

# 14

## Clusters

Communication, Planning, and Troubleshooting

# Objectives

After completing this lesson, you should be able to:

- Describe the differences between unicast and multicast cluster communication
- Configure a replication channel for a cluster
- Describe planning for a cluster
- Monitor and troubleshoot a cluster

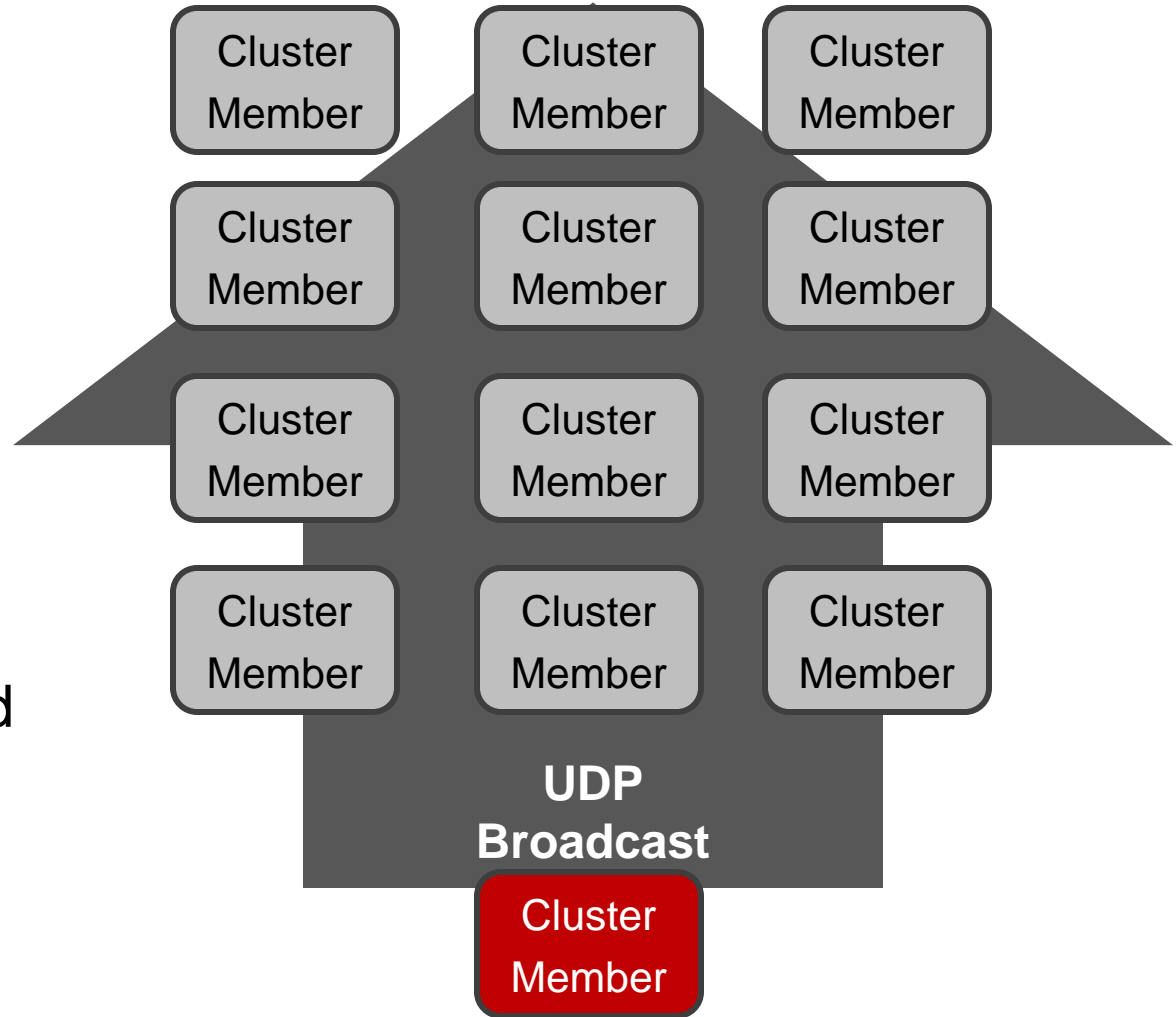
# Review: Cluster Communication

- Cluster members communicate with each other in two ways:
  - One-to-many messages
    - For periodic “heartbeats” to indicate continued availability
    - To announce the availability of clustered services
    - **Note:** This communication can use either:
      - IP unicast (recommended): No additional configuration is required.
      - IP multicast: A multicast host and port must be configured.
  - Peer-to-peer messages
    - For replicating HTTP session and stateful session EJB state
    - To access clustered objects that reside on a remote server (multi-tier architecture)
    - **Note:** This communication uses sockets.



# How Multicast Works

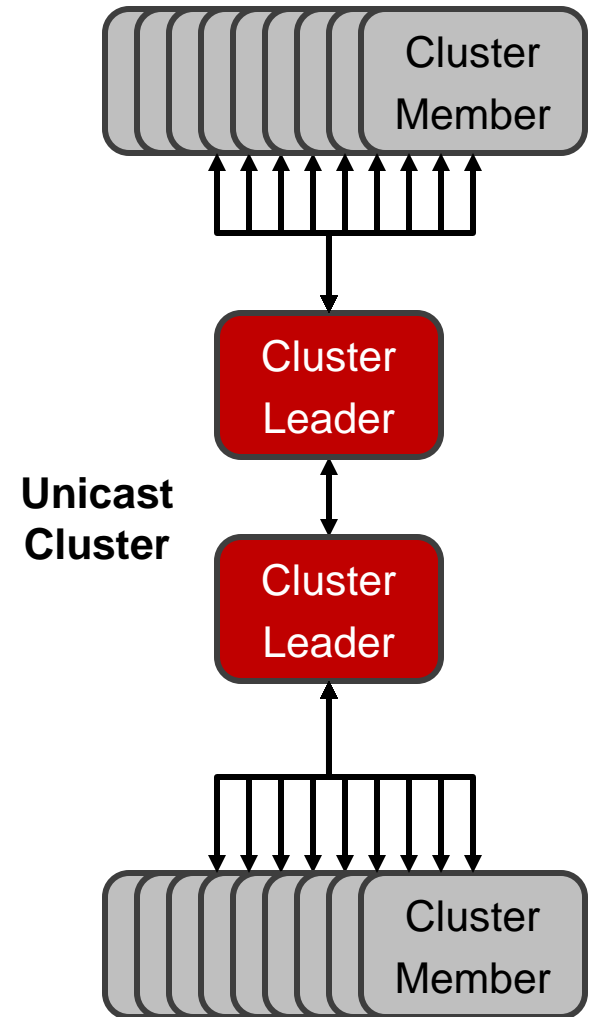
Oracle does not recommend using multicast communication and supports it only for backward compatibility.



# How Unicast Works

Unicast messaging:

- Uses TCP-IP networking
- Creates a connection for each server
- Uses a hub-and-spoke design so that it scales
- Divides a cluster into groups and assigns a group leader to each
- Enables group leaders to manage communication between groups



# Unicast Versus Multicast

Unicast communication is preferred for the following reasons:

- TCP-IP is the defacto standard protocol of the Internet.
- Many companies do not support multicast in production.
- Several networking devices do not support multicast.
- Unicast is easier to configure and reduces traffic.

# Configure Multicast

First, you should test if the multicast address you want to use is working using the MulticastTest tool.

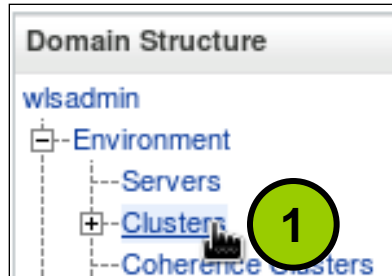
```
. ./setDomainEnv.sh
java utils.MulticastTest -n hello -a 237.0.0.1 -p 30000
```

Command Line Host 1

```
. ./setDomainEnv.sh
java utils.MulticastTest -n world -a 237.0.0.1 -p 30000
. . .
Using multicast address 237.0.0.1:30000
Will send messages under the name server1 every 2 seconds
Will print warning every 600 seconds if no messages are received
    New Neighbor hello found on message number 2
    I (world) sent message num 1
Received message 3 from hello
    I (world) sent message num 2
Received message 2 from world
Received message 4 from hello
    I (world) sent message num 3
```

Command Line Host 2

# Configure Multicast



	Name	Cluster Address
	cluster1	(2)



Messaging Mode: Multicast (4)

Unicast Broadcast Channel:

Multicast Address: 237.0.0.1

Multicast Port: 7001

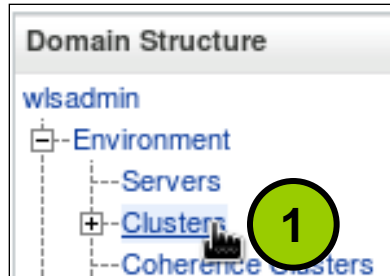
Advanced

Save

Name	Cluster Address	Cluster Messaging Mode
cluster1		Multicast (5)



# Configure Unicast



	Name	Cluster Address
	cluster1	(2)

Unicast can use its own network channel if you have one configured for this purpose.



Messaging Mode: Unicast (4)

Unicast Broadcast Channel:

Multicast Address: 237.0.0.1

Multicast Port: 7001

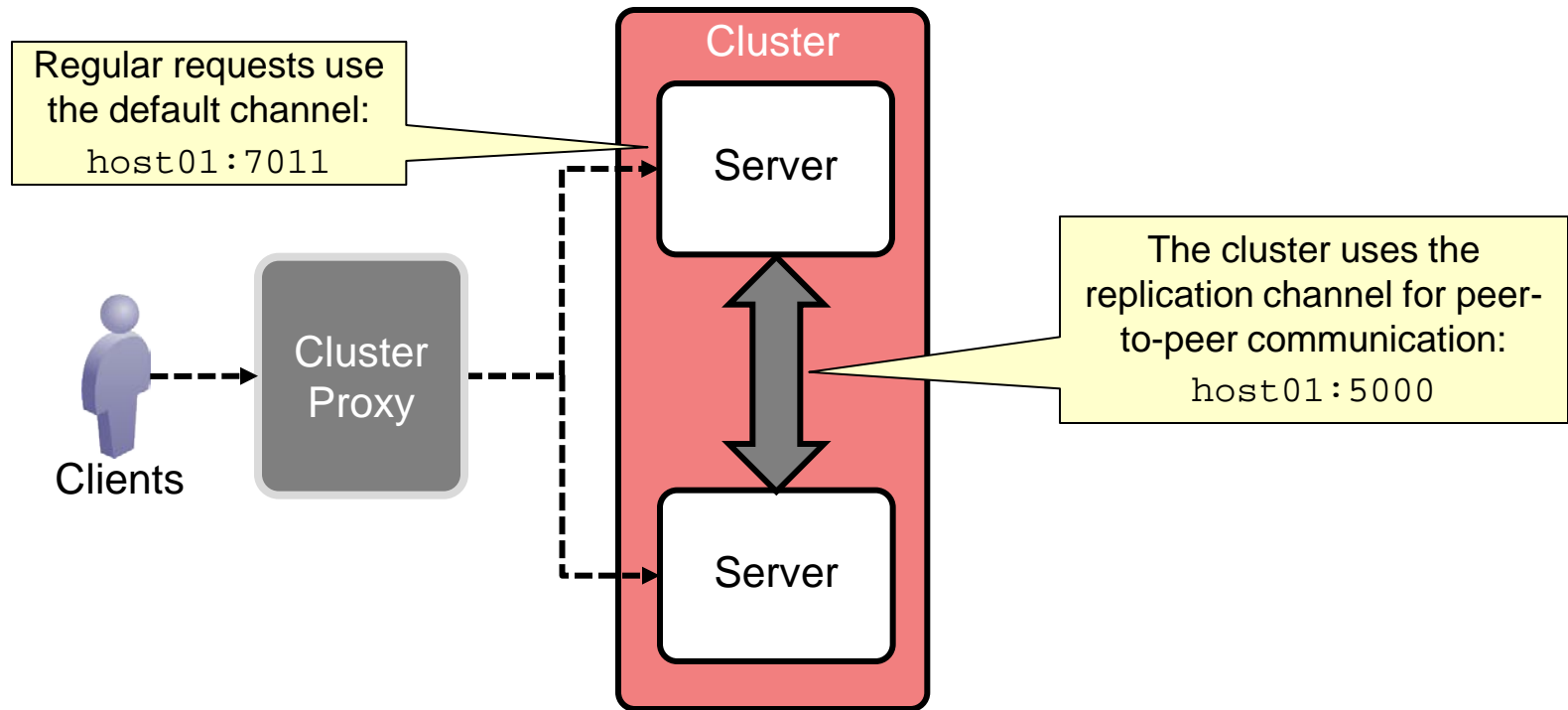
Advanced

Save

Name	Cluster Address	Cluster Messaging Mode
cluster1		Unicast (5)

# Replication Channel

WebLogic Server allows you to configure a separate network channel for peer-to-peer cluster communication (replication).



# Configure Replication Channels: Servers

First, configure each server with a network channel:

**Domain Structure**

wlsadmin

- Environment
- Servers
- Clusters

Name	Type	Cluster	Machine
AdminServer(admin)	Configured		machine1
server1	Configured	cluster1	machine1
server2	Configured	cluster1	machine2

**Settings for server1**

Configuration Protocols Logging Debug

General HTTP jCOM IIOP Channels

**Network Channels**

New

# Configure Replication Channels: Servers

Create a New Network Channel

Back Next Finish Cancel 6

Create a new Network Channel.

Identity Properties

\* Indicates required fields

What would you like to name your new Channel?

\* Name: ReplicationChannel

What protocol will be used on this channel?

\* Protocol: t3

Create a New Network Channel

Back Next Finish Cancel 7

Create a new Network Channel.

Network Channel Addressing

How would you like to address your new Network Channel?

Listen Address: host01

Listen Port: 5000

External Listen Address:

External Listen Port:

8

host02

5000

# Configure Replication Channels: Servers

Create a New Network Channel

Back Next Finish Cancel

Create a new Network Channel. **9**

Network Channel Properties

Set any additional properties for this channel. The

☒ Enabled

☐ Tunneling Enabled

☒ HTTP Enabled for This Protocol

☒ Outbound Enabled

Create a New Network Channel

Back Next Finish Cancel

Create a new Network Channel. **10**

Secure Network Channel




Define the security configuration of this network




This server is configured to use Demo Identity A

☐ Two Way SSL Enabled

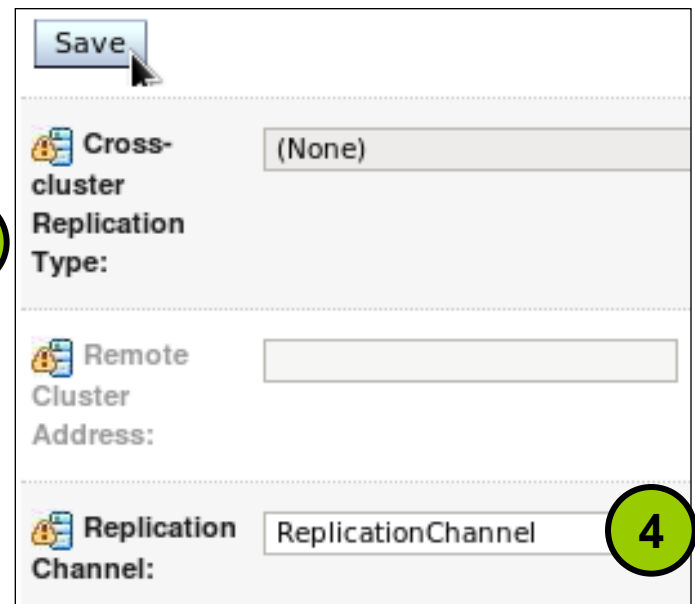
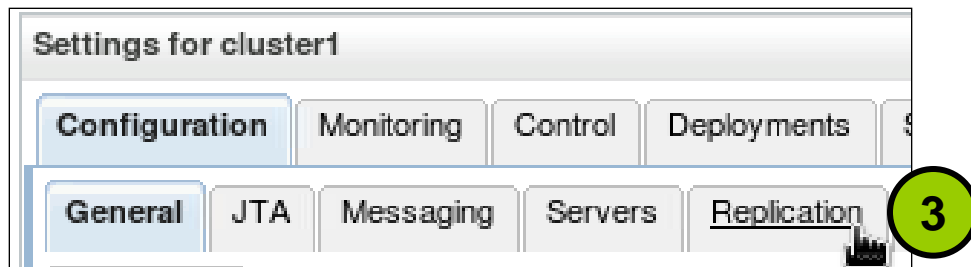
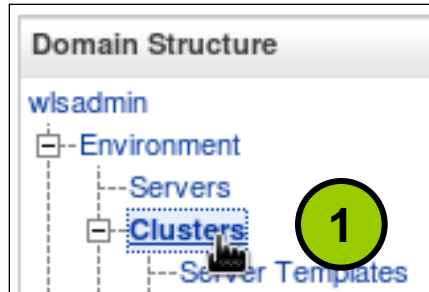
☐ Client Certificate Enforced

**11**

 Name 	Protocol	Enabled	Listen Address	Listen Port
 ReplicationChannel	t3	true	host01	5000

 Name 	Protocol	Enabled	Listen Address	Listen Port
 ReplicationChannel	t3	true	host02	5000

# Configure Replication Channels: Cluster



# Configure Replication Channels

You can verify that your replication channel is enabled by checking the system out of each server in the cluster.

```
<Notice> <Server> <BEA-002613> <Channel "ReplicationChannel"  
is now listening on 192.0.2.11:5000 for protocols t3, CLUSTER-  
BROADCAST, http.>
```

server1 output

```
<Notice> <Server> <BEA-002613> <Channel "ReplicationChannel"  
is now listening on 192.0.2.12:5000 for protocols t3, CLUSTER-  
BROADCAST, http.>
```

server2 output

# Planning for a Cluster

1. Determine your cluster architecture.
  - Basic
  - Multi-tier
2. Consider your network and security topologies.
  - A. Where to place firewalls
    - Do not place firewalls in between cluster members.
  - B. Decide on one-to-many cluster communication
    - Multicast
    - Unicast
3. Determine the type of cluster you will define.
  - Regular cluster
  - Dynamic cluster



# Planning for a Cluster

4. Choose hosts for the cluster.
  - A. Note each host's DNS name or IP address. DNS or virtual host names are recommended.
  - B. Choose the port number for each managed server\*. Note the admin server host and port.
  - C. Decide on the names of servers\*, machines, clusters, and so on (each WebLogic resource must have a unique name).
  - D. Start with one managed server per CPU core.
    - You can scale up later based on performance testing.
5. Choose your cluster proxy
  - Web server with a proxy plug-in
  - Hardware load balancer

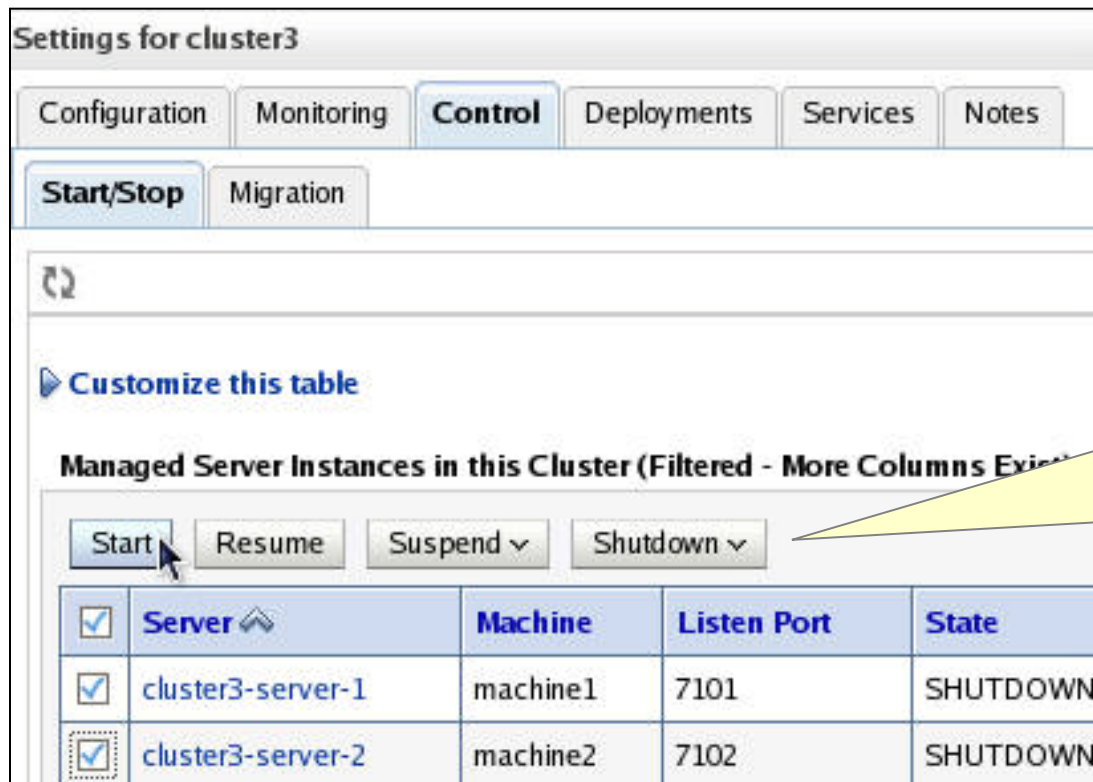
\* With Dynamic Clusters, some of these values are generated. For example, the server name prefix or starting port number is defined.

# Planning for a Cluster

6. Decide how to handle HTTP session failover.
  - In-memory replication
  - File storage
  - JDBC storage
  - Coherence\*Web
7. If using the multi-tier architecture with EJBs, decide on the EJB load balancing algorithm.
  - Round-robin
  - Random
  - Weight-based
8. Decide how pinned services will be handled.
  - Service-level migration
  - Whole server migration

# Managing a Cluster

Select the cluster in the Clusters table and click the **Control** tab. The **Start/Stop** subtab shows the servers in the cluster and allows you to start, stop, suspend, and resume them.



Settings for cluster3

Configuration Monitoring **Control** Deployments Services Notes

**Start/Stop** Migration

Customize this table

Managed Server Instances in this Cluster (Filtered - More Columns Exist)

Start Resume Suspend ▾ Shutdown ▾

<input checked="" type="checkbox"/>	Server ⤴	Machine	Listen Port	State
<input checked="" type="checkbox"/>	cluster3-server-1	machine1	7101	SHUTDOWN
<input checked="" type="checkbox"/>	cluster3-server-2	machine2	7102	SHUTDOWN

The same functions as under the Servers table **Control** tab.

# Troubleshooting a Cluster

When there are issues with a cluster, you have tools to help:

- WebLogic Server logs
- OHS logs
- Monitoring by using the administration console or the Monitoring Dashboard
- WLST

Common problems include:

- OHS to WebLogic Server connectivity issues
- Multicast communication issues (if using multicast)
- Cluster member uniformity problems
- Session failover issues

# Monitoring a Cluster: Admin Console

- In the administration console: Select the cluster from the Clusters table, use its Monitoring tab and its subtabs.

Settings for cluster3

Configuration

Monitoring

Control

Deployments

Services

Notes


Summary

Health

Failover

Server Status (Filtered - More Columns Exist)

Showing 1 to 2 of 2   Previous

Name 	State	Drop-out Frequency	Heap Free Current	Heap Size Current	Open Sockets
cluster3-server-1	RUNNING	Never	197064912	322437120	3
cluster3-server-2	RUNNING	Rarely	207932504	322699264	3

# WebLogic Server and OHS Logs

- The WebLogic Server logs contain cluster subsystem messages.
  - Set debug flags to generate more detailed log messages:
    - In the administration console, select the server from Servers table, click the **Debug** tab, expand scopes, and select flags.
- The OHS logs are found here:  
`<ORACLE_INSTANCE>/diagnostics/logs/OHS/OHS_instance_name`
  - The OHS error log: `<OHS_INSTANCE_NAME>.log`
    - Records OHS errors, but can be configured to record events.
  - The OHS access log: `access_log`
    - Records which components and applications are accessed and by whom

# Common OHS to WLS Connectivity Issues

- Connectivity problems can cause unnecessary failovers or HTTP errors to be sent to the client.
- Causes of unexpected connection failures include problems with these OHS parameters:
  - `WebLogicCluster` (the initial list of cluster members)
    - If this list is incorrect, the plug-in may not be able to proxy.
  - `ConnectTimeoutSecs` (how long the plug-in waits to establish a connection)
    - If this is set too low, the plug-in can give up on a server and not connect to it.
  - `ConnectRetrySecs` (pause time before retrying a connection)
    - If this is accidentally set higher than `ConnectTimeoutSecs`, the plug-in will always time out during a retry.

# Common OHS to WLS Connectivity Issues

- Causes of unexpected request failures include problems with these OHS parameters:
  - `WLIOTimeoutSecs` (the amount of time the plug-in waits for a response from WebLogic Server)
    - If this is set too low, and WebLogic Server sometimes takes a long time to process a request, that server will be considered dead by OHS, even though it is not.
  - `MaxPostSize` (the size of a post)
    - If this is set too low on either the proxy or on the WebLogic Server instance, a request can fail because the request is too large.
  - `MaxSkipTime` (the wait time before the plug-in retries a server marked as “bad”)
    - If this is set too high, the proxy will be slow to use a restarted cluster member, affecting overall performance.



# Multicast Communication Issues

- Problem with the multicast address
  - For each cluster on the network, the combination of the multicast address and port must be unique.
  - Ensure no other applications use that address and port.
- Missed multicast messages (heartbeats) can cause cluster members to be marked as “failed.”
  - Ensure the multicast time to live (TTL) value is large enough for the messages to get to all cluster members.
  - If multicast buffers fill up, messages are missed.
    - Increase the size of the multicast buffer.
    - Increase the multicast send delay, which helps avoid buffer overflow.

One reason unicast is recommended is that there are generally less issues with it.

# Cluster Member Uniformity

- Every instance of WebLogic Server in a cluster should be like every other. All servers should:
  - Be the same version of WebLogic Server
  - Have the same `CLASSPATH`
  - Have the same deployments
  - Have the same services (like data sources)
- When cluster members are not the same, you have intermittent problems, which are very hard to debug.

# Session Failover Issues

- Session replication or persistence problems often result in the loss of session data. This affects your clients:
  - A client must log in again.
  - The client's shopping cart items disappear.
- Typical culprits include:
  - Invalid session persistence settings
  - Session or cookie timeout settings are too low.
  - The developers of the web application did not use the `HttpSession` API appropriately.
  - The developers of the web application are storing non-serializable objects in the session.
    - Objects must be serializable so they can be streamed from the primary server to the secondary server (in-memory replication) or to files or the database (file or database persistence).

# Quiz

A replication channel is:

- a. Another name for replication group
- b. Another name for the preferred secondary group
- c. A network channel used by cluster members for peer-to-peer communication
- d. The title of the tab in the Monitoring Dashboard that shows cluster charts

# Summary

In this lesson, you should have learned how to:

- Describe the differences between unicast and multicast cluster communication
- Configure a replication channel for a cluster
- Describe planning for a cluster
- Monitor and troubleshoot a cluster

# **Practice 14-1 Overview: Configuring a Replication Channel**

This practice covers configuring a replication channel for a cluster.