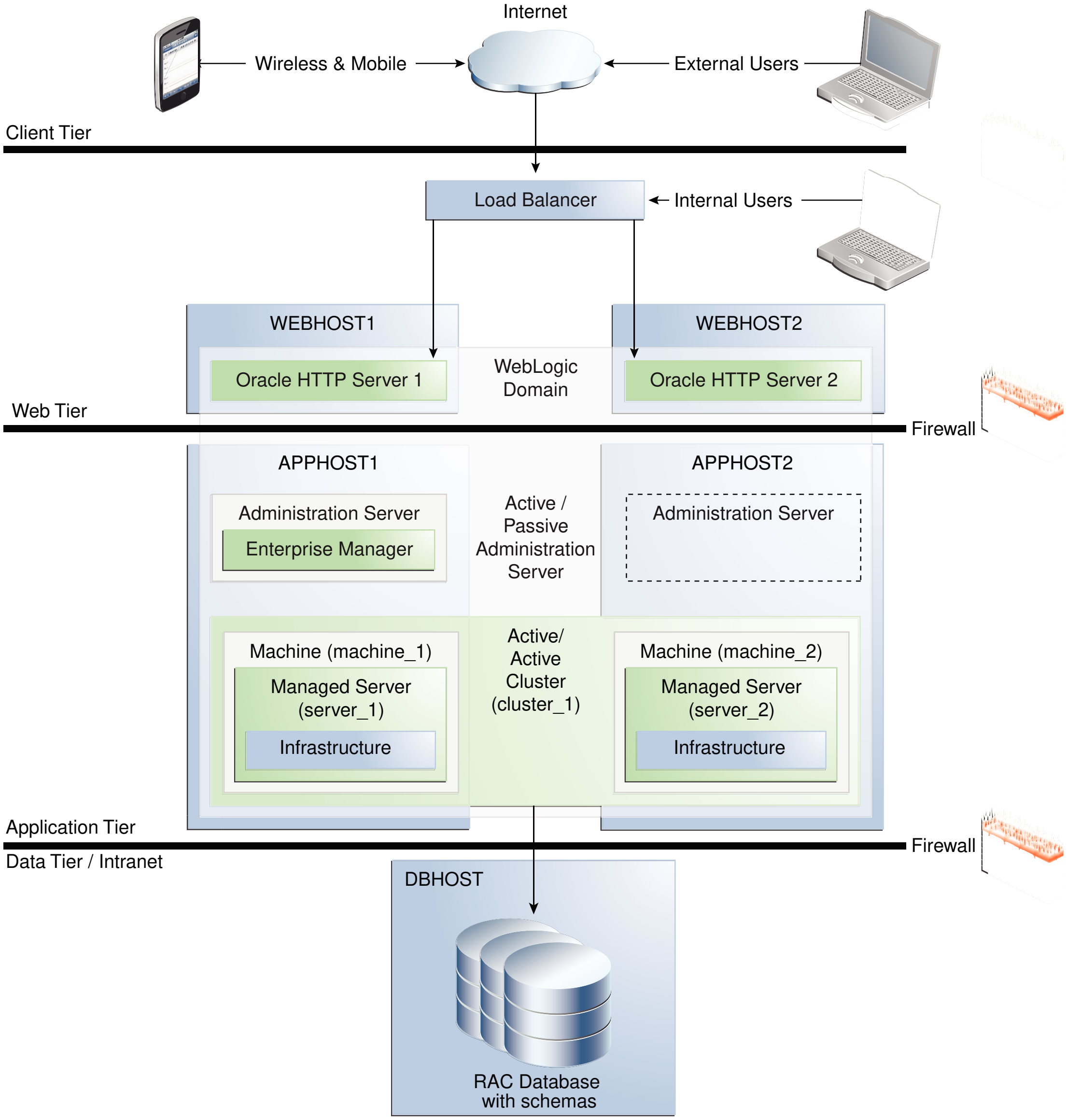
About the Oracle Fusion Middleware Standard HA Topology

Oracle recommends a standard high availability topology that has elements that this topic describes.

Figure 1-1 shows the recommended standard high availability topology for a local, highly available Oracle Fusion Middleware deployment.

This deployment is consistent with the infrastructure SIT and Oracle HTTP Server SIT if you followed steps in Roadmap for Installing and Configuring the Standard Installation Topology and Roadmap for Installing and Configuring Oracle HTTP Server in a WebLogic Server Domain.

Figure 1-1 Oracle Fusion Middleware Highly Available Deployment Topology (Typical Enterprise)



This topology shows a multi-tiered architecture. Users access the system from the client tier. Requests go through a hardware load balancer, which routes them to Web servers running Oracle HTTP Servers in the web tier. Web servers use Proxy Plug-in (`mod_wl_ohs)` to route requests to the WebLogic cluster in the application tier. Applications running on the WebLogic cluster in the application tier then interact with the database cluster in the data tier to service the request.

- Elements in the Standard High Availability Topology
  A Standard High Availability Installation Topology includes certain elements.

Elements in the Standard High Availability Topology

A Standard High Availability Installation Topology includes certain elements.

Table 1-2 describes elements in Figure 1-1.

Table 1-2 Description of Elements in the Oracle Fusion Middleware Infrastructure Standard High Availability Topology

Element	Description and Links to Additional Documentation
APPHOST	Machine that hosts the application tier.
WEBHOST	Machine that hosts the web tier.
WebLogic Domain	A logically related group of Java components, in this case, the Administration Server, Managed Servers, and other software components. For more, see What is an Oracle WebLogic Server Domain? in *Understanding Oracle Fusion Middleware.*
Administration Server	Central control entity of a domain. Maintains a domain's configuration objects and distributes configuration changes to Managed Servers.
Enterprise Manager	Oracle Enterprise Manager Fusion Middleware Control. The main tool that you use to manage a domain.
Cluster	A collection of multiple WebLogic Server instances running simultaneously and working together.
Machine	Logical representation of the computer that hosts one or more WebLogic Server instances (servers). Machines are the logical 'glue' between Managed Servers and Node Manager; to start or stop a Managed Server with Node Manager, the Managed Server must be associated with a machine.
Managed Server	Host for applications, application components, Web services, and their associated resources. See Oracle Enterprise Manager Fusion Middleware Control in *Understanding Oracle Fusion Middleware*.
Infrastructure	Collection of services that includes: • Metadata repository (MDS). Contains metadata for components, such as Oracle Application Developer Framework. See What is the Metadata Repository? in *Understanding Oracle Fusion Middleware*. • Oracle Application Developer Framework (Oracle ADF) • Oracle Web Services Manager (OWSM)

Note:

- To view a figure of the Infrastructure SIT and follow a roadmap to install it, see Understanding the Infrastructure Standard Installation Topology in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
- To view a figure of the Oracle HTTP Server SIT and follow a roadmap to install it, see Introducing the Oracle HTTP Server Standard Installation Topologies in *Installing and Configuring Oracle HTTP Server*.

2

High Availability Concepts

High availability involves elements such as load balancing, failover, Real Application Clusters, and profiles (settings).

- Oracle Fusion Middleware Concepts
  Get familiar with important concepts before scaling out.
- Server Load Balancing in a High Availability Environment
  Typically, a load balancer front ends high availability deployments.
- Application Failover
  There are different types of failover and application behavior.
- Real Application Clusters
  Oracle Real Application Clusters (RAC) enable you to cluster an Oracle database. A **cluster** comprises multiple interconnected computers or servers that appear as if they are one server to end users and applications.
- Coherence Clusters and High Availability
  Every standard installation topology (SIT) includes a standard Coherence cluster. This cluster is a starting point for additional configuration.
- Disaster Recovery
  For maximum availability, you may need to deploy services at different geographical locations to protect against entire site failures due to unforeseen disasters and natural calamities.
- Install Time Configuration (Profiles)
  There are domain, file, and database-based profile types to consider as you configure a SIT.
- Application and Service Failover for JMS and JTA
  Migration in WebLogic Server is the process of moving 1) a clustered WebLogic Server instance or 2) a component running on a clustered instance elsewhere if failure occurs.
- Roadmap for Setting Up a High Availability Topology
  Oracle recommends completing install and configuration steps in a certain order to set up an example high availability topology.

Oracle Fusion Middleware Concepts

Get familiar with important concepts before scaling out.

Table 2-1 describes relevant topics in *Oracle Fusion Middleware Understanding Oracle Fusion Middleware Concepts*.

Table 2-1 Oracle Fusion Middleware Concepts

For information on...	See this topic...
Oracle Home, Oracle Common, WebLogic Server Domain	What Are the Key Oracle Fusion Middleware Directories?

Table 2-1 (Cont.) Oracle Fusion Middleware Concepts

For information on...	See this topic...
WebLogic Server Domain	What Is an Oracle WebLogic Server Domain?
Administration Server	What Is the Administration Server?
Managed Servers and Clusters	Understanding Managed Servers and Managed Server Clusters
Node Manager	What Is Node Manager?

> **Note:**
>
> See Communications in a Cluster in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*

Server Load Balancing in a High Availability Environment

Typically, a load balancer front ends high availability deployments.

- About Load Balancing
  Load balancing is the even distribution of jobs and associated communications across computing and networking resources in your environment.
- Third-Party Load Balancer Requirements
  You can use third-party load balancers in your high availability deployments, as long as they have certain features.
- Configuring Third-Party Load Balancers
  The detailed load balancer configuration steps depend on two elements that are explained in this topic.
- Server Load Balancing with Oracle HTTP Server or Oracle Traffic Director
  Oracle has two options for server load balancing products: Oracle HTTP Server and Oracle Traffic Directory.

About Load Balancing

Load balancing is the even distribution of jobs and associated communications across computing and networking resources in your environment.

You use Oracle HTTP Server or Oracle Traffic Director to configure load balancing between different components or applications. See Server Load Balancing with Oracle HTTP Server or Oracle Traffic Director.

You can also combine of a load balancer and Oracle HTTP Server (as Figure 1-1 shows) to provide maximum availability.

Third-Party Load Balancer Requirements

You can use third-party load balancers in your high availability deployments, as long as they have certain features.

Oracle recommends load balancers that support *sticky* session routing. **Sticky session routing** is the ability, for the entire session, to route traffic to the same server that processes the first request.

External load balancer must have these features:

- Ability to load-balance traffic to a pool of real servers through a virtual host name: clients access services using the *virtual* host name instead of *actual* host names. The load balancer can then load balance requests to servers in the pool. Typically, the load balancer can balance across Oracle HTTP Server instances and then the Oracle HTTP Servers can balance across application servers.
- Port translation configuration.
- Monitoring of ports (HTTP and HTTPS).
- Ability to configure virtual server names and ports on the load balancer.
- Configuration of multiple virtual servers. For each virtual server, the load balancer should allow configuration of traffic management on more than one port. For example, for clusters, you must configure the load balancer with a virtual server and ports for HTTP and HTTPS traffic.
- Virtual server names must be associated with IP addresses and be part of your DNS. Clients must be able to access the external load balancer through virtual server names.
- Resource monitoring/port monitoring/process failure detection: the load balancer must be able to detect service and node failures and to stop directing non-Oracle Net traffic to a failed node. If your external load balancer can automatically detect failures, Oracle recommends that you use it.
- Fault tolerant mode: Oracle recommends that you configure the load balancer to be in fault-tolerant mode.
- Virtual server returning to calling client: Oracle highly recommends that you configure the load balancer virtual server to return immediately to the calling client when back-end services that it forwards traffic to are unavailable. This is preferred over a client disconnecting on its own after a timeout based on TCP/IP settings on a client machine.
- SSL acceleration. Oracle recommends this feature but doesn't require it.
- Client IP Address (Preserving): the load balancer must be able to insert a request's original client IP address in an X-Forwarded-For HTTP header to preserve it.

Configuring Third-Party Load Balancers

The detailed load balancer configuration steps depend on two elements that are explained in this topic.

- The environment you are using the load balancer in.
- The type of load balancer you are using.

For these reasons, Oracle recommends that you follow your load balancer's documentation. For high-level load balancer configuration steps, see the enterprise deployment guide for the component you are working with.

Server Load Balancing with Oracle HTTP Server or Oracle Traffic Director

Oracle has two options for server load balancing products: Oracle HTTP Server and Oracle Traffic Directory.

> **Note:**
>
> As of 12.2.1.4.0, Oracle Traffic Director is deprecated. In the future, for equivalent functionality, use Oracle HTTP Server, Microsoft IIS Web Server, or Apache HTTP Server plug-ins, or a native Kubernetes load balancer, such as Traefik.

Table 2-2 Server Load Balancing Products

Product	Description
Oracle HTTP Server (OHS)	Web server with a built-in WebLogic Server Proxy Plug-In module to act as HTTP front-end for WebLogic servers. Receives incoming requests from clients and load balances each request to WebLogic servers. The OHS `mod_wl_ohs` module routes requests to WebLogic servers. Has the same load balancing functionality as `mod_weblogic` (Oracle WebLogic Server Plug-in for Apache HTTP Server). See Configuring the mod_wl_ohs Plug-In for Oracle HTTP Server in *Oracle Fusion Middleware Using Web Server 1.1 Plug-Ins with Oracle WebLogic Server*. SeeConfiguring mod_wl_ohs.conf for more on the `mod_wl_ohs` module.
Oracle Traffic Director	Fast, reliable, and scalable layer-7 software load balancer. Reliable entry point for all HTTP, HTTPS and TCP traffic. Distributes client requests based on specified load-balancing method, routes requests based on specified rules, caches frequently accessed data, prioritizes traffic, and controls service quality. See Features of Oracle Traffic Director.

Application Failover

There are different types of failover and application behavior.

- About Failover, Application Failover, and State
  Failover is relocating an overloaded or failed resource such as a server, disk drive, or network to its backup location.
- Session Failover Requirements
  For seamless failover, an application must meet certain conditions.
- Application Failover Expected Behavior
  If you configure the environment correctly, users don't notice when an application instance in a cluster becomes unavailable.

About Failover, Application Failover, and State

Failover is relocating an overloaded or failed resource such as a server, disk drive, or network to its backup location.

Application failover is when an application component doing a certain job becomes unavailable and a copy of the failed component finishes the job.

State is information about what has been done on a job. WebLogic Server maintains state information using session replication and replica-aware stubs. If a component unexpectedly stops doing its job, replication techniques enable a copy of the component to pick up where the failed component stopped and finish the job.

Session Failover Requirements

For seamless failover, an application must meet certain conditions.

> **Note:**
>
> Oracle applications meet these session failover requirements unless a specific exception is made for an application.

An application must meet these conditions to failover seamlessly:

- The application is in a cluster and at least one member of the cluster is available to serve the request.
- For stateful applications, state replication is configured correctly.
- If you use Oracle HTTP Server, the server is configured with the WebLogic Cluster directive to balance among all available application instances.
- If you are using a hardware load balancer, the load balancer is:
  - Routing traffic to all available instances
  - Configured correctly with a health monitor to mark unavailable instances
  - Configured to support persistence of session state

Application Failover Expected Behavior

If you configure the environment correctly, users don't notice when an application instance in a cluster becomes unavailable.

The application failover sequence of events is (for example):

1. A user makes a request. A hardware load balancer routes it to Instance A of the application.
2. Instance A of the application becomes unavailable due to node failure, process failure, or network failure.
3. The hardware load balancer marks Instance A as unavailable.
4. The user makes another request. The request is routed to Instance B.

5. Instance B is configured as a replication partner of Instance A and has the user's session state.
6. The application resumes using the session state on Instance B. The user continues working without interruption.

> **Note:**
>
> See *Domain Template Reference* for domain and extension templates that support high availability.
>
> See Failover and Replication in a Cluster in *Administering Clusters for Oracle WebLogic Server* for more on failover and replication at the application level.

Real Application Clusters

Oracle Real Application Clusters (RAC) enable you to cluster an Oracle database. A **cluster** comprises multiple interconnected computers or servers that appear as if they are one server to end users and applications.

Oracle RAC uses Oracle Clusterware for the infrastructure to bind multiple servers so that they operate as one system. Along with a collection of hardware (cluster), Oracle RAC unites the processing power of each component to become a single, robust computing environment. Oracle RAC provides a highly scalable and highly available database for Oracle Fusion Middleware.

Every Oracle RAC instance in a cluster has equal access and authority. Node and instance failure may affect performance but don't result in downtime because the database service is available or can be made available on surviving server instances.

> **Note:**
>
> For more on Oracle RAC see:
>
> - Introduction and Roadmap in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server*
> - Overview of Oracle Clusterware in *Oracle Clusterware Administration and Deployment Guide*
> - Overview of Oracle RAC in *Oracle Real Application Clusters Administration and Deployment Guide*

Coherence Clusters and High Availability

Every standard installation topology (SIT) includes a standard Coherence cluster. This cluster is a starting point for additional configuration.

A **Coherence cluster** is a collection of Java Virtual Machine (JVM) processes running Coherence. In 12c, these processes are called **WebLogic Managed Coherence Servers**. JVMs that join a cluster are **cluster members** or **cluster nodes**. Cluster members can be:

- Dedicated storage members
- Client members that have storage disabled
- Proxy members that allow non-cluster members to access Coherence caches

Cluster members communicate using Tangosol Cluster Management Protocol (TCMP). Cluster members use TCMP for both multicast communication (broadcast) and unicast communication (point-to-point communication).

Coherence characteristics include the following:

- Each domain typically contains one Coherence Cluster.
- Each managed Coherence server in a domain is associated with a Coherence cluster, defined through a Coherence Cluster System Resource.
- Each application has its Coherence configuration in a Grid Archive (GAR) file, which deploys with the application and to all dedicated storage nodes.

All applications that use Coherence use the cluster associated with the managed Coherence server and deploy their GAR files co-located with their applications. Table 2-3 lists sources of information about Coherence.

Table 2-3 Coherence and Coherence Clusters

For information on...	See this topic...
Coherence concepts and features	Introduction to Coherence in *Oracle Fusion Middleware Developing Applications with Oracle Coherence*
Creating Coherence clusters	Setting Up a WebLogic Server Domain Topology for Coherence in *Coherence Administrator's Guide*
Configuring a Coherence Cluster	Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server*

Disaster Recovery

For maximum availability, you may need to deploy services at different geographical locations to protect against entire site failures due to unforeseen disasters and natural calamities.

Oracle Fusion Middleware products support the configuration of a geographically separate standby site to act as a backup. Applications and services can fail over to this backup in the event of natural or unplanned outages at a production site.

For more on disaster recovery, see Introduction to Oracle Fusion Middleware Disaster Recovery.

Install Time Configuration (Profiles)

There are domain, file, and database-based profile types to consider as you configure a SIT.

- Domain (Topology) Profiles
  Use the Configuration Wizard or WebLogic Scripting Tool (offline) to set up domains.
- Persistence Profile Types
  Persistence profiles are a collection of settings that Oracle recommends for a specific persistence environment. There are two primary persistence profiles: database and file based.

- Post-Configuration Defaults
  When you complete a standard installation, a domain is set up with a file-based persistence profile.

Domain (Topology) Profiles

Use the Configuration Wizard or WebLogic Scripting Tool (offline) to set up domains.

See Creating a WebLogic Domain in *Creating WebLogic Domains Using the Configuration Wizard* to create, update, and configure domains.

12*c* (12.2.1.4.0) installation guides have steps to set up a single machine, multi-server domain, which is the **standard installation topology**. About the Oracle Fusion Middleware Standard HA Topology describes this topology in detail. See:

- About the Oracle HTTP Server Installation in *Oracle Fusion Middleware Installing and Configuring Oracle HTTP Server*
- Configuring the Oracle Fusion Middleware Infrastructure Domain in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Persistence Profile Types

Persistence profiles are a collection of settings that Oracle recommends for a specific persistence environment. There are two primary persistence profiles: database and file based.

Table 2-4 shows persistence types for both profiles.

Although you can *mix & match* a component or service with the persistence profile, the persistence type groups work together optimally. Oracle recommends that you use all options consistently within their respective profile.

Table 2-4 Persistence Types for Database and File Persistence Profiles

Component/Service	Database Persistence Profile	File Persistence Profile
JMS	WLS JMS in Database Store	WebLogic Server JMS in File Store
JTA	JTA in Database Store	JTA in File Store
OPSS	Database Store	Database Store
MDS	Database Store	Database Store
Service Table	Database Store	Database Store
Failover	Whole Server Migration	Whole Server Migration

> **Note:**
>
> An MDS data source has a WebLogic Server file persistence store allocated along with the data source. Because you use the file persistence store only in development mode, you can ignore it for high availability purposes. You don't need to recover the file persistence store in the event of failure.

Note:

See Interoperability with Supported Databases in the *Interoperability and Compatibility Guide* for database version requirements for selected products and features.

Post-Configuration Defaults

When you complete a standard installation, a domain is set up with a file-based persistence profile.

To configure database-based persistence for JMS/JTA resources, see JMS and JTA High Availability. To set up whole server migration, see Whole Server Migration .

Table 2-5 describes additional sources of information.

Note:

Some products may have specific requirements for shared file stores; Oracle recommends that you refer to your product's requirements for details.

Table 2-5 Domain Configuration Topics

For additional information on...	See this topic...
Shared file systems to use with a file persistence profile	Using Shared Storage
JMS and JTA	About Migratable Targets for JMS and JTA Services
Failover	Application Failover

Application and Service Failover for JMS and JTA

Migration in WebLogic Server is the process of moving 1) a clustered WebLogic Server instance or 2) a component running on a clustered instance elsewhere if failure occurs.

Whole server migration occurs when a server instance migrates to a different physical system upon failure.

Service-level migration is when services move to a different server instance within the cluster.

See these topics for server and service failover for JMS and JTA.

- About Automatic Service Migration (JMS/JTA)
  Service-level migration in WebLogic Server is the process of moving pinned services from one server instance to a different available server instance in the cluster.

About Automatic Service Migration (JMS/JTA)

Service-level migration in WebLogic Server is the process of moving pinned services from one server instance to a different available server instance in the cluster.

You can configure JMS and JTA services for high availability by using migratable targets. A **migratable target** is a special target that can migrate from one server in a cluster to another. A migratable target provides a way to group migratable services that should move together. High availability is achieved by migrating a migratable target from one clustered server to another when a problem occurs on the original server. When the migratable target migrates, all services that the target hosts migrate.

From release 12*c* (12.2.1.4.0), you can use the **High Availability Options** screen in the Configuration Wizard to automate the service-level migration configuration for the Fusion Middleware components. This screen appears for the first time when you create a cluster that uses Automatic Service Migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

If a product doesn't support Automatic Service Migration, you can use whole service migration.

> **Note:**
>
> See Service Migration in *Oracle Fusion Middleware Administering Clusters for Oracle* for these topics:
>
> - Understanding the Service Migration Framework
> - Pre-Migration Requirements
> - Roadmap for Configuring Automatic Migration of JMS-related Services
> - Roadmap for Configuring Automatic Migration of the JTA Transaction Recovery Service

Roadmap for Setting Up a High Availability Topology

Oracle recommends completing install and configuration steps in a certain order to set up an example high availability topology.

Table 2-6 describes high level steps required to set up an example middleware topology with high availability.

Table 2-6 Roadmap for Setting Up a High Availability Topology

Task	Description	Documentation
1. Install Real Application Clusters	Install Real Application Clusters	Install Oracle Database 12c Software with Oracle RAC in *Oracle Real Application Clusters Administration and Deployment Guide*

Table 2-6 (Cont.) Roadmap for Setting Up a High Availability Topology

Task	Description	Documentation
2. Install and configure middleware components	Install and configure the application by following instructions in an application installation guide.	Roadmap for Installing and Configuring the Standard Installation Topology in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure* or the installation guide for your product
3. Install and configure Oracle HTTP Server	Install and configure Oracle HTTP Server in the same domain	Roadmap for Installing and Configuring Oracle HTTP Server in a WebLogic Server Domain in *Installing and Configuring Oracle HTTP Server*
4. Configure a load balancer	Configure a third- party load balancer that meets specific requirements, or Oracle HTTP Server/Oracle Traffic Director.	Server Load Balancing in a High Availability Environment.
5. Scale out the topology (machine scale out)Oracle Fusion MiddlewareOracle Fusion Middleware	Steps for scaling out a topology (machine scale-out) for all products that are part of a Fusion Middleware WebLogic Server domain.	Scaling Out a Topology (Machine Scale Out)
6. Configure high availability for the Administration Server	Configure high availability for the Administration Server	Administration Server High Availability

3 Whole Server Migration

When **whole server migration** occurs, a server instance migrates to a different physical machine upon failure. You must configure whole server migration for Managed Servers if your environment uses special (pinned) services such as JMS and JTA.

- About Whole Server Migration
  WebLogic Server has **migratable servers** to make JMS and the JTA transaction system highly available.
- Configuring Whole Server Migration for Managed Server Failover
  If you configure Managed Server failover and one server in a cluster fails, whole server migration restarts a Managed Server on another machine.

About Whole Server Migration

WebLogic Server has **migratable servers** to make JMS and the JTA transaction system highly available.

A cluster provides high availability and failover by duplicating an object or service on redundant Managed Servers in the cluster. However, some services, such as JMS servers and JTA transaction recovery service, are designed with the assumption that there is only *one* active instance of the service in a cluster at any given time. These types of services are known as **pinned services** because they remain active on only one server at a time.

Most services deploy homogeneously on all Managed Servers in a cluster. With this setup, they can failover transparently from one Managed Server to another. However, pinned services target *one* Managed Server in a cluster. For pinned services, WebLogic Server supports failure recovery with migration instead of failover.

Migratable servers provide for both automatic and manual migration at the server-level, rather than the service level.

When a migratable server is unavailable for any reason (for example, if it hangs, loses network connectivity, or its host system fails), migration is automatic. Upon failure, a migratable server automatically restarts on the same system if possible. If a migratable server can't restart on the system it failed on, it migrates to another system. Also, you can manually start migration of a server instance.

Note:

See Whole Server Migration in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server* to prepare for automatic whole server migration, configure automatic whole server migration, and server migration processes and communications.

See Service Details in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server* for details on service migration mechanisms.

See JMS and JTA High Availability for details on JMS and JTA services.

Configuring Whole Server Migration for Managed Server Failover

If you configure Managed Server failover and one server in a cluster fails, whole server migration restarts a Managed Server on another machine.

Your system must meet specific requirements before you configure automatic whole server migration. See Preparing for Automatic Whole Server Migration in *Administering Clusters for Oracle WLS*.

To configure whole server migration, follow steps in Configuring Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.

See these topics for more on configuring whole server migration:

- Using High Availability Storage for State Data
- Server Migration Processes and Communications

Part II

Creating a High Availability Environment

Part II describes recommendations and steps you need to take to create a high availability environment.

This part contains the following topics:

4
Using Shared Storage

Oracle recommends locating specific artifacts in shared storage for a high availability environment.
There are benefits to placing artifacts in a common location that multiple hosts or servers share. This common location typically resides in a shared file system, which is mounted on each server with standard operating system protocols such as NFS and CIFS.

About Shared Storage

Shared storage allows sharing of dynamic state and server configuration. It simplifies administration, configuration, failover, and backup/recovery.

In a highly available environment, shared storage is *required* when you use file based persistent stores (for JMS and JTA logs) and certain Oracle products. Shared storage is *optional* for product binaries and domain directories.

The following artifacts are typical candidates to place on a shared file system:

- **Product binaries**: All files and directories related to product executables, JAR files, and scripts that install during product installation.
- **Domain directory**: Directory containing WebLogic Server domains and their configuration.
- **File-based persistent stores**: File-based persistent stores for JMS persistence and JTA transaction logs.

Table 4-1 has more information about shared storage.

Table 4-1 Shared Storage Topics

Topic/Task	For More Information
Structure and contents of an Oracle home	Understanding the Oracle Fusion Middleware Infrastructure Directory Structure in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*
Saving JMS and JTA information in a file store	The WebLogic Persistent Store in *Administering the WebLogic Server Persistent Store*. Persistent Store High Availability in *Administering JMS Resources for Oracle WebLogic Server*. Default File Store Availability for JTA in *Administering Clusters for Oracle WebLogic Server*.

Shared Storage Prerequisites

There are shared storage prerequisites that apply only when you use file-based persistent stores.

Following are the list of shared storage prerequisites:

- For proper recovery in the event of a failure, you must store both JMS and JTA transaction logs in a location that is accessible to all nodes that can resume operations after a Managed Server failure. This setup requires a shared storage location that multiple nodes can reference. See Directory Structure and Configurations for the recommended directory structure.
- Oracle recommends that you use a shared storage device that is network-attached storage (NAS) or storage area network (SAN).

  If you use NFS-mounted systems, issues related to file locking and abrupt node failures have been detected. See Using File Stores on NFS and check with your storage vendor for the main recommended parameters for mount options.

  The following example command is based on a NAS device. Note: your options may be different from those in this example; see UNIX/Linux documentation for more on the mount command.

  ```
  mount nasfiler:/vol/vol1/u01/oracle /u01/oracle -t nfs -o
  rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsize=32768
  ```
- For maximum availability, Oracle recommends a *highly available* NAS or SAN device for shared storage. Shared storage devices that are not highly available can be a single point of failure. Check with your storage provider for options to achieve this.

For more on saving JMS and JTA information in a file store, see The WebLogic Persistent Store in *Administering the WebLogic Server Persistent Store*.

Using Shared Storage for Binary (Oracle Home) Directories

Oracle has guidelines for using shared storage for your Oracle home directories.

- About the Binary (Oracle Home) Directories
  When you install any Oracle Fusion Middleware product, you install product binaries into an Oracle home (*ORACLE_HOME*). The binary files are read-only

and don't change unless you patch or upgrade the Oracle home to a newer version

- About Sharing a Single Oracle Home
  You can configure multiple servers from one Oracle home. The benefit is that you can install the Oracle home in one location on a shared volume and reuse the Oracle home for multiple servers.
- About Using Redundant Binary (Oracle Home) Directories
  For maximum availability, Oracle recommends using redundant binary installations on shared storage.

About the Binary (Oracle Home) Directories

When you install any Oracle Fusion Middleware product, you install product binaries into an Oracle home (*ORACLE_HOME*). The binary files are read-only and don't change unless you patch or upgrade the Oracle home to a newer version

In a typical production environment, you save Oracle home files in a separate location from domain configuration files, which you create using the Configuration Wizard.

The Oracle home for an Oracle Fusion Middleware installation contains binaries for Oracle WebLogic Server, Oracle Fusion Middleware infrastructure files, and any Oracle Fusion Middleware product-specific directories.

> **Note:**
>
> The Configuration Wizard writes its logs to the `logs` directory in Oracle home. If you use a read-only Oracle home, you must specify the `-log` option to redirect logs to a different directory.

> **Note:**
>
> For more on the Oracle home structure and contents, see What are the Key Oracle Fusion Middleware Directories? in *Oracle Fusion Middleware Understanding Oracle Fusion Middleware Concepts*.

About Sharing a Single Oracle Home

You can configure multiple servers from one Oracle home. The benefit is that you can install the Oracle home in one location on a shared volume and reuse the Oracle home for multiple servers.

If multiple servers on different hosts share an Oracle home, there are some best practices to keep in mind. For example, because the Oracle inventory directory (`oraInventory`) is updated only on the host from which the Oracle home was originally installed, Oracle recommends that you perform all subsequent operations on the Oracle home (such as patching and upgrade) from that original host. If that host is unavailable, ensure that the Oracle inventory is updated on another host before you apply patches or upgrades to the Oracle home from the other host.

For more about `oraInventory`, see Oracle Universal Installer Inventory.

About Using Redundant Binary (Oracle Home) Directories

For maximum availability, Oracle recommends using redundant binary installations on shared storage.

In this model:

1. You install two identical Oracle homes for your Oracle Fusion Middleware software on two different shared volumes.
2. You then mount one of the Oracle homes to one set of servers and the other Oracle home to the remaining servers.

   Each Oracle home has the same mount point, so the Oracle home always has the same path, regardless of which Oracle home the server is using.

If one Oracle home becomes corrupted or unavailable, only half your servers are affected. For additional protection, Oracle recommends that you disk mirror these volumes. To restore affected servers to full functionality, you can simply remount the surviving Oracle Home.

If separate volumes are not available on shared storage, Oracle recommends simulating separate volumes using different directories within the same volume and mounting these to the same mount location on the host side. Although this doesn't guarantee the protection that multiple volumes provide, it does protect from user deletions and individual file corruption.

> **Note:**
>
> For maximum protection, Oracle recommends that you evenly distribute the members of a cluster across redundant binary Oracle homes. This is particularly important if cluster members are not running on all available servers.

Using Shared Storage for Domain Configuration Files

There are guidelines for using shared storage for the Oracle WebLogic Server domain configuration files that you create when you configure Oracle Fusion Middleware products in an enterprise deployment.

- About Oracle WebLogic Server Administration and Managed Server Domain Configuration Files
  When you configure an Oracle Fusion Middleware product, you create or extend an Oracle WebLogic Server domain. Each domain consists of a single Administration Server and one or more Managed Servers.
- Shared Storage Considerations for Administration Server Configuration Directory
  Oracle does not require you to store domain configuration files in shared storage. However, to support Administration Server recovery, you must place the Administration Server configuration directory on shared storage and mount it on the host that the Administration Server runs on.

- Shared Storage Considerations for Managed Server Configuration Files
  Oracle recommends that you keep Managed Server configuration files in local, or, host private, storage.

About Oracle WebLogic Server Administration and Managed Server Domain Configuration Files

When you configure an Oracle Fusion Middleware product, you create or extend an Oracle WebLogic Server domain. Each domain consists of a single Administration Server and one or more Managed Servers.

WebLogic uses a replication protocol to push persisted changes on the Administration Server to all Managed Servers. This gives redundancy to the Managed Servers so that you can start them without the Administration Server running. This mode is called *Managed Server independence*.

For more information about Oracle WebLogic Server domains, see Understanding Oracle WebLogic Server Domains.

Shared Storage Considerations for Administration Server Configuration Directory

Oracle does not require you to store domain configuration files in shared storage. However, to support Administration Server recovery, you must place the Administration Server configuration directory on shared storage and mount it on the host that the Administration Server runs on.

If that host fails, you can mount the directory on a different host and bring up the failed Administration Server on the other host. See Administration Server High Availability .

Shared Storage Considerations for Managed Server Configuration Files

Oracle recommends that you keep Managed Server configuration files in local, or, host private, storage.

You can keep Managed Server configuration files on shared storage. However, doing so can affect performance because multiple servers concurrently access the same storage volume.

Shared Storage Requirements for JMS Stores and JTA Logs

When you use file-based persistence in a high availability setup, you must configure the JMS persistent stores and JTA transaction log directories to reside in shared storage.

See Using File Persistence.

Directory Structure and Configurations

When you use shared storage, there are multiple ways to lay out storage elements. Oracle recommends certain best practices.

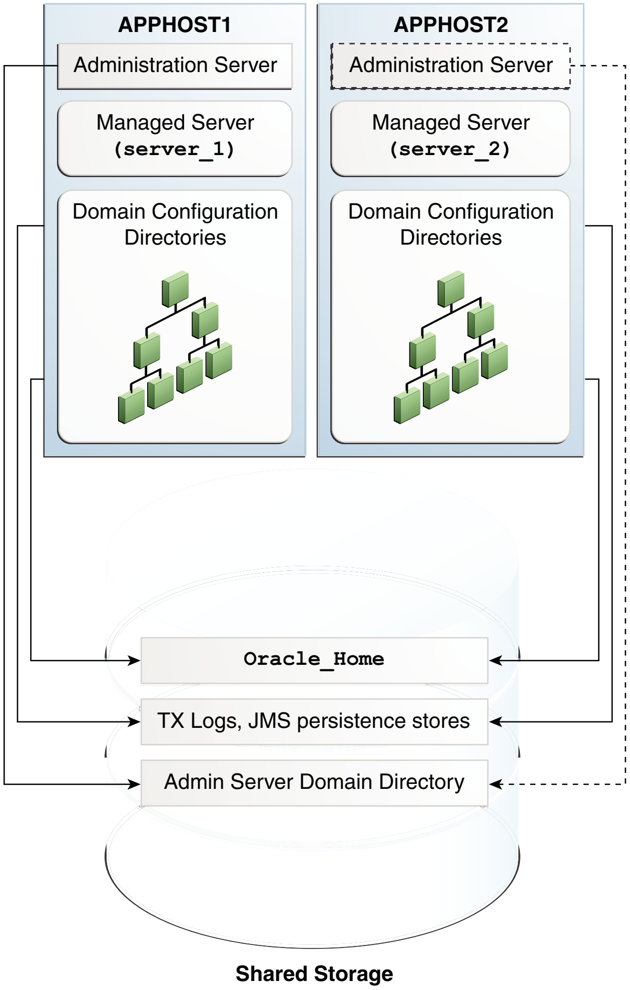
Table 4-2   Shared Storage Elements Directory Structure

Element	Location
ORACLE_HOME	Share in read-only mode by all servers.
JMS file stores and Transaction logs	Place on shared storage if you use file-based persistence.
Administration Server domain configuration directory	Place in shared storage to facilitate failing over the Administration Server to a different host.

> **Note:**
>
> Place Managed Server domain configuration directories on storage that is local to the corresponding host. See Shared Storage Considerations for Administration Server Configuration Directory.

Figure 4-1 illustrates the directory structure.

Figure 4-1   Shared Storage Directory Structure



5 Database Considerations

As you configure database connections for Oracle Fusion Middleware in a high availability setup, you must make decisions about Oracle Real Application Clusters (Oracle RAC). Oracle RAC is a commonly-deployed database high availability solution. Most components use a database as the persistent store for their data. When you use an Oracle database, you can configure it in a variety of highly available configurations.

For more information about database options, see Oracle Database High Availability Overview in *High Availability Overview and Best Practices*.

- About Oracle Real Application Clusters
  A **cluster** comprises multiple interconnected computers or servers that appear as one server to end users and applications. With Oracle RAC, you can cluster an Oracle database so that it is highly scalable and highly available.
- Roadmap for Setting Up Oracle Real Application Clusters
  Use this roadmap to set up Oracle RAC.
- About RAC Database Connections and Failover
  To establish connection pools, Oracle Fusion Middleware supports Active GridLink data sources and multi data sources for the Oracle RAC back end (for both XA and non-XA JDBC drivers). These data sources also support load balancing across Oracle RAC nodes.
- About Data Sources
  A **data source** is an abstraction that components use to obtain connections to a relational database.
- Configuring Active GridLink Data Sources with Oracle RAC
  You configure component data sources as Active GridLink data sources for a RAC database during domain creation.
- Configuring Multi Data Sources
  There are multiple tools available to configure multi data sources.

About Oracle Real Application Clusters

A **cluster** comprises multiple interconnected computers or servers that appear as one server to end users and applications. With Oracle RAC, you can cluster an Oracle database so that it is highly scalable and highly available.

All Oracle Fusion Middleware components that you deploy to Oracle WebLogic Server support Oracle RAC.

Every Oracle RAC instance in a cluster has equal access and authority. Node and instance failure may affect performance, but doesn't result in downtime; the database service is available or can be made available on surviving server instances.

Roadmap for Setting Up Oracle Real Application Clusters

Use this roadmap to set up Oracle RAC.

Table 5-1 outlines tasks and information to set up Oracle RAC.

Table 5-1 Roadmap for Setting up Oracle RAC

Task/Topic	More Information
About Oracle RAC	Introduction to Oracle RAC in *Oracle Real Application Clusters Administration and Deployment Guide*
Installing Oracle RAC	*Oracle Real Application Clusters Administration and Deployment Guide*
Managing Oracle RAC	Overview of Managing Oracle RAC Environments in *Oracle Real Application Clusters Administration and Deployment Guide*
Configuring and tuning GridLink and multi data sources	*Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*
Configuring Single Client Access Name (SCAN) URLs. (To specify the host and port for TNS and ONS listeners in WebLogic console.)	SCAN Addresses in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

About RAC Database Connections and Failover

To establish connection pools, Oracle Fusion Middleware supports Active GridLink data sources and multi data sources for the Oracle RAC back end (for both XA and non-XA JDBC drivers). These data sources also support load balancing across Oracle RAC nodes.

If an Oracle RAC node or instance fails, WebLogic Server or Oracle Thin JDBC driver redirects session requests to another node in the cluster. Existing connections don't failover. However, new application connection requests are managed using existing connections in the WebLogic pool or by new connections to the working Oracle RAC instance.

If the database is the transaction manager, in-flight transactions typically roll back.

If WebLogic Server is the transaction manager, in-flight transactions fail over; they are driven to completion or rolled back based on the transaction state when failure occurs.

For more on XA Transactions, see About XA Transactions.

- About XA Transactions
  XA transaction support enables access to multiple resources (such as databases, application servers, message queues, transactional caches) within one transaction.

About XA Transactions

XA transaction support enables access to multiple resources (such as databases, application servers, message queues, transactional caches) within one transaction.

A *non-XA transaction* always involves just one resource.

An XA transaction involves a coordinating transaction manager with one or more databases, or other resources such as JMS, all involved in a single global transaction.

Java EE uses the terms **JTA transaction**, **XA transaction**, **user transaction**, and **global transaction** interchangeably to refer to one global transaction. This transaction type may include operations on multiple different XA- capable or non-XA resources and even different resource types. A JTA transaction is always associated with the current thread, and may pass from server to server as one application calls another. A common example of an XA transaction is one that includes both a WebLogic JMS operation and a JDBC (database) operation.

About Data Sources

A **data source** is an abstraction that components use to obtain connections to a relational database.

Connection information, such as the URL or user name and password, is set on a data source object as properties. The application's code does not need to explicitly define the properties. Due to this abstraction, you can build applications in a portable manner, because they are not tied to a specific back-end database. The database can change without affecting application code.

Active GridLink data sources and multi data sources support database connection high availability, load balancing, and failover. Oracle recommends the following data source types depending on your Oracle RAC database version:

- If you use Oracle RAC database version 11g Release 2 and later, use Active GridLink data sources.
- If you use an Oracle RAC database version earlier than 11g Release 2 or a non-Oracle database, use multi data sources.

> **Note:**
>
> Oracle recommends using Active GridLink data sources with Oracle RAC database for maximum availability. For Oracle RAC database versions that don't support Active GridLink data sources, Oracle recommends using multi data sources for high availability.

- Active GridLink Data Sources
  An **Active GridLink data source** provides connectivity between WebLogic Server and an Oracle database service, which may include multiple Oracle RAC clusters.
- Multi Data Sources
  A **multi data source** is an abstraction around a group of data sources that provides load balancing or failover processing. Multi data sources support load balancing for both XA and non-XA data sources.

Active GridLink Data Sources

An **Active GridLink data source** provides connectivity between WebLogic Server and an Oracle database service, which may include multiple Oracle RAC clusters.

An Active GridLink data source has features of generic data sources, plus the following support for Oracle RAC:

- Uses the ONS to respond to state changes in an Oracle RAC.
- Responds to Fast Application Notification (FAN) events to provide Fast Connection Failover (FCF), Runtime Connection Load-Balancing, and RAC instance graceful shutdown. FAN is a notification mechanism that Oracle RAC uses to quickly alert applications about configuration and workload.
- Provides Affinities (or XA Affinity) policies to ensure all database operations for a session are directed to the same instance of a RAC cluster for optimal performance.
- SCAN Addresses
- Secure Communication Using Oracle Wallet

See Using Active GridLink Data Sources in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server* for more on the following topics:

- What is an Active GridLink Data Source
- Using Socket Direct Protocol
- Configuring Connection Pool Features
- Configuring Oracle Parameters
- Configuring an ONS Client
- Tuning Active GridLink Data Source Connection Pools
- Monitoring GridLink JDBC Resources

Multi Data Sources

A **multi data source** is an abstraction around a group of data sources that provides load balancing or failover processing. Multi data sources support load balancing for both XA and non-XA data sources.

A **failover multi data source** provides an ordered list of data sources to fulfill connection requests. Normally, every connection request to this kind of multi data source is served by the first data source in the list.

A **load-balancing multi data source** chooses from a circular list of datasources in a round robin method. The stream of incoming connection requests is spread evenly around the datasources. If a database connection test fails and the connection can't be replaced, or if the data source is suspended, a connection is sought from the next data source on the list.

Multi data sources are bound to the JNDI tree or local application context just like regular data sources. Applications look up a multi data source on the JNDI tree or in the local application context (`java:comp/env`) just as they do for data sources, and then request a database connection. The multi data source determines which data source to use to satisfy the request depending on the algorithm selected in the multi data source configuration: load balancing or failover.

> **Note:**
>
> To configure Multi Data Sources with Oracle RAC, see Using Multi Data Sources with Oracle RAC in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

Configuring Active GridLink Data Sources with Oracle RAC

You configure component data sources as Active GridLink data sources for a RAC database during domain creation.

How you configure an Active GridLink data source depends on:

- The Oracle component that you are working with
- The domain you are creating

> **Note:**
>
> To create and configure Active GridLink data sources, see Using Active GridLink Data Sources in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

- Requirements to Configure Component Data Sources as Active Gridlink Data Sources
  Your system must meet certain requirements before you configure component data sources as Active GridLink data sources to use with an Oracle RAC database.
- Configuring Component Data Sources as Active GridLink Data Sources
  You configure component data sources as Active GridLink data sources for a RAC database during domain creation.
- Using SCAN Addresses for Hosts and Ports
  Oracle recommends using Oracle Single Client Access Name (SCAN) addresses to specify the host and port for the TNS listener and ONS listener in the WebLogic console.

Requirements to Configure Component Data Sources as Active Gridlink Data Sources

Your system must meet certain requirements before you configure component data sources as Active GridLink data sources to use with an Oracle RAC database.

- You use Oracle RAC database version 11g Release 2 or later.
- You have run RCU to create component schemas.
- You use the Configuration Wizard to create or configure a domain and have arrived at the JDBC Component Schema screen where you select **Component Datasources**.

Configuring Component Data Sources as Active GridLink Data Sources

You configure component data sources as Active GridLink data sources for a RAC database during domain creation.

To configure component data sources as Active GridLink data sources:

1. In the JDBC Component Schema screen, select one or more component schemas to configure GridLink data sources for.
2. Select **Convert to GridLink** then select **Next**.
3. In the GridLink Oracle RAC Component Schema screen, select one of the GridLink JDBC drivers.
4. In the **Service Name** field, enter the service name of the database using lowercase characters. For example, `mydb.example.com`.
5. In the **Schema Owner** field, enter the name of the database schema owner for the corresponding component.
6. In the **Schema Password** field, enter the password for the database schema owner.
7. In the **Service Listener**, **Port**, and **Protocol** field, enter the SCAN address and port for the RAC database being used. The protocol for Ethernet is TCP; for Infiniband it is SDP. Click **Add** to enter multiple listener addresses.

   You can identify the SCAN address by querying the appropriate parameter in the database using the TCP protocol:

   ```
   show parameter remote_listener


   NAME TYPE VALUE

   --------------------------------------------------------------------

   remote_listener string db-scan.example.com:1521
   ```

   You can also identify the SCAN address by using the `srvctl config scan` command. Use the command `srvctl config scan_listener` to identify the SCAN listener port.
8. Select **Enable FAN** to receive and process FAN events. Enter one or more ONS daemon listen addresses and port information. Select **Add** to enter more entries.

   > **Note:**
   >
   > Verify that the ONS daemon listen address(es) that you enter is valid. The domain creation process do not validate the address.

To determine the Scan (ONS) port, use the RAC `srvctl` command on the Oracle Database server, as the following example shows:

```
srvctl config nodeapps -s

ONS exists: Local port 6100, remote port 6200, EM port 2016
```

9. Select **Enable SSL** for SSL communication with ONS. Enter the Wallet File, which has SSL certificates, and the Wallet Password.
10. Select **Next**. Verify that all connections are successful.

> **Note:**
>
> See:
>
> - JDBC Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard* for information about the JDBC Component Schema screen.
> - GridLink Oracle RAC Component Schema in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard* for information about configuring component schemas.
> - Using Active GridLink Data Sources in *Administering JDBC Data Sources for Oracle WebLogic Server* for information on GridLink RAC data sources.

Using SCAN Addresses for Hosts and Ports

Oracle recommends using Oracle Single Client Access Name (SCAN) addresses to specify the host and port for the TNS listener and ONS listener in the WebLogic console.

You do not need to update an Active GridLink data source containing SCAN addresses if you add or remove Oracle RAC nodes. Contact your network administrator for appropriately configured SCAN URLs for your environment. See SCAN Addresses in *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

Configuring Multi Data Sources

There are multiple tools available to configure multi data sources.

- Oracle Fusion Middleware Configuration Wizard (during WebLogic Server domain creation)
- Oracle WebLogic Server Administration Console
- WLST Commands

- Configuring Multi Data Sources with Oracle RAC
  Configuring Multi Data Sources with Oracle RAC has specific requirements and steps that you must complete.
- Configuring Multi Data Sources for MDS Repositories
  You can configure applications that use an MDS database-based repository for high availability Oracle database access.

Configuring Multi Data Sources with Oracle RAC

Configuring Multi Data Sources with Oracle RAC has specific requirements and steps that you must complete.

- Requirements to Configure Multi Data Sources with Oracle RAC
  Verify that your system meets the following requirements before you configure component data sources as multi data sources to use with an Oracle RAC database.
- Configuring Component Data Sources as Multi Data Sources
  When you configure component data sources as multi data sources, you select data sources to convert then enter database information.
- About Adding Multi Data Sources For RAC Databases
  Multi data sources have constituent data sources for each RAC instance that provides a database service. If you add an instance to the RAC back end, Oracle recommends adding an additional data source to the multi data source on the Fusion Middleware tier.
- Modifying or Creating Multi Data Sources After Initial Configuration
  For multi data sources that you create manually or modify after initial configuration, Oracle strongly recommends specific XA and non-XA data source property values for optimal high availability. Make changes only after careful consideration and testing if your environment requires that you do so.
- Troubleshooting Warning Messages (Increasing Transaction Timeout for XA Data Sources)
  If WARNING messages in server logs have the following exception, you may need to increase the XA timeout value in your setup.
- Configuring Schemas for Transactional Recovery Privileges
  You want to enable WebLogic Server transaction manager to perform schema tasks.

Requirements to Configure Multi Data Sources with Oracle RAC

Verify that your system meets the following requirements before you configure component data sources as multi data sources to use with an Oracle RAC database.

- You are using an Oracle RAC database.
- You have run RCU to create component schemas.
- You are using the Configuration Wizard to create or configure a domain and have arrived at the JDBC Component Schema Screen where you select Component Schemas. Before you arrive at the JDBC Component Schema screen, you must select the option **Manual Configuration** in the Database Configuration Type screen.

Configuring Component Data Sources as Multi Data Sources

When you configure component data sources as multi data sources, you select data sources to convert then enter database information.

To configure component data sources as multi data sources:

1. In the Component Datasources screen, select one or more component schemas to configure RAC Multiple data sources for.
2. Select **Convert to RAC multi data source** then select **Next**.
3. In the Oracle RAC Multi Data Source Component Schema screen, the JDBC driver **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:10 and later**.
4. In the **Service Name** field, enter the database service name enter in lowercase, for example, `mydb.example.com`.
5. In the **Schema Owner** field, enter the username of the database schema owner for the corresponding component.
6. In the **Schema Password** field, enter the password for the database schema owner.
7. In the **Host Name**, **Instance Name**, and **Port** field, enter the RAC node hostname, database instance name, and port. Click **Add** to enter multiple listener addresses.
8. Click **Next**.Verify that all connections are successful.

About Adding Multi Data Sources For RAC Databases

Multi data sources have constituent data sources for each RAC instance that provides a database service. If you add an instance to the RAC back end, Oracle recommends adding an additional data source to the multi data source on the Fusion Middleware tier.

When you migrate a database from a non-RAC to a RAC database, you must create an equivalent, new multi data source for each affected data source. Multi data sources that you create must have constituent data sources for each RAC instance. Data source property values must be identical to the original single instance data source for properties in Configuring Multi Data Sources with Oracle RAC. For example, if a single instance data source driver is `oracle.jdbc.xa.client.OracleXADataSource`, it must be `oracle.jdbc.xa.client.OracleXADataSource` for each constituent data source of the new multi data source.

Modifying or Creating Multi Data Sources After Initial Configuration

For multi data sources that you create manually or modify after initial configuration, Oracle strongly recommends specific XA and non-XA data source property values for optimal high availability. Make changes only after careful consideration and testing if your environment requires that you do so.

Table 5-2 describes XA and non-XA data source property values that Oracle recommends.

Table 5-2 Recommended Multi Data Source Configuration

Property Name	Recommended Value
test-frequency-seconds	5
algorithm-type	Load-Balancing

For individual data sources, Oracle recommends the configuration values in Table 5-3 for high availability environments. Oracle recommends that you set any other parameters according to application requirements.

Table 5-3 XA Data Source Configuration

Property Name	Recommended Value
Driver	oracle.jdbc.xa.client.OracleXADataSource
Property command	<property> <name>oracle.net.CONNECT_TIMEOUT</name> <value>10000</value> </property>
initial-capacity	0
connection-creation-retry-frequency-seconds	10
test-frequency-seconds	300
test-connections-on-reserve	true
test-table-name	SQL SELECT 1 FROM DUAL
seconds-to-trust-an-idle-pool-connection	0
global-transactions-protocol	TwoPhaseCommit
keep-xa-conn-till-tx-complete	true
xa-retry-duration-seconds	300
xa-retry-interval-seconds	60

Troubleshooting Warning Messages (Increasing Transaction Timeout for XA Data Sources)

If WARNING messages in server logs have the following exception, you may need to increase the XA timeout value in your setup.

```
[javax.transaction.SystemException: Timeout during commit processing
```

To increase the transaction timeout for the XA Data Sources setting, use the Administration Console:

1. Access the data source configuration.
2. Select the **Transaction** tab.
3. Set the XA Transaction Timeout to a larger value, for example, **300**.
4. Select the **Set XA Transaction Timeout** checkbox. You *must* select this checkbox for the new XA transaction timeout value to take effect.
5. Click **Save**.

Repeat this configuration for all individual data sources of an XA multi data source.

Table 5-4 Non-XA Data Source Configuration

Property Name	Recommended Value
Driver	oracle.jdbc.OracleDriver

Table 5-4 (Cont.) Non-XA Data Source Configuration

Property Name	Recommended Value
Property to set	<property> <name>oracle.net.CONNECT_TIMEOUT</name> <value>10000</value> </property>
initial-capacity	0
connection-creation-retry-frequency-seconds	10
test-frequency-seconds	300
test-connections-on-reserve	true
test-table-name	SQL SELECT 1 FROM DUAL
seconds-to-trust-an-idle-pool-connection	0
global-transactions-protocol	None

Configuring Schemas for Transactional Recovery Privileges

You want to enable WebLogic Server transaction manager to perform schema tasks.

You must have sysdba privileges to enable transaction manager privileges:

- Query for transaction state information.
- Issue the appropriate commands, such as commit and rollback, during recovery of in-flight transactions after a WebLogic Server container failure.

To configure schemas for transactional recovery privileges:

1. Log on to SQL*Plus as a user with sysdba privileges. For example:

   ```
   sqlplus "/ as sysdba"
   ```

2. Grant select on `sys.dba_pending_transactions` to the `appropriate_user`.
3. Grant force any transaction to the `appropriate_user`.

Configuring Multi Data Sources for MDS Repositories

You can configure applications that use an MDS database-based repository for high availability Oracle database access.

With this configuration, failure detection, recovery, and retry by MDS (and by WebLogic infrastructure) protect application read-only MDS operations from Oracle RAC database planned and unplanned downtimes

The Fusion Middleware Control navigation tree exposes multi data sources as MDS repositories. You can select these multi data sources when you customize application deployment and use them with MDS WLST commands.

- Configuring an application to retry read-only operations

  To configure an application to retry the connection, configure the `RetryConnection` attribute of the application's MDS AppConfig MBean. See *Oracle Fusion Middleware Administrator's Guide*.

- Registering an MDS multi data source

  In addition to steps in Configuring Multi Data Sources with Oracle RAC, note the following:

  - You must configure child data sources that comprise a multi data source used for an MDS repository as non-XA data sources.
  - A multi data source's name must have the prefix `mds-`. This ensures it is recognized as an MDS repository.

  > **Note:**
  >
  > When you add a MDS data source as a child of a multi data source, this data source is no longer exposed as an MDS repository. It does not appear under the Metadata Repositories folder in the Fusion Middleware Control navigation tree. You can not perform MDS repository operations on it and it does not appear in the list of selectable repositories during deployment.

- Converting a data source to a multi data source

  Keep in mind when you convert a data source to a multi data source:

  - To create a new multi data source with a new, unique name, redeploy the application and select this new multi data source as the MDS repository during deployment plan customization.
  - To avoid redeploying the application, you can delete the data source and recreate a new multi data source using the same name and jndi-name attributes.

6 Scaling Out a Topology (Machine Scale Out)

Steps to scale out a topology (machine scale-out) are similar for all Fusion Middleware products that are a part of a WebLogic Server domain. To enable high availability, it is important to provide failover capabilities to another host computer. When you do so, your environment can continue to serve your application consumers if a computer goes down.

- About Machine Scale Out
  Scalability is the ability of a piece of hardware or software, or a network, to *expand* or *shrink* to meet future needs and circumstances. A scalable system can handle increasing numbers of requests without adversely affecting response time and throughput.
- Roadmap for Scaling Out Your Topology
  When your product is installed, configured, and has a cluster of Managed Servers, use this roadmap to scale out your topology.
- Optional Scale Out Procedure
  If you follow a standard installation topology (SIT), you have multiple Managed Servers assigned to a single host computer.
- About Scale Out Prerequisites
  Before you start the scale out process, you must have a **standard installation topology** (SIT) set up for your product. A SIT is the starting point for scale out.
- Resource Requirements
  Before you scale out the topology, verify that your environment meets certain requirements.
- Creating a New Machine
  A **machine** is the logical representation of the computer that hosts one or more WebLogic Server instances (Managed Servers). In a WebLogic domain, the machine definitions identify physical units of hardware and are associated with Managed Servers that they host.
- Configuring WLS JMS After Machine Scale Up or Scale Out
  You must manually recreate all JMS system resources on a new Managed Server when you add one to a cluster.
- Packing the Domain on APPHOST1
  You create a Managed Server template by running the `pack` command on a WebLogic domain.
- Preparing the New Machine
  To prepare the new machine, **machine_2**, verify that **APPHOST2** can access the shared disk where the Oracle home is installed, or install the same software that you installed on **machine_1**.
- Running Unpack to Transfer the Template
  To unpack the template and transfer the *domain_name*.jar file from **APPHOST1** to **APPHOST2**, run the unpack command.
- Starting the Node Manager
  You use Node Manager to start Managed Servers (using the Administration Console or Fusion Middleware Control).

- Starting Managed Servers
  You start Managed Servers in the Administration Console.
- Verifying Machine Scale Out
  To determine if the machine scale out succeeded, verify that the server status is **RUNNING** (after you use Administration Console to start it).
- Configuring Multicast Messaging for Clusters
  You configure clusters to use messaging so that they can communicate whether or not services are available and other information.

About Machine Scale Out

Scalability is the ability of a piece of hardware or software, or a network, to *expand* or *shrink* to meet future needs and circumstances. A scalable system can handle increasing numbers of requests without adversely affecting response time and throughput.

Machine scale-out is moving a server, one of many on one machine, to another machine for high availability. Machine scale out is different from Managed Server **scale up**, which is adding a new Managed Server to a machine that already has one or more Managed Servers running on it. See Scaling Up Your Environment in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*.

Roadmap for Scaling Out Your Topology

When your product is installed, configured, and has a cluster of Managed Servers, use this roadmap to scale out your topology.

Table 6-1 describes typical steps you must take to scale out a topology.

If you already have a SIT (as *Installing and Configuring the Oracle Fusion Middleware Infrastructure* and product installation guides such as *Installing and Configuring Oracle SOA Suite and Business Process Management* describe), you do *not* need to repeat Resource Requirements, Creating a New Machine, and Configuring WLS JMS After Machine Scale Up or Scale Out steps.

Table 6-1 has start-to-finish steps if you did not complete SIT steps.

Table 6-1 Roadmap for Scaling Out Your Topology

Task	Description	More Information
Product is ready for scale out	Product is installed, configured, and a cluster of Managed Servers is available; your product is in a SIT	About Scale Out Prerequisites
Verify that you meet resource requirements	You must verify that your environment meets certain requirements	Resource Requirements
Create a new machine and assign servers to it	Use the Administration Console to create a new machine and add Managed Servers to it.	Creating a New Machine

Table 6-1 (Cont.) Roadmap for Scaling Out Your Topology

Task	Description	More Information
Create a new JMS server and target it	Create a new JMS server and target it to the new Managed Server	Configuring WLS JMS After Machine Scale Up or Scale Out
Run the pack command	Pack up the domain directory	Packing the Domain on APPHOST1
Prepare the new machine	Install the same software that you installed on the first machine	Preparing the New Machine
Run the unpack command	Create a Managed Server template.	Running Unpack to Transfer the Template
Start the server	Starts the Managed Server on the new machine	Starting Managed Servers
Verify the topology	Test the new setup	Verifying Machine Scale Out
Set the cluster messaging mode to Multicast	Modifies messaging mode from Unicast to Multicast (preferred for multi-server domains)	Configuring Multicast Messaging for Clusters

> **Note:**
>
> *APPHOST* refers to a physical host computer. *Machine* refers to the WebLogic Server machine definition describing that host. See Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

Optional Scale Out Procedure

If you follow a standard installation topology (SIT), you have multiple Managed Servers assigned to a single host computer.

This set up is the most flexible way to create and scale out a domain topology so that it can meet changing requirements. It allows you to 1) create and validate a single-host domain, which is targeted to a single machine on a single host computer, and then 2) *retarget* the Managed Servers to additional machines, when additional computing resources are required. Also, it facilitates troubleshooting; you can validate the basic domain then scale up and scale out (and troubleshoot) at a later time.

However, if you know ahead of time what your target topology is, you can create additional machines during domain creation then just run pack and unpack steps.

If you assigned Managed Servers to target machines during installation or through an online administrative operation, skip Creating a New Machine when you go through the roadmap (Roadmap for Scaling Out Your Topology).

See Creating a New Machine in your product installation guide for more on machine mapping.

About Scale Out Prerequisites

Before you start the scale out process, you must have a **standard installation topology** (SIT) set up for your product. A SIT is the starting point for scale out.

If you followed the steps in your product installation guide, you should have a SIT. For an example, see the SIT that Understanding the Oracle Fusion Middleware Infrastructure Standard Installation Topology describes in *Oracle Fusion Middleware Installing and Configuring the Oracle Fusion Middleware Infrastructure*.

> **Note:**
>
> For more on the SIT, see your product's installation guide or About the Standard Installation Topology in *Planning an Installation of Oracle Fusion Middleware*.

> **Note:**
>
> Application tier products in this release don't support dynamic clusters. **Dynamic clusters** consist of server instances that can dynamically scale up to meet your application resource needs. A dynamic cluster uses a single server template to define configuration for a specified number of generated (dynamic) server instances.

Resource Requirements

Before you scale out the topology, verify that your environment meets certain requirements.

- At least one machine runs multiple Managed Servers configured with a product. This is the result of following your product installation guide or administration guide to add additional servers.
- A host computer in addition to your starting host computer.
- Each host computer can access the Oracle home that contains the product binaries by one of the following means:
  - Shared disk with binaries from the original installation
  - Dedicated disk with a new installation (matches the original installation)
  - Dedicated disk with a clone of the binaries from the original installation

  See Using Shared Storage.
- Sufficient storage available for the domain directory.
- Access to the same Oracle or third-party database used for the original installation.
- A shared disk for JMS and transaction log files (required when using a file persistence profile).

Creating a New Machine

A **machine** is the logical representation of the computer that hosts one or more WebLogic Server instances (Managed Servers). In a WebLogic domain, the machine definitions identify physical units of hardware and are associated with Managed Servers that they host.

Follow steps in this section to create a new machine.

- Shutting Down the Managed Server
  The Managed Server must be shut down before moving to a new machine.
- Creating a New Machine (Using the Administration Console)
  You create a new machine to host Managed Servers and identify physical units of hardware. Typically you use the Administration Console to create a new machine.
- Assigning Managed Servers to a New Machine
  You assign Managed Servers to the new machine to specify which servers the machine hosts.

Shutting Down the Managed Server

The Managed Server must be shut down before moving to a new machine.

If it is up and running, see the topic Shut Down a Server Instance in Administration Console Online Help to shut down the Managed Server that the new machine will host. See Shut Down Server Instances in a Cluster if the Managed Server is in a cluster.

Creating a New Machine (Using the Administration Console)

You create a new machine to host Managed Servers and identify physical units of hardware. Typically you use the Administration Console to create a new machine.

To use WLST to create a new machine, see Creating a New Machine for Certain Components in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*.

> **Note:**
>
> The machine you create in this procedure must have a listen address on a specific network interface, not just a local host.

To create a new machine in the domain:

1. Start the domain Administration Server if it isn't running. Go to the *DOMAIN_HOME*/bin directory and run:

   ```
   ./startWeblogic.sh
   ```

2. When the Administration Server is running, access the Administration Console. Open a web browser and enter the URL:

   http://*hostname*:*port*/**console**

   On the Welcome screen, log in.

3. In the Administration Console Change Center, click **Lock & Edit**.

> **Note:**
>
> For production domains, Oracle recommends creating the domain in **Production** mode, which enables Change Center. If Production mode is *not* enabled, Change Center steps are not required.

4. Under Domain Structure, expand Environment then click **Machines**.
5. Above the Machines table (above Summary), click **New**.
6. In the Create a New Machine screen, enter a name for the machine, such as **machine_2**. Select the **Machine OS** using the drop-down list then click **Next**.
7. On the next screen, for **Type:**, use the drop-down list to select **Plain**.

   For the Node Manager **Listen Address**, enter the IP address or host name of the host computer that will represent this machine. Both machines appear in the Machines table.
8. Click **Finish**.
9. In the Change Center, click **Activate Changes**.

   The message **All changes have been activated. No restarts are necessary.** opens to indicate that you have a new machine.

Assigning Managed Servers to a New Machine

You assign Managed Servers to the new machine to specify which servers the machine hosts.

To add Managed Servers to a newly-created machine, use the Administration Console:

1. In the Change Center, click **Lock & Edit**.
2. In the Machines table, select the checkbox for **machine_1**.
3. Click the machine name (represented as a hyperlink).
4. Under the Settings for **machine_1**, click the Configuration tab then the Servers subtab.
5. Above the Servers table, click **Add**.
6. On the Add a Server to Machine screen, click **Create a new server and associate it with this machine**. Click **Next** then enter the **Server Name** and the **Server Listen Port** in the fields (required).
7. Under Domain Structure, click **Machines**.

   In the Machines table, click the machine **machine_2**.
8. Under the Settings for **machine_2**, click the Configuration tab and then the Servers tab. Above the Servers tab, click **Add**.

   On the **Add a Server to Machine** screen, select the button **Select an existing server**, **and associate it with this machine**.

   Use the Select a server drop-down list to choose **server_2** then select **Finish**.

   The message **Server created successfully** appears.

9. Verify that all Managed Servers have the correct Server Listen Address. Under Domain Structure, click **Servers**. In the Servers table, click the name of the Managed Server. Select the Configuration tab. Verify/set the Listen Address to the IP address of the machine it is associated with. Click **Save**.

10. To complete the changes, go back to the Change Center. Click **Activate Changes**. The message **All changes have been activated. No restarts are necessary.** appears.

    To see a summary of Managed Server assignments, under Domain Structure, under Environment, click **Servers**. The Servers table shows all servers in the domain and their machine assignments.

Configuring WLS JMS After Machine Scale Up or Scale Out

You must manually recreate all JMS system resources on a new Managed Server when you add one to a cluster.

New JMS system resources are cloned from an existing Managed Server in the cluster. New JMS system resources must have unique names in the domain. When you create a domain, the Configuration Wizard creates JMS system resource names that follow a pattern. For ease of use and manageability, Oracle recommends that you follow the same naming pattern.

To configure JMS resources on a new Managed Server `server_`*n*:

1. In the **Domain Structure** tree, select **Services** then select **Messaging**. Select **JMS Servers** to open the JMS Servers table.
2. In the Name column, identify all JMS servers that target one of the existing Managed Servers in the cluster, for example, `server_1`.

   The JMS server name format is *ResourceName*_**auto**_*number*.

   - *ResourceName* is the resource's identifying name
   - *number* identifies the JMS server; servers with the suffix 1 or 2 were created when the domain was created.
3. Note the name of the JMS server and its persistent store. For example, `UMSJMSServer_auto_1` and `UMSJMSFileStore_auto_1`.
4. Click **New** to create a new JMS server.

   a. Name the JMS server for `server_`*n* using the same format as `server_1`. For example, `UMSJMSServer_auto_`*n*.

   b. For the **Persistent Store**, click **Create a New Store**.

   c. For **Type**, select the same persistence profile used that the JMS Server uses on `server_1`. Click **Next**.

   d. Enter a persistent store in the **Name** field. Use the same format as the `server_1` persistent store. For example, `UMSJMSFileStore_auto_`*n*.

   e. In the **Target** field, select the migratable target for migratable target `server_`*n* `(migratable)` from the drop down list.

   f. In the **Directory** field, use the same name as the persistent store. Click **OK** to create the persistent store.

   g. In the Create a New JMS Server screen, select the new persistent store and click **Next**.

   h. For JMS Server Target, select the migratable target for `server_`*n* (migratable) from the drop down list.

   i. Click **Finish** to create the JMS server.

5. Update Subdeployment targets for the corresponding JMS Modules to include the new JMS server:

   a. In the Administration Console's **Domain Structure** tree, expand the **Services** node, expand the **Messaging** node, and select **JMS Modules** to open the JMS Modules table.

   b. Identify the JMS system module that corresponds to the JMS server; its name has a common root. For example for JMS server `UMSJMSServer_auto_1`, the JMS module name is `UMSJMSSystemResource`. Click on the JMS module (a hyperlink) in the Name column to open the Module Settings page.

   c. Open the Subdeployments tab and click on the subdeployment for the JMS module in the Name column.

   d. Select the new JMS Server `server_`*n*. Click **Save**.

   e. Go back to the module Settings page. Select the **Targets** tab. Verify that **All servers in the cluster** is enabled.

   f. Click **Save** if you changed anything.

You now have a new Managed Server that has configured JMS resources in the cluster.

Packing the Domain on APPHOST1

You create a Managed Server template by running the `pack` command on a WebLogic domain.

> **Note:**
>
> The Administration Server should be running on APPHOST1 when you go through pack and unpack steps.

Run the `pack` command on **APPHOST1** to create a template pack. You unpack the template file on **APPHOST2** later in the scale out process. (Running Unpack to Transfer the Template describes unpack steps.)

For example:

```
ORACLE_HOME/oracle_common/common/bin/pack.sh \
     -domain=DOMAIN_HOME \
     -template=dir/domain_name.jar \
     -managed=true \
     -template_name="DOMAIN"
```

In the preceding example:

- Replace *DOMAIN_HOME* with the full path to the domain home directory.
- Replace *dir* with the full path to a well-known directory where you will create the new template file.

- Replace *domain_name* in the JAR file name with the domain name. This is the name of the template file that you create with the `pack` command. For example: `mydomain.jar`.

> **Note:**
>
> See pack and unpack Command Reference in *Creating WebLogic Domains and Domain Templates* for more information about creating a Managed Server template.

Preparing the New Machine

To prepare the new machine, **machine_2**, verify that **APPHOST2** can access the shared disk where the Oracle home is installed, or install the same software that you installed on **machine_1**.

For example, if you are scaling out an Oracle Fusion Middleware Infrastructure domain, verify that **APPHOST2** can access the Infrastructure Oracle home.

> **Note:**
>
> If you use shared storage, you can reuse the same installation location.

> **Note:**
>
> If you are running a new installation or reusing binaries by means of shared storage, the path location of the new files must match the original machine's path location exactly.

Running Unpack to Transfer the Template

To unpack the template and transfer the *domain_name*.jar file from **APPHOST1** to **APPHOST2**, run the unpack command.

For example:

```
ORACLE_HOME/oracle_common/common/bin/unpack.sh \
    -domain=user_projects/domains/base_domain2 \
    -template=/tmp/domain_name.jar \
    -app_dir=user_projects/applications/base_domain2
```

Starting the Node Manager

You use Node Manager to start Managed Servers (using the Administration Console or Fusion Middleware Control).

To start Node Manager, run:

```
DOMAIN_HOME/bin/startNodeManager.sh &
```

If you use machine scoped Node Manager, see Using Node Manager in *Administering Node Manager for Oracle WebLogic Server* for more on Node Manager start options.

Starting Managed Servers

You start Managed Servers in the Administration Console.

To start Managed Servers:

1. In the left pane of the Console, expand **Environment**, and select **Servers**.
2. In the **Servers** table, click the name of the Managed Server that you moved to the new machine. Select the **Control** tab.
3. Select the check box next to the name of the server(s) you want to start and click **Start** to start the server(s).

> **Note:**
>
> To use WLST commands or Fusion Middleware Control to start Managed Servers, see Starting and Stopping Managed Servers in *Oracle Fusion Middleware Administering Oracle Fusion Middleware*.

Verifying Machine Scale Out

To determine if the machine scale out succeeded, verify that the server status is **RUNNING** (after you use Administration Console to start it).

Configuring Multicast Messaging for Clusters

You configure clusters to use messaging so that they can communicate whether or not services are available and other information.

- About Multicast and Unicast Messaging for Clusters
  Clusters use messaging to broadcast the availability of services, and heartbeats that indicate continued availability. You configure clusters to use Unicast or Multicast messaging.
- Requirements to Configure Multicast Messaging
  Before you configure Multicast messaging, verify that your system meets system and network requirements.
- Configuring Multicast Messaging
  You configure Multicast messaging in the Administration Console by selecting the cluster you want to enable it for then selecting the Multcast Messaging mode.

About Multicast and Unicast Messaging for Clusters

Clusters use messaging to broadcast the availability of services, and heartbeats that indicate continued availability. You configure clusters to use Unicast or Multicast messaging.

- **Multicast** is a simple broadcast technology. Multiple applications subscribe to an IP address and port number then 'listen' for messages. Multicast set up is more complex than Unicast because it needs hardware configuration and support.
- **Unicast** provides TCP-based, reliable, one-to-many communication. It is easier to set up than multicast.

When you create clusters in the Configuration Wizard, Unicast is the default cluster messaging mode. When the number of Managed Servers in a domain increases, Oracle recommends Multicast messaging.

Requirements to Configure Multicast Messaging

Before you configure Multicast messaging, verify that your system meets system and network requirements.

- A configured domain with at least one cluster.
- A hardware configuration set up to support Multicast in your network.
- The IP address and port numbers for Multicast communications in the cluster. A multicast address is an IP address between 224.0.0.0 and 239.255.255.255. (Weblogic uses default value of 239.192.0.0).
- You have run the MulticastTest utility to verify that your environment can support Multicast. The utility debugs Multicast problems; it sends out multicast packets and returns data about how effectively multicast works in your network.

> **Note:**
>
> See Communications In a Cluster in *Administering Clusters for Oracle WebLogic Server* for details on Multicast messaging and cluster configuration.

Configuring Multicast Messaging

You configure Multicast messaging in the Administration Console by selecting the cluster you want to enable it for then selecting the Multcast Messaging mode.

To configure Multicast Messaging:

1. Log in to the Administration Console.
2. Shut down all servers that you want to use for Multicast messaging.
3. In the Domain Structure pane on the left, expand **Environment** and click **Clusters**.
4. Select the cluster name that you want to enable Multicast for.
5. Click the **Configuration** tab then the **Messaging** tab.

6. For Messaging Mode, select **Multicast**. Enter the Multicast Address then the Multicast Port.

   The **Multicast Address** must be unique for each cluster. See Cluster Multicast Address and Port in *Oracle Fusion Middleware Administering Clusters for Oracle WebLogic Server* for guidelines on Multicast addresses.

7. Click **Advanced** to configure these parameters:

Table 6-2 Multicast Advanced Parameters

Parameter	Description
Multicast Send Delay	Amount of time (between 0 and 250) in milliseconds to delay sending message fragments over multicast to avoid OS-level buffer overflow.
Multicast TTL	Number of network hops (between 1 and 255) that a cluster multicast message can travel. If your cluster spans multiple subnets in a WAN, this value must be high enough to ensure that routers don't discard multicast packets before they reach their final destination. This parameter sets the number of network hops a multicast message makes before a router can discard a packet.
Multicast Buffer Size	Multicast socket send/receive buffer size (at least 64 kilobytes).
Idle Periods Until Timeout	Maximum number of periods a cluster member waits before timing out a cluster member.
Enable Data Encryption	Enables multicast data encryption. Only multicast data is encrypted; Multicast header information isn't encrypted.

8. Click **Save**.
9. Restart all servers in the cluster.

The cluster can now use Multicast messaging. When you select **Clusters** in the Domain Structure pane in the Administration Console, **Multicast** appears for **Cluster Messaging Mode**.

7

Using Dynamic Clusters

A **dynamic cluster** is a cluster that contains one or more dynamic servers. A **dynamic server** is a server instance that gets its configuration from a server template. This is in contrast to Managed Servers, which require you to configure each server individually.

- About Dynamic Clusters
  Dynamic clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your application. A dynamic cluster uses a single server template to define configuration for a specified number of generated (dynamic) server instances.
- Why Do You Use Dynamic Clusters?
  With dynamic clusters, you can easily scale up your cluster when you need additional server capacity by simply starting one or more of the preconfigured dynamic server instances. You do not need to manually configure a new server instance and add it to the cluster or perform a system restart.
- How Do Dynamic Clusters Work?
  You set the number of managed server instances that you want, and create or select a template for them.
- Creating Dynamic Clusters in a High Availability Topology
  You can create dynamic clusters to scale out a high availability topology.
- Expanding or Reducing Dynamic Clusters
  When you create a dynamic cluster, WebLogic Server generates the number of dynamic servers you specify. Before you decide upon the number of server instances, ensure you have the performance capacity to handle the desired number.

About Dynamic Clusters

Dynamic clusters consist of server instances that can be dynamically scaled up to meet the resource needs of your application. A dynamic cluster uses a single server template to define configuration for a specified number of generated (dynamic) server instances.

When you create a dynamic cluster, the dynamic servers are preconfigured and automatically generated for you, enabling you to easily scale up the number of server instances in your dynamic cluster when you need additional server capacity. You can simply start the dynamic servers without having to first manually configure and add them to the cluster.

If you need additional server instances on top of the number you originally specified, you can increase the maximum number of servers instances (dynamic) in the dynamic cluster configuration or manually add configured server instances to the dynamic cluster. A dynamic cluster that contains both dynamic and configured server instances is called a mixed cluster.

The following table defines terminology associated with dynamic clusters:

Term	Definition
dynamic cluster	A cluster that contains one or more generated (dynamic) server instances that are based on a single shared server template.
configured cluster	A cluster in which you manually configure and add each server instance.

Term	Definition
dynamic server	A server instance that is generated by WebLogic Server when creating a dynamic cluster. Configuration is based on a shared server template.
configured server	A server instance for which you manually configure attributes.
mixed cluster	A cluster that contains both dynamic and configured server instances.
server template	A prototype server definition that contains common, non-default settings and attributes that can be assigned to a set of server instances, which then inherit the template configuration. For dynamic clusters, the server template is used to generate the dynamic servers. See Server Templates in *Understanding Domain Configuration for Oracle WebLogic Server*.

For more information about dynamic clusters, see Dynamic Clusters in *Administering Clusters for Oracle WebLogic Server*.

Why Do You Use Dynamic Clusters?

With dynamic clusters, you can easily scale up your cluster when you need additional server capacity by simply starting one or more of the preconfigured dynamic server instances. You do not need to manually configure a new server instance and add it to the cluster or perform a system restart.

How Do Dynamic Clusters Work?

You set the number of managed server instances that you want, and create or select a template for them.

- Creating and Configuring Dynamic Clusters
  To create a dynamic cluster, you need to complete three steps in the Configuration Wizard.
- Using Server Templates
  Server templates define common configuration attributes that a set of server instances share. Dynamic clusters use server templates for dynamic server configuration.
- Calculating Server-Specific Attributes
  You cannot configure individual dynamic server instances or override server template values at the dynamic server level when using a dynamic cluster. Server-specific attributes, such as server name, machines, and listen ports, must be calculated using the values you set when creating the dynamic cluster.

Creating and Configuring Dynamic Clusters

To create a dynamic cluster, you need to complete three steps in the Configuration Wizard.

- Specify the number of server instances you anticipate needing at peak load.
- Create or select the server template that you want to base server configuration on.
- Define how WebLogic Server should calculate server-specific attributes.

WebLogic Server then generates the number of dynamic server instances you specified and applies the calculated attribute values to each dynamic server instance.

Note:

Ensure you have the performance capacity to handle the maximum number of server instances you specify in the dynamic cluster configuration. For design and deployment best practices when creating a cluster, see Clustering Best Practices.

Using Server Templates

Server templates define common configuration attributes that a set of server instances share. Dynamic clusters use server templates for dynamic server configuration.

See Server Templates in *Understanding Domain Configuration for Oracle WebLogic Server*.

Calculating Server-Specific Attributes

You cannot configure individual dynamic server instances or override server template values at the dynamic server level when using a dynamic cluster. Server-specific attributes, such as server name, machines, and listen ports, must be calculated using the values you set when creating the dynamic cluster.

Note:

You must set a unique Listen Address value for the Managed Server instance that will host the JTA Transaction Recovery service. Otherwise, the migration fails.

WebLogic Server calculates and applies these server-specific attributes using the dynamic server instance ID:

- Server name
- (Optional) Listen ports (clear text and SSL)
- (Optional) Network access point listen ports
- (Optional) Machines or virtual machines

- Calculating Server Names
  The **Server Name Prefix** controls the calculated server name.
- Calculating Listen Ports
  Calculating Listen Ports specifies whether listen ports for the server are calculated.
- Calculating Machine Names
  The **Calculated Machine Names** and **Machine Name Match Expression** control how server instances in a dynamic cluster are assigned to a machine.

Calculating Server Names

The **Server Name Prefix** controls the calculated server name.

Server names are the prefix that you enter, followed by the index number. For example, if the prefix is set to `dyn-server-`, then the dynamic servers have the names `dyn-server-1`, `dyn-server-2`, and so on for the number of server instances you specified.

Calculating Listen Ports

Calculating Listen Ports specifies whether listen ports for the server are calculated.

If you do not calculate listen ports when creating your dynamic cluster, WebLogic Server uses the value in the server template. If you don't define listen ports in the dynamic cluster configuration or server template, WebLogic Server uses the default value.

If you define a base listen port for your dynamic cluster in the server template or the dynamic cluster configuration, the listen port value for the first dynamic server instance is the base listen port incremented by one. For each additional dynamic server instance, the listen port value increments by one. If the default base listen port is used, WebLogic Server increments the *hundreds* digit by one and continues from there for each dynamic server instance.

Calculating Machine Names

The **Calculated Machine Names** and **Machine Name Match Expression** control how server instances in a dynamic cluster are assigned to a machine.

If you select the **Calculated Machine Names** option, you can use the **Machine Name Match Expression** to choose the set of machines used for the dynamic servers. If you don't set **Machine Name Match Expression**, then *all* machines in the domain are selected. Assignments are made using a round robin algorithm.

The following table shows examples of machine assignments in a dynamic cluster.

Table 7-1 Calculating Machine Names

Machines in Domain	Machine Name Match Expression Configuration	Dynamic Server Machine Assignments
M1, M2	Not set	dyn-server-1: M1 dyn-server-2: M2 dyn-server-3: M1 ...
Ma1, Ma2, Mb1, Mb2	Ma1, Mb*	dyn-server-1: Ma1 dyn-server-2: Mb1 dyn-server-3: Mb2 dyn-server-4: Ma1 ...

Creating Dynamic Clusters in a High Availability Topology

You can create dynamic clusters to scale out a high availability topology.

Before you create dynamic clusters, verify the following:

- Oracle Fusion Middleware Infrastructure and the product software are installed.
- Product schemas are created in the database.

To create dynamic clusters for a high availability topology:

1. Go to the `bin` directory.

   On UNIX operating systems:

   ```
   ORACLE_HOME/oracle_common/common/bin
   ```

   On Windows operating systems:

   ```
   ORACLE_HOME\oracle_common\common\bin
   ```

   where *ORACLE_HOME* is your 12*c* (12.2.1.4.0) Oracle home.
2. Launch the Configuration Wizard:

   On UNIX operating systems:

   ```
   ./config.sh
   ```

   On Windows operating systems:

   ```
   config.cmd
   ```
3. On the **Configuration Type** screen, select **Create a new domain**
4. On the **Templates** screen, enter a name for the template you are creating. Select **Next**. Continue with the Configuration Wizards screens to follow the typical steps to create a cluster, until you reach the **Managed Servers** screen.

   If you need information on the steps to create a cluster starting with the **Application Location** screen (the screen after **Templates** in the Configuration Wizard), see your product's installation guide.
5. (Optional) When you reach the **Managed Servers** screen, you can delete *static* Managed Servers if you want to create a domain with dynamic Managed Servers only. To do this, select the static Managed Servers that you don't want in the domain then select **Delete**. (Don't select the Managed Servers that were already listed.) Click **Next**.
6. On the **Clusters** screen, select **Add**. Enter a name for the cluster and values for **Frontend HTTP Port** and **Frontend HTTPS Port**. Select *`product`*`-DYN-CLUSTER` in the **Dynamic Server Groups** drop down menu. Select **Unspecified** if a dynamic server group is not listed.
7. The **Server Templates** screen shows templates available for the domain. In the Cluster drop down menu, select the cluster that each server template belongs to.
8. In the **Dynamic Servers** screen, you customize the cluster you just created. Enter **2** as the **Maximum Dynamic Server Count** value. This is the typical number of servers for your first cluster.
9. Select **Calculated Machine Names** and enter a value for **Machine Name Match Expression** to control *how* server instances in a dynamic cluster are assigned to a machine.
   - You must select **Calculated Machine Names** to assign dynamic servers to a specific machine.
   - To choose the set of machines that dynamic servers use, enter a value for **Machine Name Match Expression**. If you *don't* enter a value for **Machine Name Match**

Expression, *all* machines in the domain are selected and a round robin method assigns machines.
If you enter a name, each specified value matches a machine name exactly. Enter a trailing asterisk (*) suffix to match multiple machine names, for example, `SOAHOST*` or `WCPHOST*`

If the **Machine Name Match Expression** is a tag, each specified value matches all machines that have those tag values.

10. Select **Calculated Listen Ports**.
11. Select **Dynamic Cluster**.
12. Click **Next**.

    The **Assign Servers to Clusters** screen opens. In remaining Configuration Wizard screens, you continue to create a domain as you would create a typical domain. See Assigning Managed Servers to the Cluster for information on this screen.

Expanding or Reducing Dynamic Clusters

When you create a dynamic cluster, WebLogic Server generates the number of dynamic servers you specify. Before you decide upon the number of server instances, ensure you have the performance capacity to handle the desired number.

The dynamic server instances are based on the configuration you specified in the server template and calculated attributes.

- To expand your cluster, start any number of the preconfigured dynamic servers.
- To shrink your dynamic cluster, shut down the excess number of dynamic servers.

If you need additional server capacity on top of the number of server instances you originally specified, increase the maximum number of dynamic servers in the dynamic cluster configuration. To reduce the number of server instances in the dynamic cluster, decrease the value of the maximum number of dynamic servers attribute. Before lowering this value, shut down the server instances you plan to remove.

8 JMS and JTA High Availability

To configure Java Message Service (JMS) and Java Transaction API (JTA) services for high availability, you deploy them to migratable targets that can migrate from one server in a cluster to another server.

- About JMS and JTA Services for High Availability
  Java Message Service (JMS) is an application program interface (API) that supports the formal communication known as *messaging* between computers in a network.
- About Migratable Targets for JMS and JTA Services
  To configure JMS and JTA services for high availability, you deploy them to a **migratable target**, a special target that can migrate from one server in a cluster to another.
- Configuring Migratable Targets for JMS and JTA High Availability
  To configure a migratable target, you specify servers that can host a target; only one server can host a migratable target at any one time. You also set the host you prefer for services and back up servers if the preferred host fails.
- User-Preferred Servers and Candidate Servers
  When you deploy a JMS service to a migratable target, you can select a user-preferred server target to host the service. You can also specify **constrained candidate servers** (CCS) that can host a service if the user-preferred server fails.
- Using File Persistence
  Oracle recommends storing the JMS and JTA data in the database for higher reliability, storage in cloud environments, and better consistency in disaster recovery scenarios instead of using file persistence. For example, use the JDBC Store and TLOG-in-DB features for JMS and JTA respectively. If you choose to use a file system, you must use a shared file system for high availability.
- Using File Stores on NFS
  If you store JMS messages and transaction logs on an NFS-mounted directory, Oracle strongly recommends that you verify server restart behavior after an abrupt machine failure. Depending on the NFS implementation, different issues can arise after a failover/restart.
- Configuring WLS JMS with a Database Persistent Store
  You can change WLS JMS configuration from a file-based persistent store (default configuration) to a database persistent store.
- Configuring Database Stores to Persist Transaction Logs
  After you confirm that your setup has a standard Oracle Fusion Middleware installation, you can configure JDBC Transaction Logs (TLOG) Stores.
- Using the Config Wizard for configuring Automatic Service Migration and JDBC Persistent stores for FMW components
  You can use the HA Options screen in the Configuration Wizard to automate the JDBC store persistence and configure service migration. This screen appears for the first time when you create a Fusion Middleware cluster that may use Automatic Service Migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

About JMS and JTA Services for High Availability

Java Message Service (JMS) is an application program interface (API) that supports the formal communication known as *messaging* between computers in a network.

Java Transaction API (JTA) specifies standard Java interfaces between a transaction manager and parties involved in a distributed transaction system: the resource manager, the application server, and the transactional applications.

In WebLogic JMS, a message is available only if its host JMS server for the destination is running. If a message is in a central persistent store, the only JMS server that can access the message is the server that originally stored the message. WebLogic has features to restart and/or migrate a JMS server automatically after failures. It also has features for clustering (distributing) a destination across multiple JMS servers within the same cluster.

You automatically restart and / or migrate (fail over) JMS Servers using either Whole Server Migration or Automatic Service Migration.

> **Note:**
>
> For more on working with JMS or JTA, see:
>
> - Configuring WebLogic JMS Clustering in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*
> - Interoperating with Oracle AQ JMS in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*
> - Configuring JTA in *Developing JTA Applications for Oracle WebLogic Server*.
> - Configure Domain JTA Options and Configure Cluster JTA Options in *Administration Console Online Help*.

> **Note:**
>
> For more on Whole Server Migration, see Whole Server Migration .

About Migratable Targets for JMS and JTA Services

To configure JMS and JTA services for high availability, you deploy them to a **migratable target**, a special target that can migrate from one server in a cluster to another.

A migratable target groups migratable services that should move together. When a migratable target migrates, all services that it hosts also migrate.

A migratable target specifies a set of servers that can host a target. Only one server can host a migratable target at any one time. It can also specify:

- A user-preferred host for services
- An ordered list of backup servers if a preferred server fails

After you configure a service to use a migratable target, it is independent from the server member that currently hosts it. For example, if you configure a JMS server with a deployed JMS queue to use a migratable target, the queue is independent of when a specific server member is available. The queue is always available when any server in the cluster hosts the migratable target.

You can migrate pinned migratable services manually from one server to another in the cluster if 1) a server fails, or 2) as part of scheduled maintenance. If you *do not* configure a migratable target in the cluster, migratable services can migrate to any server in the cluster.

See Configuring Migratable Targets for JMS and JTA High Availability to configure migratable targets.

> **Note:**
>
> For more on administering JMS, see the following topics in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*:
>
> - High Availability Best Practices
> - Interoperating with Oracle AQ JMS

Configuring Migratable Targets for JMS and JTA High Availability

To configure a migratable target, you specify servers that can host a target; only one server can host a migratable target at any one time. You also set the host you prefer for services and back up servers if the preferred host fails.

To configure migratable targets, see these topics in Administration Console Online Help:

- Configure Migratable Targets for JMS-Related Services
- Configure Migratable Targets for JTA Transaction Recovery Service

User-Preferred Servers and Candidate Servers

When you deploy a JMS service to a migratable target, you can select a user-preferred server target to host the service. You can also specify **constrained candidate servers** (CCS) that can host a service if the user-preferred server fails.

If a migratable target doesn't specify a CCS, you can migrate the JMS server to any available server in the cluster.

You can create separate migratable targets for JMS services so that you can always keep each service running on a different server in the cluster, if necessary. Conversely, you can configure the same selection of servers as the CCSs for both JTA and JMS, to ensure that services stay co-located on the same server in the cluster.

Using File Persistence

Oracle recommends storing the JMS and JTA data in the database for higher reliability, storage in cloud environments, and better consistency in disaster recovery scenarios instead of using file persistence. For example, use the JDBC Store and TLOG-in-DB features for JMS and JTA respectively. If you choose to use a file system, you must use a shared file system for high availability.

WebLogic supplies multiple types of file stores which have multiple purposes as follows:

- Each WebLogic Server has a default file store. However, default file stores should not be used for storing critical data such as JMS messages, JTA transactions, and EJB timers. You should use JDBC stores, TLOG-in-DB, and database stored timers instead. An example of non-critical data that is stored in a default store is application life-cycle state, such as whether a particular application has been administratively paused. If there is no critical data in a default file store, it is safe to delete such stores in the event of a catastrophic corruption as this mitigates the risk of disabling file locking for the default file store.
- WebLogic JMS paging stores are active if JMS has a large message backlog. The data in JMS paging files is not reloaded when the server is not running, and so the files can be safely deleted when a WebLogic Server is not running. The corresponding persistent messages are simultaneously stored in default file stores, custom file stores, or custom JDBC stores.
- WebLogic diagnostic stores contain non-critical diagnostic data. They are run within a buffering mode that allows for very high performance in order to minimize the overhead of diagnostics, but this increases the risk of corruption after a failure. If such files become corrupt, then it is safe for WebLogic Servers to reboot, and it is also safe to delete the files.

See Tuning the WebLogic Persistent Store section in *Tuning Performance of Oracle WebLogic Server*.

Using File Stores on NFS

If you store JMS messages and transaction logs on an NFS-mounted directory, Oracle strongly recommends that you verify server restart behavior after an abrupt machine failure. Depending on the NFS implementation, different issues can arise after a failover/restart.

- Verifying Server Restart Behavior
  To verify server restart behavior, abruptly shut down the node that hosts WebLogic servers while the servers are running.
- Prerequisites for Disabling File Locking
- Disabling File Locking for all Stores Using a System Property
- Disabling File Locking for the Default File Store
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on file stores, you can disable file locking.

- Disabling File Locking for a Custom File Store
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on custom file stores, you can disable file locking.
- Disabling File Locking for a JMS Paging File Store
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on JMS paging file stores, you can disable file locking.
- Disabling File Locking for Diagnostics File Stores
  If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on diagnostics paging file stores, you can disable file locking.

Verifying Server Restart Behavior

To verify server restart behavior, abruptly shut down the node that hosts WebLogic servers while the servers are running.

- If you *configured the server for server migration,* it should start automatically in failover mode after the failover period.
- If you *did not* configure the server for server migration, you can manually restart the WebLogic Server on the same host after the node completely reboots.

If WebLogic Server doesn't restart after abrupt machine failure, review server log files to verify whether or not it is due to an I/O exception similar to the following:

```
<MMM dd, yyyy hh:mm:ss a z> <Error> <Store> <BEA-280061> <The persistent
store "_WLS_server_1" could not be deployed:
weblogic.store.PersistentStoreException: java.io.IOException:
[Store:280021]There was an error while opening the file store file
"_WLS_SERVER_1000000.DAT"
weblogic.store.PersistentStoreException: java.io.IOException:
[Store:280021]There was an error while opening the file store file
"_WLS_SERVER_1000000.DAT"
at weblogic.store.io.file.Heap.open(Heap.java:168)
at weblogic.store.io.file.FileStoreIO.open(FileStoreIO.java:88)
...
java.io.IOException: Error from fcntl() for file locking, Resource
temporarily unavailable, errno=11
```

This error occurs when the NFSv3 system doesn't release locks on file stores. WebLogic Server maintains locks on files that store JMS data and transaction logs to prevent data corruption that can occur if you accidentally start two instances of the same Managed Server. Because the NFSv3 storage device doesn't track lock owners, NFS holds the lock indefinitely if a lock owner fails. As a result, after abrupt machine failure followed by a restart, subsequent attempts by WebLogic Server to acquire locks may fail.

How you resolve this error depends on your NFS environment: (See *Oracle Fusion Middleware Release Notes* for updates on this topic.)

- **For NFSv4 environments**, you can set a tuning parameter on the NAS server to release locks within the approximate time required to complete server migration; you don't need to follow procedures in this section. See your storage vendor's documentation for information on locking files stored in NFS-mounted directories on the storage device, and test the results.

- **For NFSv3 environments**, the following sections describe how to disable WebLogic file locking mechanisms for: the default file store, a custom file store, a JMS paging file store, a diagnostics file store.

> **WARNING:**
>
> NFSv3 file locking prevents severe file corruptions that occur if more than one Managed Server writes to the same file store at any point in time.
>
> If you disable NFSv3 file locking, you must implement administrative procedures /policies to ensure that only one Managed Server writes to a specific file store. Corruption can occur with two Managed Servers in the same cluster or different clusters, on the same node or different nodes, or on the same domain or different domains.
>
> Your policies could include: never copy a domain, never force a unique naming scheme of WLS-configured objects (servers, stores), each domain must have its own storage directory, no two domains can have a store with the same name that references the same directory.
>
> When you use a file store, always configure the database-based leasing option for server migration. This option enforces additional locking mechanisms using database tables and prevents automated restart of more than one instance of a particular Managed Server.

Prerequisites for Disabling File Locking

All file stores are locked by default. The WebLogic file locking feature is designed to help prevent severe file corruptions that can occur in concurrency scenarios. Perform the following prerequisite tasks to mitigate the risk of disabling file locks:

- If the server using the file store is configured for server migration, always configure the database-based cluster leasing option instead of the default consensus leasing. This enforces additional locking mechanisms using database tables and prevents automated concurrent restart of more than one instance of a particular WebLogic Server.
- Avoid disabling file locks on a file store that is using Automatic Service Migration (ASM).
  - ASM requires file store locking to work safely and is activated in the following scenarios:
    1. A custom file store target is set to a Migratable Target and the Migratable Target is configured with a Migration Policy other than 'manual' (the default).
    2. A custom file store target is set to a WebLogic cluster when the store is configured with a Migration Policy other than 'Off' (the default).
    3. A WebLogic Server is configured with a JTA Migratable Target with a Migration Policy value other than ‘manual’ (the default), as this in turn can lead to default file store migrations.
  - If both ASM and disabling file locks are required, store your critical data in database stores instead of file stores to avoid the risk of file corruptions. For

example, use a custom JDBC store instead of a file store and configure JTA to use a JDBC TLOG store instead of each WebLogic Server’s default file store.

- Additional procedural precautions must be implemented to avoid any human error and to ensure that only one instance of a server is manually started at any given point in time. Similarly, precautions must be taken to ensure that no two domains have a store with the same name that references the same directory.

Disabling File Locking for all Stores Using a System Property

Starting from WebLogic Server 14.1.2 release, you can specify a Java system property on the `weblogic.Server` command line or start script of the JVM to disable locking on all of its file stores including default, paging, diagnostic, and custom file stores as shown below:

"`-Dweblogic.store.file.LockEnabled=false`"

Disabling File Locking for the Default File Store

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on file stores, you can disable file locking.

To disable file locking for the default file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center.
2. In the **Domain Structure** tree, expand the **Environment** node and select **Servers**.
3. In the **Summary of Servers** list, select the server you want to modify.
4. Select the **Configuration > Services** tab.
5. Scroll down to the **Default Store** section and click **Advanced**.
6. Scroll down and deselect the **Enable File Locking** check box.
7. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
8. **Restart** the server you modified for the changes to take effect.

The resulting `config.xml` entry looks like the following:

```
<server>
 <name>examplesServer</name>
 ...
 <default-file-store>
 <synchronous-write-policy>Direct-Write</synchronous-write-policy>
 <io-buffer-size>-1</io-buffer-size>
 <max-file-size>1342177280</max-file-size>
 <block-size>-1</block-size>
 <initial-size>0</initial-size>
 <file-locking-enabled>false</file-locking-enabled>
 </default-file-store>
 </server>
```

Disabling File Locking for a Custom File Store

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on custom file stores, you can disable file locking.

To disable file locking for a custom file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center to get an Edit lock for the domain.
2. In the **Domain Structure** tree, expand the **Services** node. Select **Persistent Stores**.
3. In the **Summary of Persistent Stores** list, select the custom file store you want to modify.
4. On the **Configuration** tab for the custom file store, click **Advanced**.
5. Scroll down and deselect the **Enable File Locking** check box.
6. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
7. If the custom file store was in use, you must restart the server for changes to take effect.

The `config.xml` entry looks like the following example:

```
<file-store>
 <name>CustomFileStore-0</name>
 <directory>C:\custom-file-store</directory>
 <synchronous-write-policy>Direct-Write</synchronous-write-policy>
 <io-buffer-size>-1</io-buffer-size>
 <max-file-size>1342177280</max-file-size>
 <block-size>-1</block-size>
 <initial-size>0</initial-size>
 <file-locking-enabled>false</file-locking-enabled>
 <target>examplesServer</target>
 </file-store>
```

Disabling File Locking for a JMS Paging File Store

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on JMS paging file stores, you can disable file locking.

To disable file locking for a JMS paging file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center to get an Edit lock for the domain.
2. In the **Domain Structure** tree, expand the **Services** node, expand the **Messaging** node, and select **JMS Servers**.
3. In the **Summary of JMS Servers** list, select a JMS server to modify.
4. On the **Configuration > General** tab for the JMS Server, scroll down. Deselect the **Paging File Locking Enabled** check box.
5. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
6. **Restart** the server you modified for changes to take effect.

The `config.xml` file entry looks like the following example:

```
 <jms-server>
 <name>examplesJMSServer</name>
 <target>examplesServer</target>
 <persistent-store>exampleJDBCStore</persistent-store>
 ...
 <paging-file-locking-enabled>false</paging-file-locking-enabled>
```

```
...
</jms-server>
```

Disabling File Locking for Diagnostics File Stores

If WebLogic Server doesn't restart after abrupt machine failure and server log files show the NFS system doesn't release locks on diagnostics paging file stores, you can disable file locking.

To disable file locking for a Diagnostics file store using the Administration Console:

1. If necessary, click **Lock & Edit** in the Change Center to get an Edit lock for the domain.
2. In the **Domain Structure** tree, expand the **Diagnostics** node. Select **Archives**.
3. In the **Summary of Diagnostic Archives** list, select the server name of the archive that you want to modify.
4. On the **Settings for [server_name]** page, deselect the **Diagnostic Store File Locking Enabled** check box.
5. Click **Save**. If necessary, click **Activate Changes** in the Change Center.
6. **Restart** the server you modified for the changes to take effect.

The resulting `config.xml` file looks like this:

```
<server>
<name>examplesServer</name>
...
<server-diagnostic-config>
<diagnostic-store-dir>data/store/diagnostics</diagnostic-store-dir>
<diagnostic-store-file-locking-enabled>false</diagnostic-store-file-locking-enabled>
<diagnostic-data-archive-type>FileStoreArchive</diagnostic-data-archive-type>
<data-retirement-enabled>true</data-retirement-enabled>
<preferred-store-size-limit>100</preferred-store-size-limit>
<store-size-check-period>1</store-size-check-period>
</server-diagnostic-config>
</server>
```

> **Note:**
>
> See Configure JMS Servers and Persistent Stores in *Oracle Fusion Middleware Administering JMS Resources for Oracle WebLogic Server*.

Configuring WLS JMS with a Database Persistent Store

You can change WLS JMS configuration from a file-based persistent store (default configuration) to a database persistent store.

- About the Persistent Store
  The **persistent store** is a built-in storage solution for WebLogic Server subsystems and services that require persistence. For example, it can store persistent JMS messages.

About the Persistent Store

The **persistent store** is a built-in storage solution for WebLogic Server subsystems and services that require persistence. For example, it can store persistent JMS messages.

The persistent store supports persistence to a file-based store or to a JDBC-accessible store in a database. For information on the persistent store, see The WebLogic Persistent Store in *Administering the WebLogic Server Persistent Store*.

For information on typical tasks to monitor, control, and configure WebLogic messaging components, see WebLogic Server Messaging in *Administering Oracle WebLogic Server with Fusion Middleware Control*.

Prerequisites for Configuring WLS JMS with a Database Persistent Store

To configure WLS JMS with database persistent stores, verify that your setup meets specific requirements.

Your setup must meet these requirements:

- An Oracle Fusion Middleware installation with at least one cluster and one or more JMS servers
- JMS servers that use file persistent stores, the default configuration.

Switching WLS JMS File-Based Persistent Stores to Database Persistent Store

You can swap JMS servers from file-based to database persistent stores.

You must follow steps in this procedure for each JMS server that you must configure to use database persistent stores.

1. Create a JDBC store. See Using a JDBC Store in *Oracle Fusion Middleware Administering Server Environments for Oracle WebLogic Server*.

   > **Note:**
   >
   > You must specify a prefix to uniquely name the database table for the JDBC store.

2. Associate the JDBC store with the JMS server:
   a. In the Weblogic Server Administration Console, go to **Services**->**Messaging**->**JMS Servers**.

b. Verify that there are no pending messages in this server. In the Control tab, stop production and insertion of messages for all destinations and wait for any remaining messages to drain.

c. Select the General Configuration tab. Under Persistent Store, select the new JDBC store then click **Save**.

The JMS server starts using the database persistent store.

Configuring Database Stores to Persist Transaction Logs

After you confirm that your setup has a standard Oracle Fusion Middleware installation, you can configure JDBC Transaction Logs (TLOG) Stores.

- Requirements for Configuring JDBC TLOG Stores
  You must have a standard Oracle Fusion Middleware installation before you configure a JDBC Transaction Logs (TLOG) Store.
- Configuring JDBC TLOG Stores
  There are a few guidelines to follow when you configure JDBC TLOG Stores for Managed Servers in a cluster.

Requirements for Configuring JDBC TLOG Stores

You must have a standard Oracle Fusion Middleware installation before you configure a JDBC Transaction Logs (TLOG) Store.

Post installation, the TLOG Store is configured in the file system. In some instances, Oracle recommends that you configure TLOGs to store in the database. To configure JDBC TLOGs to stored to a database store, see Using a JDBC TLOG Store in *Administering the WebLogic Server Persistent Store*.

Configuring JDBC TLOG Stores

There are a few guidelines to follow when you configure JDBC TLOG Stores for Managed Servers in a cluster.

When you configure JDBC TLOG Stores:

- You must repeat the procedure for each Managed Server in the cluster.
- Use the Managed Server name as a prefix to create a unique TLOG store name for each Managed Server.
- Verify that the data source that you created for the persistent store targets the cluster for a high availability setup.

When you finish the configuration, TLOGs are directed to the configured database-based persistent store.

> **Note:**
>
> When you add a new Managed Server to a cluster by scaling up or scaling out, you must also create the corresponding JDBC TLOG Store for the new Managed Server.

Using the Config Wizard for configuring Automatic Service Migration and JDBC Persistent stores for FMW components

You can use the HA Options screen in the Configuration Wizard to automate the JDBC store persistence and configure service migration. This screen appears for the first time when you create a Fusion Middleware cluster that may use Automatic Service Migration, persistent stores, or both, and all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply the selected HA options.

If you select the **Enable Automatic Service Migration** option, it configures migrtable target definitions that are required for automatic service migration. You can either select database or consensus leasing option. If you select **Database Leasing**, the leasing datasource is also configured.

In the same screen, use the options **JTA Transaction Log Persistence** and **JMS Server Persistence** to configure JDBC stores automatically. Fusion Middleware component templates automatically define the JDBC persistent stores for JMS Servers and Transaction Logs.

For information about what to select in the High Availability options screen during the domain creation process, see Configuring High Availability Options .

9

Administration Server High Availability

The Administration Server plays a unique role in domains. To set up high availability, you configure the Administration Server on a virtual host.

- Administration Server Role
  The Administration Server is the central control entity for configuring the entire domain, and manage and monitor all domain resources. It maintains domain configuration documents and distributes changes in them to Managed Servers.
- Role of Node Manager
  For each WebLogic Server domain you create, a *per domain* Node Manager configuration is created by default. This Node Manager comes complete with security credentials, a properties file, domain registration, and start scripts.
- Administration Server High Availability Topology
  An Administration Server can be active on one host at any one time. To set up high availability, you configure the Administration Server on a virtual host so that if the machine that it runs on fails, you can fail it over to another host in the domain.
- Configuring Administration Server High Availability
  In a highly available Administration Server environment, both hosts must have access to shared storage because the domain directory is maintained in shared storage.
- Failing Over or Failing Back Administration Server
  You fail over or fail back the Administration Server after host failure.

Administration Server Role

The Administration Server is the central control entity for configuring the entire domain, and manage and monitor all domain resources. It maintains domain configuration documents and distributes changes in them to Managed Servers.

Each domain requires one server that acts as the Administration Server. For more information on the Administration Server and Node Manager, see the following topics:

Table 9-1 Administration Server and Node Manager Topics

For information on...	See this topic...
Starting and Stopping the Administration Server	Starting and Stopping Administration Server in *Administering Oracle Fusion Middleware*
Configuring virtual hosting	Configuring Virtual Hosting in *Administering Server Environments for Oracle WebLogic Server*
Using Node Manager	In *Administering Node Manager for Oracle WebLogic Server*: • Node Manager and System Crash Recovery • How Node Manager Works in a WLS Environment • Node Manager Configuration and Log Files

- Administration Server Failure and Restart
  Administration Server failure doesn't affect how Managed Servers in a domain operate.

- Shared Storage and Administration Server High Availability
  With shared storage, a backup host can access the same artifacts (Oracle binaries, configuration files, domain directory, and data) that an active host can.

Administration Server Failure and Restart

Administration Server failure doesn't affect how Managed Servers in a domain operate.

If an Administration Server fails due to a hardware or software failure on its host computer, other server instances on the same computer may also be affected.

For more information on Administration Server failure, see Impact of Managed Server Failure in *Administering Oracle Fusion Middleware*.

To restart an Administration Server, see What Happens if the Administration Server Fails? in *Oracle Fusion Middleware Using Clusters for Oracle Server*.

Shared Storage and Administration Server High Availability

With shared storage, a backup host can access the same artifacts (Oracle binaries, configuration files, domain directory, and data) that an active host can.

You configure this access by placing artifacts in storage that all hosts in the highly available Administration Server configuration can access. Shared storage is a network-attached storage (NAS) or storage area network (SAN) device. See Using Shared Storage.

Role of Node Manager

For each WebLogic Server domain you create, a *per domain* Node Manager configuration is created by default. This Node Manager comes complete with security credentials, a properties file, domain registration, and start scripts.

You can configure the scope of Node Manager:

- **per domain** Node Manager is associated with a *domain* to control all servers for the domain on a machine. Default configuration.
- **per host** Node Manager is associated with a specific *machine*, not a domain. One Node Manager process can control server instances in any domain, as long as the server instances reside on the same machine as the Node Manager process. A per host Node Manager must run on each computer that hosts WebLogic Server instances that you want to control with Node Manager, whether the WebLogic Server instances are an Administration Server or Managed Server(s).

> **Note:**
>
> Oracle recommends that you run Node Manager as an operating system service so that it restarts automatically if its host machine restarts.

Node Manager failure doesn't affect any servers running on the machine.

See What is Node Manager? in *Oracle Fusion Middleware Understanding Oracle Fusion Middleware Concepts*.

Administration Server High Availability Topology

An Administration Server can be active on one host at any one time. To set up high availability, you configure the Administration Server on a virtual host so that if the machine that it runs on fails, you can fail it over to another host in the domain.

Administration Server is configured to use a virtual IP to overlap the backup hosts. You configure Administration Server to listen on this virtual IP. The benefit of a virtual host and virtual IP is that you don't need to add a third machine; if failover occurs, you can map the virtual host to a surviving host in the domain by *moving* the virtual IP.

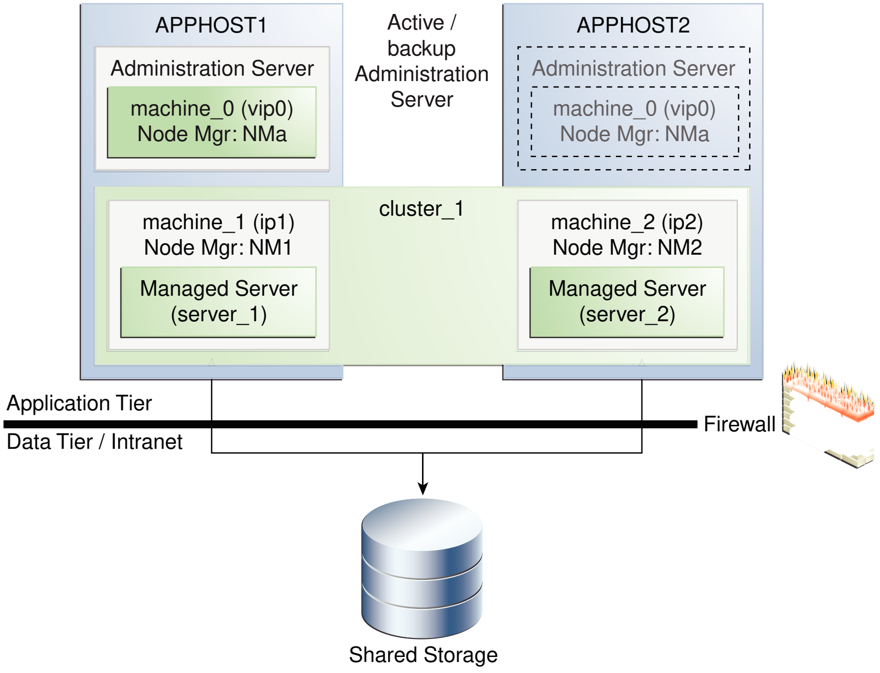
The two hosts share a virtual hostname and a virtual IP. However, only the active host can use this virtual IP at any one time. If the active host fails, the backup host becomes active and you must move (manually) the virtual IP to the new active host. The new active host then services all requests through the virtual IP. (You configure a high availability deployment to listen on this virtual IP.)

Figure 9-1 shows a highly available Administration Server.

In this topology, the Administration Server runs on a virtual host, `APPHOST0`. `APPHOST0` overlaps to `APPHOST1` or `APPHOST2` by means of a virtual IP.

At first, `APPHOST0` maps to `APPHOST1`. However, if the Administration Server fails due to an `APPHOST1` failure, `APPHOST0` fails over to `APPHOST2` by moving the virtual IP.

Figure 9-1 Administration Server High Availability Topology



Configuring Administration Server High Availability

In a highly available Administration Server environment, both hosts must have access to shared storage because the domain directory is maintained in shared storage.

During normal operation, the Administration Server on the active host owns the domain directory in shared storage. If the active host fails, the backup host takes over and restarts the Administration Server from the shared domain directory.

- Administration Server High Availability Requirements
  To configure a highly available Administration Server, your environment must meet certain requirements.
- Configuring the Administration Server for High Availability
  To configure the Administration Server for high availability, you must start with the standard high availability topology that has one cluster (`cluster_1`).

Administration Server High Availability Requirements

To configure a highly available Administration Server, your environment must meet certain requirements.

- Conform to the standard installation topology. See About the Oracle Fusion Middleware Standard HA Topology and Figure 1-1.
- Include two hosts, APPHOST1 and APPHOST2, to implement a WebLogic Server cluster (`cluster_1`). In Configuring the Administration Server for High Availability, IP addresses for APPHOST1 and APPHOST2 are `ip1` and `ip2`, respectively.
- APPHOST1 and APPHOST2 mount a common directory from shared storage and have read/write access rights to the directory. This directory is for installing products and storing domain directories.
- A reserved virtual IP address (`vip0`) to *point* to the host that runs the Administration Server. This floating virtual server IP address is configured dynamically on the host that the Administration Server runs on.
- Node Manager instances to manage the Administration Server and migrate it from the failed host to the designated standby host.

Configuring the Administration Server for High Availability

To configure the Administration Server for high availability, you must start with the standard high availability topology that has one cluster (`cluster_1`).

See About the Oracle Fusion Middleware Standard HA Topology.

To set up a highly available Administration Server, you run the Administration Server on a separate, virtual host (`APPHOST0`). You set up `APPHOST0` so that it maps to one of the existing hosts in the cluster (`APPHOST1` or `APPHOST2`) by configuring a (virtual) server IP for `APPHOST0` on that existing host. If failover occurs, `APPHOST0` fails over by moving its virtual server IP to a surviving host. The domain configuration is on shared storage so that the surviving host can access it.

Note:

There are multiple ways for Administration Server services to accomplish configuration tasks. No matter which method you use, the Administration Server must be running when you change the configuration.

Table 9-2 Host and Node Manager Terms

Term	Description
`APPHOST0`, `machine_0`	Virtual machine that the Administration Server runs on
`APPHOST1`, `APPHOST2`	Machines that host the application tier.
`vip0`	Virtual server IP address that the Administration Server listens on
`NMa`	Per-domain Node Manager that manages the Administration Server that runs on APPHOST0
`NM1`, `NM2`	Node Manager instances that run on APPHOST1 and APPHOST2, respectively
`ip1`, `ip2`	IP addresses of APPHOST1 and APPHOST2, respectively

To configure the Administration Server for high availability:

1. Configure a virtual server IP address (`vip0`) on `APPHOST1` to represent virtual host `APPHOST0`.

   See Configuring Virtual Hosting in *Administering Server Environments for Oracle WebLogic Server*.

2. Use an Oracle Fusion Middleware expanded installation procedure to install Oracle Fusion Middleware binaries and configure the domain into a directory on shared storage. Use `vip0` as the Administration Server listen address.

3. Create a virtual machine, `machine_0` and add the Administration Server to it. `machine_0` represents the virtual server host `APPHOST0` with the IP address `vip0`.

4. Create a cluster (`cluster_1`) that has two Managed Servers, `server_1` and `server_2`, that are assigned to `machine_1` and `machine_2`, respectively.

   - `machine_1` represents `APPHOST1` and `machine_2` represents `APPHOST2`.
   - `server_1` and `server_2` are set up to listen on `ip1` and `ip2`, respectively.

5. Scale out the virtual server to `APPHOST1` and `APPHOST2`. See Roadmap for Scaling Out Your Topology. To scale out, you pack the domain in shared storage and unpack it to a local directory on `APPHOST1` and `APPHOST2`.

6. From `APPHOST1`, start a per-domain Node Manager (`NMa`) to manage the Administration Server listening on the configured virtual server IP `vip0` on `APPHOST1`. Start this instance of Node Manager from the domain directory in shared storage.

7. On `APPHOST1`, start a per-domain Node Manager (`NM1`) to manage `server_1` listening on `ip1`. Start this Node Manager (`NM1`) from the domain directory that you unpacked to local storage in `APPHOST1` in step 5.

8. On APPHOST2, start a per-domain Node Manager (NM2) to manage server_2 listening on ip2. Start this Node Manager (NM2) from the domain directory that you unpacked to local storage in APPHOST2 step 5.
9. Use Node Manager (NMa) to start the Administration Server on APPHOST1.
10. Start Managed Servers server_1 and server_2 using NM1 and NM2, respectively.
11. Verify that the Administration Server and Managed Servers work properly. Connect to the Administration Server using the virtual IP address vip0.

Failing Over or Failing Back Administration Server

You fail over or fail back the Administration Server after host failure.

- Failing Over the Administration Server if Original Host Fails
  You fail over the Administration Server to another host if its original host (APPHOST1) fails. To do this, you configure the virtual IP address on the alternate host (APPHOST2), then start Node Manager and the Administration Server.
- Failing Back the Administration Server to the Original Host
  You fail back the Administration Server to its original host after it restarts.

Failing Over the Administration Server if Original Host Fails

You fail over the Administration Server to another host if its original host (APPHOST1) fails. To do this, you configure the virtual IP address on the alternate host (APPHOST2), then start Node Manager and the Administration Server.

To fail over the Administration Server to APPHOST2 if APPHOST1 fails:

1. Configure vip0 on APPHOST2.
2. Start Node Manager NMa on APPHOST2 from the domain directory in shared storage.
3. Start the Administration Server on APPHOST2 using NMa.
4. Start the Administration Console to verify that the Administration Server is running.

Failing Back the Administration Server to the Original Host

You fail back the Administration Server to its original host after it restarts.

To fail back the Administration Server to APPHOST1 when APPHOST1 comes back online:

1. Stop the Administration Server on APPHOST2 using Node Manager NMa.
2. Remove vip0 from APPHOST2.
3. Stop Node Manager NMa on APPHOST2.
4. Configure vip0 on APPHOST1.
5. Start Node Manager NMa on APPHOST1 using the domain directory in shared storage.
6. Use Node Manager NMa to start the Administration Server on APPHOST1.
7. Start the Administration Console to verify that the Administration Server is running.

Part III

Component Procedures

This topic provides links to procedures that are unique to certain component products.

Topics:

- Configuring High Availability for Oracle Identity Governance Components
- Configuring High Availability for Oracle Access Manager Components
- Configuring High Availability for Oracle Directory Services Components
- Configuring High Availability for Web Tier Components
- Configuring High Availability for SOA Components. See Administering Oracle SOA Suite and Oracle Business Process Management Suite.
- Configuring High Availability for Oracle WebCenter Components:
  - Administering Oracle WebCenter Content
  - Administering Oracle WebCenter Portal
  - Administering Oracle WebCenter Enterprise Capture
  - Administering Oracle WebCenter Sites
- Configuring High Availability for Other Components:
  - Deploying Oracle Data Integrator. See High Availability for Oracle Data Integrator.
  - Deploying Oracle Application Development Framework. See Administering Oracle ADF Applications.
  - Deploying BI. See System Administrator's Guide for Oracle Business Intelligence Enterprise Edition.
  - Deploying Forms. See Working With Oracle Forms.
  - Deploying Reports. See Oracle Reports User's Guide to Building Reports.
  - Deploying Oracle Business Process Management. See Managing and Monitoring Processes with Oracle Business Process Management.