

Managing Clusters

At the end of this module, you will be able to:

- ✓ Deploy applications to a cluster
- ✓ Manage session state in a cluster
- ✓ Troubleshoot a cluster

1. Deploying Applications to Clusters

- Packaging applications
- Two-phase and side-by-side deployment
- Deployment best practices

2. Session Management

3. Troubleshooting a Cluster

Packaging Applications



- ▶ When deploying applications to a single managed server, they can be deployed in exploded format.
- ▶ When deploying applications to a cluster of managed servers, they must be packaged into a .war, .ear or .jar.

Two-phase Deployment



- ▶ Applications are deployed using two-phase deployment (TPD).
- ▶ Applications are copied to the cluster and activated in two phases:
 - Phase 1: distributes application components and modules to the server
 - Phase 2: deploys the application if phase 1 is successful and permits client access
- ▶ This ensures that an application is available and active on each node before clients can access them

Deploying Applications to a Cluster

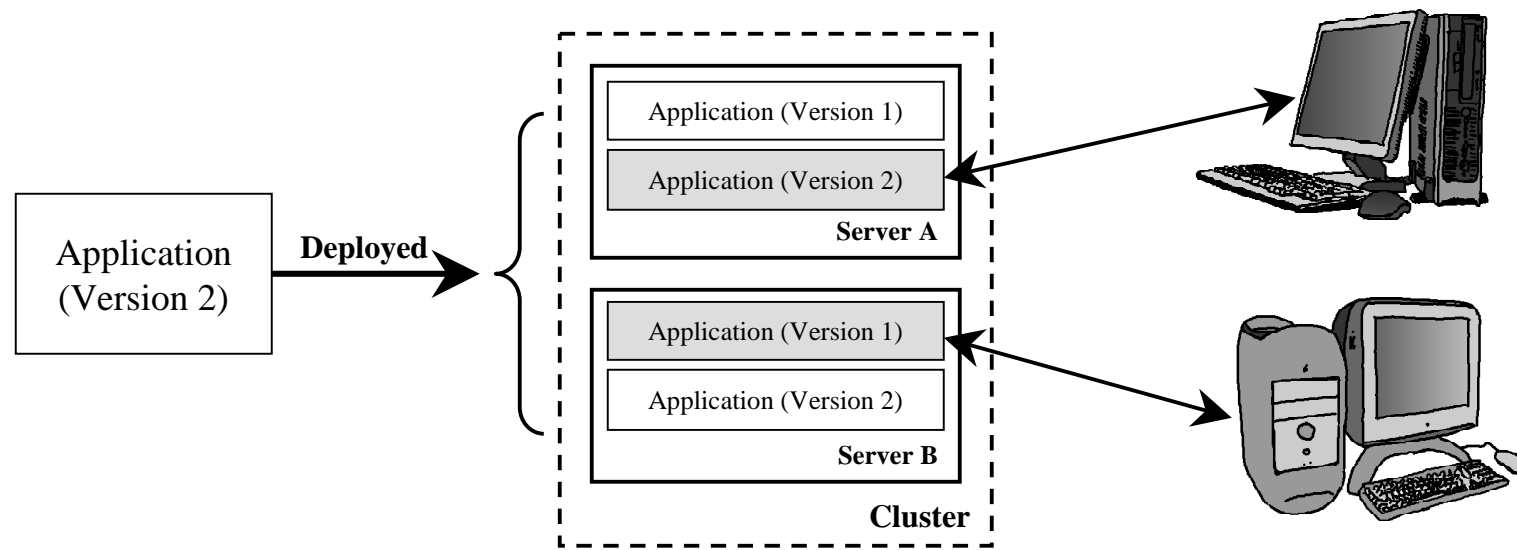


- ▶ All nodes must be running before an application can be deployed to a cluster.
- ▶ If phase 2 of two-phase deployment fails, the application will still be deployed to other nodes in the cluster.
- ▶ WebLogic allows partial deployment of applications to a partitioned server.
- ▶ Session replication for deployed applications may fail if a node was partitioned at the time of deployment.
 - Avoid this by using the `enforceClusterConstraints` tag with `weblogic.Deployer`
 - Or check the Enable Cluster Constraints box in the console
- ▶ Do not change cluster membership while deploying applications to the cluster.

Side-by-Side Redeployment in a Cluster



- ▶ When using production redeployment of an application in a cluster, each server instance in the cluster will retire the old version when work is complete on that server.
 - Therefore, different servers may be running different versions for a period of time



Deploying Applications to a Cluster...



Domain Structure

- onlinestore
 - Environment
 - Deployments** 1
 - Services
 - Security Realms
 - Interoperability
 - Diagnostics

Summary of Deployments

Control **Monitoring**

This page displays a list of J2EE Applications and standalone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Deployments

2

Showing 1 - 1 of 1 Previous | Next

<input type="checkbox"/>	Name ^	State	Type	Deployment Order
<input type="checkbox"/>	browsestore	Active	Web Application	100
<input type="checkbox"/>	proxyApp	Prepared	Web Application	100
<input type="checkbox"/>	shoppingcart	Active	Web Application	100

Showing 1 - 1 of 1 Previous | Next

...Deploying Applications to a Cluster



Settings for browstore 3

Overview Configuration Security **Targets** Control Testing Monitoring Notes

Use this page to select the WebLogic Server instances and clusters, and virtual hosts on which you would like to deploy this Web application.

Servers

☐ adminserver

☐ proxyserver

Clusters

☒ storecluster

☒ All servers in the cluster

☐ Part of the cluster

☐ nodea

☐ nodec

☐ nodeb

4

Section Review



In this section, we learned how to:

- ✓ Package applications for deployment in a cluster
- ✓ Deploy applications in a cluster



1. Deploying Applications to Clusters

2. Session Management

- HTTP session clustering
- Replication groups
- In-memory replication
- Persistent replication
- Replication within a MAN or WAN

3. Troubleshooting a Cluster

HTTP Session State Replication



- ▶ WebLogic Server provides clustering support for JSPs and Servlets by replicating HTTP session state.
- ▶ To benefit from HTTP session state clustering, you must ensure that the session state is persistent, by configuring:
 - In-memory replication
 - JDBC replication
 - File system replication
- ▶ You must also access the cluster via a collection of Web servers with identically configured proxy plug-ins or load balancing hardware.
- ▶ Session persistence is configured using the `<session-descriptor>` element in the `weblogic.xml` deployment descriptor file.
 - Each persistence method has its own set of configurable parameters.

- ▶ In WebLogic server, machines names are used to indicate that a managed server runs on a particular piece of hardware.
- ▶ Machine definitions are one of the factors WebLogic takes into account when it chooses another server as its backup for session information.

Replication Groups



- ▶ A *replication group* is a logical grouping of related servers in a cluster.
- ▶ WLS lets you determine where to put backup objects, using replication groups.
- ▶ WLS attempts to:
 - Send backup objects to a *preferred secondary replication group*, if it is configured
 - Send backup objects to a different machine
 - *Avoid* sending backup objects to servers in the *same* replication group

Configuring Replication Groups



- Select each server in a cluster, and assign each a pair of replication groups.

Settings for ManagedServerA

Configuration Protocols Logging Debug Monitoring Control Deployments S

General **Cluster** Services Keystores SSL Federation Services Deplo

Save

This page allows you to define the cluster configuration for this server. A Web

Replication Group:	RepGroupA
Preferred Secondary Group:	RepGroupB
Cluster Weight:	100
Interface Address:	

Ranking Servers



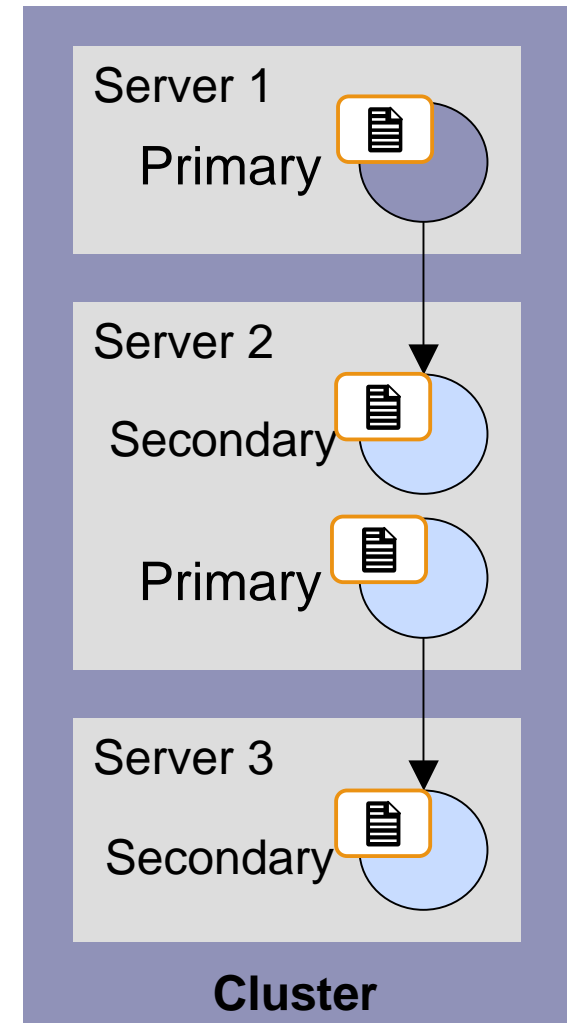
- WebLogic Server ranks the servers in a cluster, to determine which server to use as the backup.

Is the Server a Member of a Preferred Secondary Group?	Does the Server Reside on a Different Machine?	Server Rank
Yes	Yes	1
Yes	No	2
No	Yes	3
No	No	4

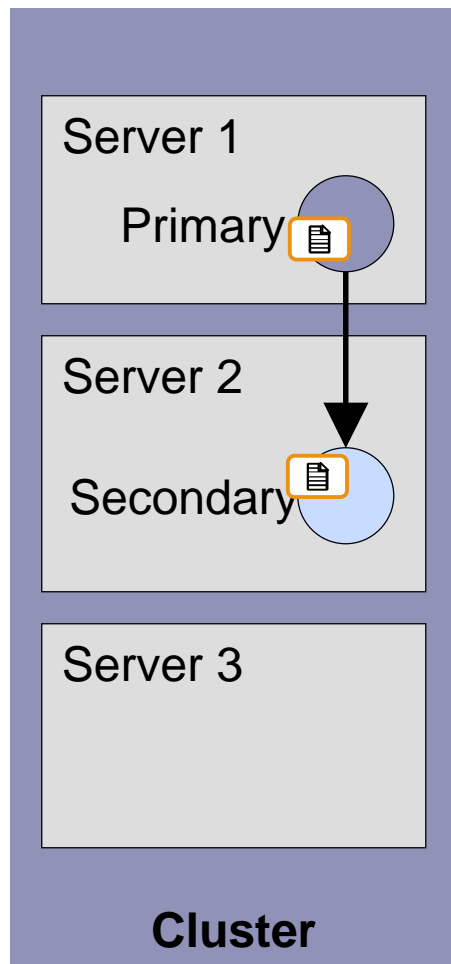
In-Memory Replication



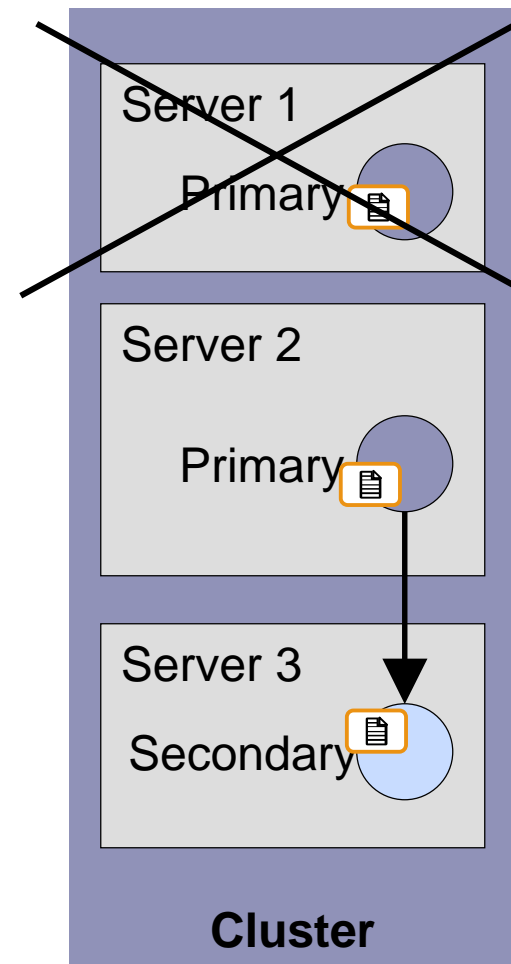
- ▶ WLS can replicate:
 - HttpSession objects
 - Stateful session EJBs
- ▶ Session objects exist on only two servers.
- ▶ Secondary:
 - The server is determined by the replication group and machine definition
 - The object is created immediately after the primary object is created
- ▶ Primary failure makes the backup object the primary object.



In-Memory Replication

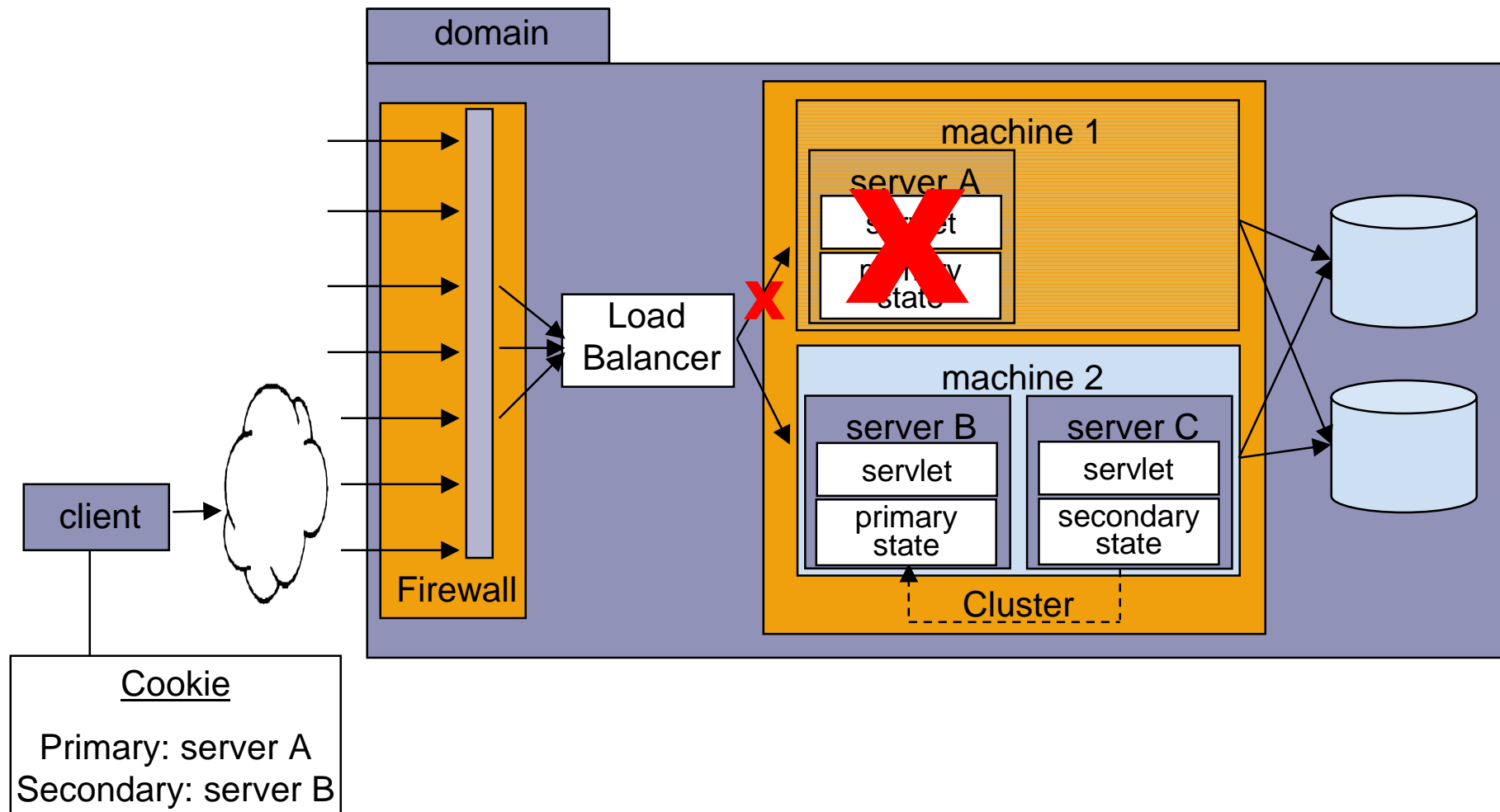


Before



After

Failover with Replication Groups



Requirements for In-Memory Replication



- ▶ Subsequent requests from the same client must have access to the same primary object.
- ▶ To utilize in-memory replication for HTTP session state, clients must access the cluster using either:
 - load-balancing hardware (WLS aware)
 - a collection of Web servers, or a single Web server, with WebLogic proxy plug-ins (configured identically)
 - WebLogic server configured with HTTPClusterServlet

Configuring In-Memory Replication



- ▶ To configure in-memory replication:
 1. Configure a proxy server (if applicable).
 2. Configure replication groups and/or machines (optional).
 3. Specify the persistence type in the `weblogic.xml` deployment descriptor:

Configuring In-Memory Replication Session Persistence:

```
<session-descriptor>  
  <persistent-store-type>replicated</persistent-store-type>  
</session-descriptor>
```

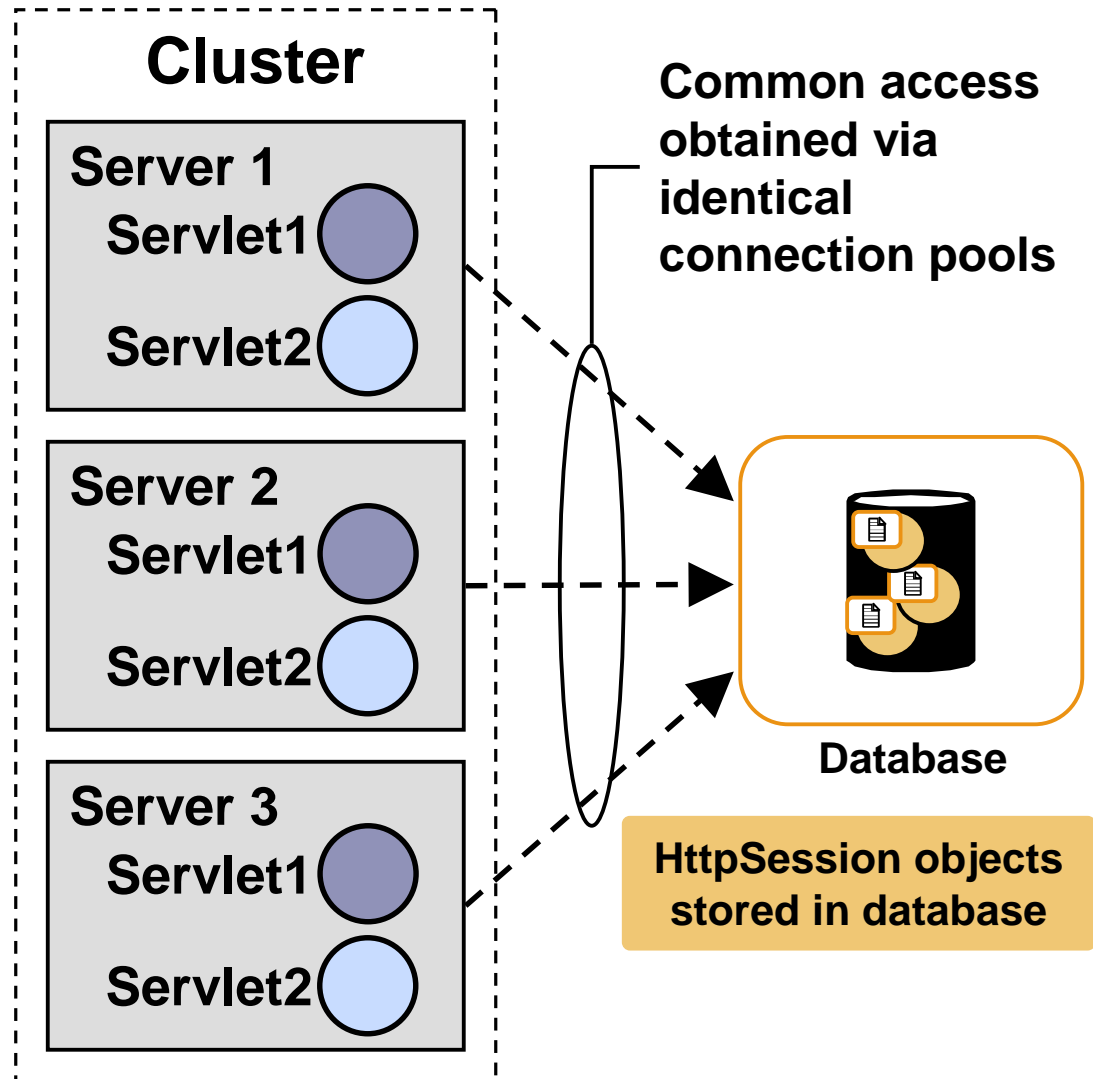
`<weblogic.xml>`



Persistent JDBC Replication



- ALL server instances have access to ALL sessions.
- Subsequent requests from same client can be handled by ANY server.
- Changing session objects causes (slow) DB synchronization.
- Great failover capability.
- Significant performance reduction.



Configuring JDBC Replication



- To configure JDBC replication:
 1. Create a table in the database.
 2. Create a connection pool that has read/write privileges for your database.
 3. Configure your session persistence in the `weblogic.xml` deployment descriptor.

Example of Configuring Session Persistence:

```
<session-descriptor>
  <persistent-store-type>jdbc</persistent-store-type>
  <persistent-store-pool>SessionDS</persistent-store-pool>
  <persistent-store-table>WL_SERVLET_SESSIONS</persistent-
store-table>
</session-descriptor>
```

`<weblogic.xml>`



JDBC Persistent Table Configuration



- ▶ A database table named WL_SERVLET_SESSIONS must exist with read/write access:

		Column Head	Column Data Type
Prim. Key {	WL_ID		char, 100 variable width char
	WL_CONTEXT_PATH		
	WL_CREATE_TIME		numeric, 20 digits
	WL_IS_VALID		char, 1 character
	WL_SESSION_VALUES		BLOB, very large
	WL_ACCESS_TIME		numeric, 20 digits
	WL_IS_NEW		numeric, 20 digits

File Persistence



- ▶ Session state may also be stored in a file
- ▶ For file-base persistence:
 - You must create the directory in which to store the file
 - The file must have the appropriate access privileges

Configuring File Persistence



- ▶ To configure file-based session persistence for a Web application:
 1. Create a folder shared by all servers on the cluster.
 2. Specify the file persistence type in the `weblogic.xml` deployment descriptor:

Example of Configuring Session Persistence:

```
<session-descriptor>
  <persistent-store-type>file</persistent-store-type>
  <persistent-store-dir>shared folder location</persistent-
store-dir>
</session-descriptor>
```

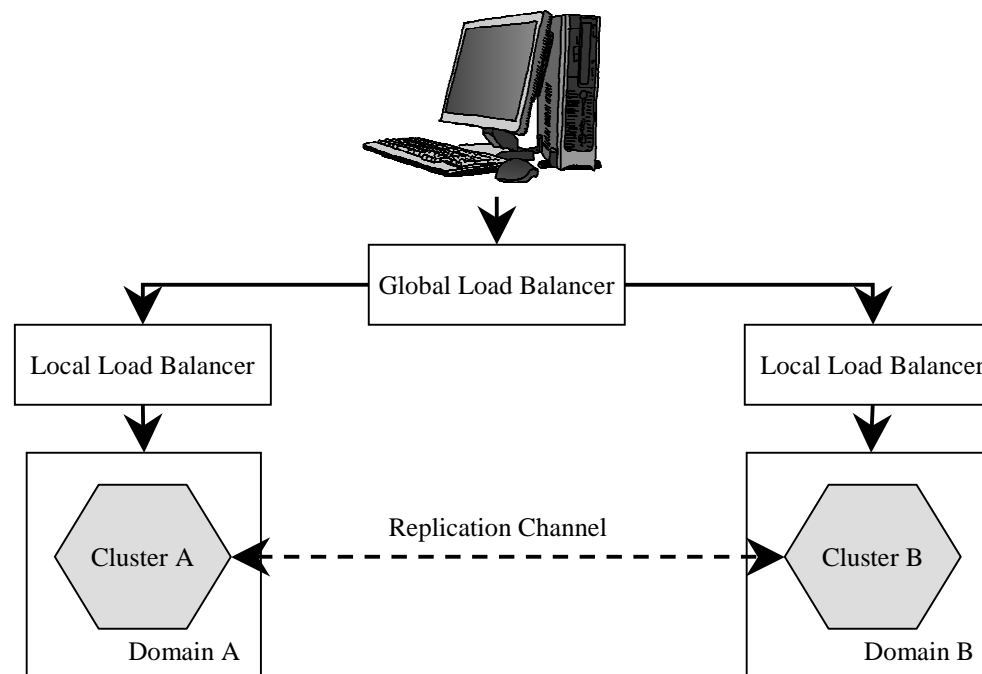
`<weblogic.xml>`



Session State Replication Across Clusters



- ▶ HTTP session state across clusters improves high-availability and fault tolerance.
- ▶ Cross-cluster replication is achieved using global and local hardware load balancers.



Session Replication in a MAN



- ▶ In the case of a MAN, if you want session replication between two cluster instances because they are relatively close together, you can replicate session states *synchronously*.
 - Assumes there is a high-speed, low-latency connection between the two clusters
 - Uses in-memory replication
- ▶ MAN replication relies on a global load balancer to maintain cluster affinity and a local load balancer to maintain server affinity.

Session Replication in a WAN



- ▶ If you want to failover requests from one cluster to another across a WAN in the event that one of the clusters completely dies, you can use the *WAN session state replication*.
- ▶ WAN session state replication entails:
 - In-memory replication of the session state to a local server instance
 - Asynchronous JDBC persistence to a remote cluster instance

Configuration Requirements for Cross-Cluster Replication




- ▶ Load balancers must be installed
 - They must be able to maintain session ids
 - The cluster failover timeout should not be set too high
 - They must know which backup cluster to use when a primary cluster or server fails
- ▶ Domains must be configured
 - Each domain must be configured identically
 - Application deployment must be identical
 - The domains need to trust each other
- ▶ In a WAN environment, there must be a data source setup to maintain session state.
- ▶ The cluster must be configured for cross-cluster replication.

Configuring a Cluster for Cross-Cluster Replication




Settings for Cluster

Configuration **Monitoring** Control Deployments Services Notes
General Multicast Servers **Replication** Migration Overload Health Monitoring HTTP


 **Cross-cluster Replication Type:** WAN (Asynchronous) HTTP Session State Replication ▼

 **Remote Cluster Address:** ServerX:8001,ServerY:8

 **Replication Channel:** ReplicationChannel

 **Data Source For Session Persistence:** ClientDS ▼

▼ Advanced

 **Session Flush Interval:** 180

 **Session Flush Threshold:** 100

 **Inter-Cluster Comm Link Health Check Interval:** 30000

☒ **Enable Replication Timeout**

State Management Best Practices



- ▶ Create WLS machines if you are replicating state across servers on different physical machines.
- ▶ Use Replication Groups to define failover strategy.
- ▶ Choose the most apropos replication strategy depending upon application needs and architecture.
- ▶ Use the ServerDebugConfig MBean to track session replication problems.
- ▶ Ensure that objects placed in replicated sessions are serializable.

Section Review



In this section, we learned how to:

- ✓ Configure HTTP session clustering
- ✓ Configure replication groups
- ✓ Configure in-memory replication
- ✓ Configure JDBC and file persistent replication
- ✓ Replicate session state within a MAN or WAN



1. Deploying Applications to Clusters
2. Session Management
3. **Troubleshooting a Cluster**
 - Invalid license
 - Server versions
 - Multicast
 - CLASSPATH
 - Thread Count
 - Garbage collection

Invalid License



- ▶ If a valid cluster-enabled license is not found in `license.bea` when WebLogic Server starts it will not join a cluster.
 - You will see the error message "Unable to find a license for clustering"

Server Version



- ▶ All servers in a cluster must have the same major version number.
 - Servers can have different minor version numbers and service packs
- ▶ The administration server should also have the same major version number

Multicast



- ▶ If there is an IP multicast problem, WLS starts but does not join a cluster.
- ▶ To verify that multicast is working, run the utility `utils.MulticastTest`.

Valid IP Multicast Addresses:

`224.0.0.0 to 239.255.255.255`

IP Multicast Test Utility:

```
java utils.MulticastTest -N <name> -A <multicastAddress>
```

`-N name (REQUIRED)`

`-P port number`

`-S send pause`

`-A address (REQUIRED)`

`-T timeout in seconds`

`-L time to live`



Example: Test Multicast



```
C:\WINNT\system32\CMD.EXE
C:\>java utils.MulticastTest -N hello -A 237.0.0.10
***** WARNING ***** WARNING ***** WARNING *****
Do NOT use the same multicast address as a running WLS cluster.

Starting test.  Hit any key to abort

Using multicast address 237.0.0.10:7001
Will send messages under the name hello every 2 seconds
Will print warning every 600 seconds if no messages are received

New Neighbor world found on message number 1
Received message 2 from world
Received message 2 from hello
Received message 3 from world
Received message 3 from hello
Received message 4 from world
Received message 4 from hello
Received message 5 from world

C:\WINNT\system32\CMD.EXE
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>java utils.MulticastTest -N world -A 237.0.0.10
***** WARNING ***** WARNING ***** WARNING *****
Do NOT use the same multicast address as a running WLS cluster.

Starting test.  Hit any key to abort

Using multicast address 237.0.0.10:7001
Will send messages under the name world every 2 seconds
Will print warning every 600 seconds if no messages are received

I (world) sent message num 1
New Neighbor hello found on message number 1
Received message 2 from world
I (world) sent message num 2
Received message 2 from hello
Received message 3 from world
I (world) sent message num 3
Received message 3 from hello
Received message 4 from world
I (world) sent message num 4
```

CLASSPATH



- ▶ The CLASSPATH value must be the same on all managed servers in the cluster.
- ▶ The CLASSPATH is set by the setEnv script when the startManagedWebLogic file is run.

Garbage Collection



- ▶ The frequency and length of the garbage collection can effect a cluster.
- ▶ If garbage collection is taking too long, the servers will not be able to make the heartbeat signals.
- ▶ Heap allocation can be tuned to adjust the length of garbage collection.
 - This can be changed using the `-ms` and `-mx` parameters in the server startup script

Section Review



In this section, we learned how to:

- ✓ Troubleshoot common issues with starting a cluster of servers



Manage HTTP Session States

- ▶ For details on the exercise, refer to the Lab Guide.
- ▶ If questions arise, ask the instructor.
- ▶ The instructor will determine the stop time.



Module Review



In this module, we learned how to:

- ✓ Deploy applications to a cluster
- ✓ Manage and configure session state in a cluster
- ✓ Troubleshoot common issues in a cluster

