

# Exercise 10 – Clustering and Workload Management

At the end of the lab, you should be able to:

- Create a cluster and add cluster members
- Map modules to clusters and web servers
- Test load balancing and failover between two cluster members
- Configure a data replication domain for session management

```
wasadm      soft  nproc      131072
wasadm      hard   nproc      131072
wasadm      soft  nofile     65536
wasadm      hard   nofile     65536
wasadm      soft   core      unlimited
wasadm      hard   core      unlimited
```

```
...ents/DigitalVizyon/labdocs — mulker@wasnd-node03: ~ — ssh + gcloud.py compute ssh --ssh-flag=-
[wasadm@wasnd-node01:/ibm$ sudo vi /etc/security/limits.conf

#*
#root      soft  core      0
#root      hard   core     100000
#*
#@student  hard   rss      10000
#@student  hard   nproc     20
#@faculty  soft   nproc     20
#@faculty  hard   nproc     50
#ftp       hard   nproc     0
#ftp       -      chroot   /ftp
#@student  -      maxlogins 4
wasadm     soft   nproc     131072
wasadm     hard   nproc     131072
wasadm     soft  nofile     65536
wasadm     hard   nofile     65536
wasadm     soft   core      unlimited
wasadm     hard   core      unlimited

# End of file
```

## Section 1: Check nodes and node agents

Before you can begin creating the cluster, make sure that both node agents are running and the nodes are synchronized.

Cell=wasnd-node01Cell01, Profile=Dmgr

**Node agents**

**Node agents**

Use this page to manage node agents and application servers on the node that a node agent manages. The node agent process serves as an intermediary between the application servers on the node and the deployment manager. The node agent process runs on every node and is specialized to perform node-specific administration functions, such as server process monitoring, configuration synchronization, file transfer, and request routing.

**Preferences**

**Stop** **Restart** **Restart all Servers on Node**

   

Select	Name	Node	Host Name	Version	Status
<input type="checkbox"/>	<a href="#">nodeagent</a>	wasnd-node01Node02	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	ND 8.5.5.20	
<input type="checkbox"/>	<a href="#">nodeagent</a>	wasnd-node01Node01	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	ND 8.5.5.20	

Total 2

Cell=wasnd-node01Cell01, Profile=Dmgr

**Nodes**

**Nodes**

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to this list by clicking Add Node.

**Preferences**

**Add Node** **Remove Node** **Force Delete** **Synchronize** **Full Resynchronize** **Stop**

   

Select	Name	Host Name	Version	Discovery Protocol	Status
<input type="checkbox"/>	<a href="#">lhsnode</a>	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	Not applicable	TCP	
<input type="checkbox"/>	<a href="#">wasnd-node01CellManager01</a>	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	ND 8.5.5.20	TCP	
<input type="checkbox"/>	<a href="#">wasnd-node01Node01</a>	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	ND 8.5.5.20	TCP	
<input type="checkbox"/>	<a href="#">wasnd-node01Node02</a>	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	ND 8.5.5.20	TCP	

Total 4

## Section 2: Creating the *PlantsCluster* cluster

A cluster is composed of two or more servers in a cell, which are assigned to run the same applications. Clusters are logical abstractions that are equivalent to servers. In this section, you create the cluster that contains the cluster members that participate in workload management of the Plants application. You create a cluster that is called PlantsCluster. This cluster is created based on the existing server1 application server. This action means that all of the applications that are already deployed to server1 are included in the cluster.

Cell=wasnd-node01Cell01, Profile=Dmgr

### WebSphere application server clusters

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New... Delete Start Stop Ripplestart ImmediateStop

Select Name Status

None

Total 0

Cell=wasnd-node01Cell01, Profile=Dmgr

### Create a new cluster

Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

Step 3: Create additional cluster members

Step 4: Summary

Enter basic cluster information

\* Cluster name: PlantsCluster

Prefer local. Specifies whether enterprise bean requests will be routed to the node on which the client resides when possible.

Configure HTTP session memory-to-memory replication

Next Cancel

Cell=wasnd-node01Cell01, Profile=Dmgr

### Create a new cluster

Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

Step 3: Create additional cluster members

Step 4: Summary

Create first cluster member

The first cluster member determines the server settings for the cluster members. A server configuration template is created from the first member and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name:

Select node: wasnd-node01Node01(ND 8.5.5.20)

\* Weight:  (0..100)

Generate unique HTTP ports

Select how the server resources are promoted in the cluster: Cluster

Select basis for first cluster member:

Create the member using an application server template.

Create the member using an existing application server as a template.

Create the member by converting an existing application server.

None. Create an empty cluster.

Previous Next Cancel

Cell=wasnd-node01Cell01, Profile=Dmgr

## Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

→ Step 3: Create additional cluster members

Step 4: Summary

**Create additional cluster members**

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name

Select node

\* Weight  (0..100)

Generate unique HTTP ports

**Add Member**

Use the Edit function to modify the properties of a cluster member in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member.

Select	Member name	Nodes	Version	Weight
<input checked="" type="checkbox"/>	server1	wasnd-node01Node01	ND 8.5.5.20	2
Total 1				

**Previous** **Next** **Cancel**

Cell=wasnd-node01Cell01, Profile=Dmgr

## Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

→ Step 3: Create additional cluster members

Step 4: Summary

**Create additional cluster members**

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name

Select node

\* Weight  (0..100)

Generate unique HTTP ports

**Add Member**

Use the Edit function to modify the properties of a cluster member in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member.

Select	Member name	Nodes	Version	Weight
<input checked="" type="checkbox"/>	server1	wasnd-node01Node01	ND 8.5.5.20	2
Total 1				

**Previous** **Next** **Cancel**

**Create a new cluster**

Step 1: Enter basic cluster information

Step 2: Create first cluster member

→ **Step 3: Create additional cluster members**

Step 4: Summary

**Create additional cluster members**

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name

Select node

\* Weight  (0..100)

Generate unique HTTP ports

Use the Edit function to modify the properties of a cluster member in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member.

Select	Member name	Nodes	Version	Weight
<input checked="" type="checkbox"/>	server1	wasnd-node01Node01	ND 8.5.5.20	2
<input type="checkbox"/>	server2	wasnd-node01Node02	ND 8.5.5.20	2

Total 2

Cell=wasnd-node01Cell01, Profile=Dmgr

### Create a new cluster

Step 1: Enter basic cluster information

Step 2: Create first cluster member

Step 3: Create additional cluster members

→ Step 4: Summary

**Summary**

Summary of actions:

Options	Values
Cluster Name	PlantsCluster
Core Group	DefaultCoreGroup
Node group	DefaultNodeGroup
Prefer local	true
Configure HTTP session memory-to-memory replication	false
Server name	server1
Node	wasnd-node01Node01(ND 8.5.5.20)
Weight	2
Clone Template	wasnd-node01Cell01/wasnd-node01Node01(ND 8.5.5.20)/server1
Clone Basis	Create the member by converting an existing application server.
Select how the server resources are promoted in the cluster.	cluster
Generate unique HTTP ports	false
Server name	server2
Node	wasnd-node01Node02(ND 8.5.5.20)
Weight	2
Clone Template	Version 8.5 member template
Generate unique HTTP ports	true

Previous | Finish | Cancel

Cell=wasnd-node01Cell01, Profile=Dmgr

### WebSphere application server clusters

**WebSphere application server clusters**

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

New... | Delete | Start | Stop | Ripplestart | ImmediateStop

Select	Name	Status
<input type="checkbox"/>	PlantsCluster	

You can administer the following resources:

Total 1
---------

## Communications

### Ports

Port Name	Port	Details
BOOTSTRAP_ADDRESS	9811	
SOAP_CONNECTOR_ADDRESS	8882	
ORB_LISTENER_ADDRESS	9103	
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9407	
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9408	
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9409	
WC_adminhost	9062	
WC_defaulthost	9081	
DCS_UNICAST_ADDRESS	9356	
WC_adminhost_secure	9045	
WC_defaulthost_secure	9444	
SIP_DEFALTHOST	5061	Specifies the TCP/IP port for SIP traffic.
SIP_DEFALTHOST_SECURE	5063	
SIB_ENDPOINT_ADDRESS	7278	
SIB_ENDPOINT_SECURE_ADDRESS	7287	
SIB_MQ_ENDPOINT_ADDRESS	5559	
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5579	
IPC_CONNECTOR_ADDRESS	9634	
OVERLAY_UDP_LISTENER_ADDRESS	11009	
OVERLAY_TCP_LISTENER_ADDRESS	11010	

Cell=wasnd-node01Cell01, Profile=Dmgr

**Virtual Hosts**

**Virtual Hosts > default\_host > Host Aliases**

Use this page to edit, create, or delete a domain name system (DNS) alias by which the virtual host is known.

Preferences

**New... Delete**

Select Host Name Port

You can administer the following resources:

<input type="checkbox"/>	*	9080
<input type="checkbox"/>	*	80
<input type="checkbox"/>	*	9443
<input type="checkbox"/>	*	5060
<input type="checkbox"/>	*	5061
<input type="checkbox"/>	*	443
<input type="checkbox"/>	*	9081
<input type="checkbox"/>	*	9444

Total 8

Cell=wasnd-node01Cell01, Profile=Dmgr

**WebSphere application server clusters**

**WebSphere application server clusters**

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

**New... Delete Start Stop Ripplestart ImmediateStop**

Select Name Status

You can administer the following

<input checked="" type="checkbox"/>	PlantsCluster	
-------------------------------------	---------------	--

Total 1

Cell=wasnd-node01Cell01, Profile=Dmgr

**WebSphere application server clusters**

**WebSphere application server clusters**

Use this page to change the configuration settings for a cluster. A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Preferences

**New... Delete Start Stop Ripplestart ImmediateStop**

Select Name Status

You can administer the following resources:

<input type="checkbox"/>	PlantsCluster	
--------------------------	---------------	--

Total 1

## Section 3: Set the applications to run on the cluster

Now that the cluster is defined, the next step is to configure the applications to run on the cluster, rather than on individual servers. Since the web server is used to workload manage the web containers, the web server also must be mapped to the applications. This step is important as it allows the customized plugin.cfg.xml files to include the appropriate URIs for each of the applications they are supposed to host.

Cell=wasnd-node01Cell01, Profile=Dmgr

**Enterprise Applications**

[Enterprise Applications](#) > [pbw-ear](#) > [Manage Modules](#)

Manage Modules

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin.cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

WebSphere:cell=wasnd-node01Cell01,cluster=PlantsCluster	WebSphere:cell=wasnd-node01Cell01,node=ihsnod,server=webserver1	<input type="button" value="Apply"/>
---	---	--------------------------------------

Select	Module	URI	Module Type	Server
<input type="checkbox"/>	<a href="#">PlantsByWebSphere</a>	PlantsByWebSphere.war,WEB-INF/web.xml	Web Module	WebSphere:cell=wasnd-node01Cell01,node=ihsnod,server=webserver1 WebSphere:cell=wasnd-node01Cell01,cluster=PlantsCluster

Cell=wasnd-node01Cell01, Profile=Dmgr

**Web servers**

[Messages](#)

Server ihsnode/webserver1 started successfully. The collection may need to be refreshed to show the current server status.

**Web servers**

Use this page to view a list of the installed web servers.

Preferences

Select	Name	Web server Type	Node	Host Name	Version	Status
<input type="checkbox"/>	<a href="#">webserver1</a>	IBM HTTP Server	ihsnod	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.Internal	Not applicable	

Total 1

Cell=wasnd-node01Cell01, Profile=Dmgr

Web servers

Messages

- PLGC0062I: The plug-in configuration file is propagated from /ibm/profiles/Dmgr/config/cells/wasnd-node01Cell01/nodes/lhsnode/servers/webserver1/plugin-cfg.xml to /ibm/WebSphere/Plugins/config/webserver1/plugin-cfg.xml on the Web server computer.
- PLGC0048I: The propagation of the plug-in configuration file is complete for the Web server. wasnd-node01Cell01.lhsnode.webserver1.

**Web servers**

Use this page to view a list of the installed web servers.

Preferences

Generate Plug-in Propagate Plug-in New... Delete Templates... Start Stop Terminate

Select Name Web server Type Node Host Name Version Status

You can administer the following resources:

<input type="checkbox"/>	webserver1	IBM HTTP Server	lhsnode	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	Not applicable	
Total 1						

## ***Section 4: Create a cluster scoped JDBC resource***

When creating the first cluster member from the existing server1, all resources that are already defined at the server and node scope are maintained.

Unfortunately, when adding the second server on the was85hostNode02 node, the resource definitions from server1 and was85hostNode01 are not automatically defined. You now have a problem: since both servers run the same applications, by virtue of being on the same cluster, they both need access to the same resources.

One solution is to re-create the resources at the node scope for each additional node as its servers are added to the cluster. That solution works, but the disadvantage is that you must do it every time a new node server is added to the cluster. A better solution is to define resources at the cluster scope.

Resources can be added at the cluster scope only if the cluster members are running in similar operating environments. Since many resources require pointers to a file system location, it does not work to define resources at the cluster scope for cluster members that run in both Windows and Linux. In that case, you must define the resources at the node level.

**JDBC providers**

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: **All scopes**

Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

All scopes

Preferences

New... Delete			
Select	Name	Scope	Description
You can administer the following resources:			
<input type="checkbox"/>	<a href="#">Derby JDBC Provider</a>	Cluster=PlantsCluster	Derby embedded non-XA JDBC Provider
<input checked="" type="checkbox"/>	<a href="#">Derby JDBC Provider</a>	Node=wasnd-node01Node01,Server=server1	Derby embedded non-XA JDBC Provider
<input checked="" type="checkbox"/>	<a href="#">Derby JDBC Provider</a>	Node=wasnd-node01Node02,Server=server2	Derby embedded non-XA JDBC Provider
<input checked="" type="checkbox"/>	<a href="#">Derby JDBC Provider (XA)</a>	Node=wasnd-node01Node01	Derby embedded XA JDBC Provider. This provider is only configurable in version 6.0.2 and later nodes
<a href="#">OTISDataSource</a>			
Total 5			

**JDBC providers**

Use this page to edit properties of a JDBC provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: **All scopes**

Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

All scopes

Preferences

New... Delete			
Select	Name	Scope	Description
You can administer the following resources:			
<input type="checkbox"/>	<a href="#">Derby JDBC Provider</a>	Cluster=PlantsCluster	Derby embedded non-XA JDBC Provider
<a href="#">OTISDataSource</a>			
Total 2			

Cell=wasnd-node01Cell01, Profile=Dmgr

**Data sources**

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**wasnd-node01Cell01**, Cluster=**PlantsCluster**

Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Cluster=PlantsCluster

**Preferences**

New... Delete Test connection Manage state...

Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	<a href="#">Default Datasource</a>	DefaultDatasource	Cluster=PlantsCluster	Derby JDBC Provider	Datasource for the WebSphere Default Application	

Total 1

Cell=wasnd-node01Cell01, Profile=Dmgr

**Create a data source**

Create a data source

→ Step 1: Enter basic data source information

Step 2: Select JDBC provider

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

**Enter basic data source information**

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope: cells:wasnd-node01Cell01:clusters:PlantsCluster

\* Data source name: Plants

\* JNDI name: jdbc/PlantsByWebSphereDataSource

Next Cancel

Cell=wasnd-node01Cell01, Profile=Dmgr

### Create a data source

Create a data source

Step 1: Enter basic data source information

→ Step 2: Select JDBC provider

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

**Select JDBC provider**

Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

Create new JDBC provider

Select an existing JDBC provider

Derby JDBC Provider

Previous | Next | Cancel

Cell=wasnd-node01Cell01, Profile=Dmgr

### Create a data source

Create a data source

Step 1: Enter basic data source information

Step 2: Select JDBC provider

→ Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

**Enter database specific properties for the data source**

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* Database name	oServer/derby/databases/PLANTSDB

Use this data source in container managed persistence (CMP)

Previous | Next | Cancel

Cell=wasnd-node01Cell01, Profile=Dmgr

### Create a data source

Create a data source

Step 1: Enter basic data source information

Step 2: Select JDBC provider

Step 3: Enter database specific properties for the data source

→ Step 4: Setup security aliases

Step 5: Summary

**Setup security aliases**

Select the authentication values for this resource.

Component-managed authentication alias

(none) ▾

Mapping-configuration alias

(none) ▾

Container-managed authentication alias

(none) ▾

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

[Global J2C authentication alias](#)

[Security domains](#)

Previous | Next | Cancel

Create a data source

<p>Step 1: Enter basic data source information</p> <p>Step 2: Select JDBC provider</p> <p>Step 3: Enter database specific properties for the data source</p> <p>Step 4: Setup security aliases</p> <p>→ <b>Step 5: Summary</b></p>	<p><b>Summary</b></p> <p>Summary of actions:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #f2f2f2;">Options</th> <th style="background-color: #f2f2f2;">Values</th> </tr> </thead> <tbody> <tr> <td>Scope</td> <td>cells:wasnd-node01Cell01:clusters:PlantsCluster</td> </tr> <tr> <td>Data source name</td> <td>Plants</td> </tr> <tr> <td>JNDI name</td> <td>Jdbc/PlantsByWebSphereDataSource</td> </tr> <tr> <td>Select an existing JDBC provider</td> <td>Derby JDBC Provider</td> </tr> <tr> <td>Implementation class name</td> <td>org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource</td> </tr> <tr> <td>Database name</td> <td>/ibm/WebSphere/AppServer/derby/databases/PLANTSDB</td> </tr> <tr> <td>Use this data source in container managed persistence (CMP)</td> <td>true</td> </tr> <tr> <td>Component-managed authentication alias</td> <td>(none)</td> </tr> <tr> <td>Mapping-configuration alias</td> <td>(none)</td> </tr> <tr> <td>Container-managed authentication alias</td> <td>(none)</td> </tr> </tbody> </table>	Options	Values	Scope	cells:wasnd-node01Cell01:clusters:PlantsCluster	Data source name	Plants	JNDI name	Jdbc/PlantsByWebSphereDataSource	Select an existing JDBC provider	Derby JDBC Provider	Implementation class name	org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource	Database name	/ibm/WebSphere/AppServer/derby/databases/PLANTSDB	Use this data source in container managed persistence (CMP)	true	Component-managed authentication alias	(none)	Mapping-configuration alias	(none)	Container-managed authentication alias	(none)
Options	Values																						
Scope	cells:wasnd-node01Cell01:clusters:PlantsCluster																						
Data source name	Plants																						
JNDI name	Jdbc/PlantsByWebSphereDataSource																						
Select an existing JDBC provider	Derby JDBC Provider																						
Implementation class name	org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource																						
Database name	/ibm/WebSphere/AppServer/derby/databases/PLANTSDB																						
Use this data source in container managed persistence (CMP)	true																						
Component-managed authentication alias	(none)																						
Mapping-configuration alias	(none)																						
Container-managed authentication alias	(none)																						
<input type="button" value="Previous"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/>																							

**Messages**

⌚ The test connection operation failed for data source Plants on server nodeagent at node wasnd-node01Node02 with the following exception: java.sql.SQLException: Failed to start database '/ibm/WebSphere/AppServer/derby/databases/PLANTSDB' with class loader java.net.URLClassLoader@f2e54dc2, see the next exception for details. DSRA0010E: SQL State = XJ040, Error Code = 40,000. [View JVM logs](#) for further details.

⌚ The test connection operation failed for data source Plants on server nodeagent at node wasnd-node01Node01 with the following exception: java.sql.SQLException: Failed to start database '/ibm/WebSphere/AppServer/derby/databases/PLANTSDB' with class loader java.net.URLClassLoader@4174dd53, see the next exception for details. DSRA0010E: SQL State = XJ040, Error Code = 40,000. [View JVM logs](#) for further details.

## Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

**Scope: Cell=wasnd-node01Cell01, Cluster=PlantsCluster**

Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

**Preferences**

<input type="button" value="New..."/> <input type="button" value="Delete"/> <input type="button" value="Test connection"/> <input type="button" value="Manage state..."/>						
Select	Name	JNDI name	Scope	Provider	Description	Category
You can administer the following resources:						
<input type="checkbox"/>	<a href="#">Default Datasource</a>	DefaultDatasource	Cluster=PlantsCluster	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input type="checkbox"/>	<a href="#">Plants</a>	Jdbc/PlantsByWebSphereDataSource	Cluster=PlantsCluster	Derby JDBC Provider	New JDBC Datasource. This Datasource type is only configurable in version 6.0.2 and later nodes	

<https://www.ibm.com/support/pages/javasqlsqlexception-sql-state-xj040-error-code-40000-starting-derby-database>

## Cause

The cause of the problem is that the database was in an inconsistent state due to a previous database write failing to complete, which led to the stored database getting corrupted. Derby uses file system space to store the tables and the stored database was corrupted. One possible cause for this error is that two different processes had the database table open concurrently and wrote at the same time.

## Environment

This is a problem that occurs using a Derby database with WebSphere Application Server.

**Data sources**

**Messages**

- ⌚ The test connection operation for data source Plants was unable to connect to node wasnd-node01Node02. Ensure that the node agent is running and retry the test connection operation.
- ⌚ The test connection operation for data source Plants on server nodeagent at node wasnd-node01Node01 was successful.

**Data sources**

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application with connections for accessing the database. Learn more about this task in a [guided activity](#). A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=**wasnd-node01Cell01**, Cluster=**PlantsCluster**

Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#).

Cluster=PlantsCluster

**Preferences**

New... Delete Test connection Manage state...						
Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	<a href="#">Default Datasource</a>	DefaultDatasource	Cluster=PlantsCluster	Derby JDBC Provider	Datasource for the WebSphere Default Application	
<input checked="" type="checkbox"/>	<a href="#">Plants</a>	jdbc/PlantsByWebSphereDataSource	Cluster=PlantsCluster	Derby JDBC Provider	New JDBC Datasource. This Datasource type is only configurable in version 6.0.2 and later nodes	

Total 2

Node agents

Node agents

Use this page to manage node agents and application servers on the node that a node agent manages. The node agent process serves as an intermediary between the application servers on the node and the deployment manager. The node agent process runs on every node and is specialized to perform node-specific administration functions, such as server process monitoring, configuration synchronization, file transfer, and request routing.

Stop    Restart    Restart all Servers on Node

Select Name ▲ Node ▲ Host Name ▲ Version ▲ Status ▲

You can administer the following resources:

Select	Name	Node	Host Name	Version	Status
<input type="checkbox"/>	<a href="#">nodeagent</a>	wasnd-node01Node02	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	ND 8.5.5.20	
<input type="checkbox"/>	<a href="#">nodeagent</a>	wasnd-node01Node01	wasnd-node01.europe-west3-c.c.enhanced-casing-342608.internal	ND 8.5.5.20	

Total 2

## Section 5: Test the application

In this section of the exercise the application is tested in a clustered environment. The application is served from both application servers (cluster members) until the application creates an HTTP session object. At that point, affinity is established. This condition means that from that point on, all requests are directed to the same application server. This action is done so that the user's session information is available locally.

If the cluster member that is holding the user session becomes unavailable, the web server plug-in reroutes the request to another cluster member. However, this situation presents a problem because the new application server does not (by default) have access to the session information.

The exercise initially demonstrates this “problem,” but then later configures a solution that allows the cluster members to share their session information. As a result, even if a cluster member fails, users are still able to access their session through another cluster member.

The screenshot shows a web browser window with the URL [34.141.79.132/PlantsByWebSphere/promo.jsf](http://34.141.79.132/PlantsByWebSphere/promo.jsf). The page title is 'PLANTS BY WEBSPHERE'. The main content features a large banner with the text 'Gardens of Summer' and the subtext 'They all start with the right flowers...'. Below the banner, a photograph of a garden with a wooden bench and a trellis covered in flowers. A sidebar on the left contains a 'Tips' section with a note about preserving grass seed. The right sidebar lists 'Specials' including a Bonsai Tree (\$30.00 each), Red Delicious Strawberries (\$3.50 (50 seeds)), and Tulips (\$17.00 (10 bulbs)).

## Section 6: Configure session replication settings

In order for members of a cluster to share session information, a strategy to share session data must be put in place. WebSphere Application Server provides various mechanisms to achieve this goal. The main strategies are database and memory-to-memory replication. Setting up either of these mechanisms is straightforward. In this exercise, memory-to-memory replication is set up to handle session data replication.

Session management can be configured on each of the servers in the cluster. This action can be completed when you create the cluster or at a later point.

The screenshot shows the WebSphere administrative console interface. The left sidebar lists various management categories: Welcome, Guided Activities, Servers (New server, All servers, Server Types, Clusters, DataPower, Core Groups), Applications, Jobs, Services, Resources, Runtime Operations, Security, Operational policies, Environment (Virtual hosts, Update global Web server plug-in configuration, WebSphere variables, Shared libraries, SIP application routers, Replication domains, URL Groups, Naming, OSGi bundle repositories), and Custom administration. The right panel is titled 'Replication domains' and contains a sub-header: 'Use this page to create a list of replication domains that the session manager uses for replication of the dynamic cache service, and the stateful session bean failover components. All components that need to share information must be in the same replication domain.' It includes a 'New...' button, a toolbar with icons for New, Delete, and Sort, and a table with columns for Select, Name, and Type. The table shows a single entry: 'None' under 'Name' and 'Total 0' under 'Type'.

Cell=wasnd-node01Cell01, Profile=Dmgr

Replication domains

[Replication domains > New...](#)

Use this page to configure the replication properties that are used by all of the components of this replication domain.

Configuration

**General Properties**

\* Name: PlantsCluster

\* Request timeout: 5 seconds

**Number of replicas**

Single replica

Entire Domain

Specify

Number of replicas:

**Buttons:** Apply, OK, Reset, Cancel

Cell=wasnd-node01Cell01, Profile=Dmgr

Replication domains

**Replication domains**

Use this page to create a list of replication domains that the session manager uses for replication of the dynamic cache service, and the stateful session bean failover components. All components that need to share information must be in the same replication domain.

Preferences

**Buttons:** New..., Delete

Select: Name ▾ Type ▾

You can administer the following resources:

<input type="checkbox"/> PlantsCluster	Data replication domain
--	-------------------------

Total 1

Cell=wasnd-node01Cell01, Profile=Dmgr

Application servers

[Application servers > server1](#)

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

Reports, Operations, Runtime, Configuration

**General Properties**

Name: server1

**Container Settings**

- Session management

## Application servers

### Application servers > server1 > Session management

Use this page to configure session manager properties to control the behavior of Hypertext Transfer Protocol (HTTP) session support. These set both the SIP container and the web container.

Configuration

#### General Properties

##### Session tracking mechanism:

- Enable SSL ID tracking
- [Enable cookies](#)
- Enable URL rewriting
- Enable protocol switch rewriting

##### Maximum in-memory session count:

1000 sessions

- Allow overflow

##### Session timeout:

- No timeout
  - Set timeout
- 30 minutes

- Security integration

##### Serialize session access:

- Allow serial access
- Maximum wait time  
5 seconds
- Allow access on timeout

[Apply](#) [OK](#) [Reset](#) [Cancel](#)

#### Additional Properties

- [Custom properties](#)
- [Distributed environment settings](#)

Cell=wasnd-node01Cell01, Profile=Dmgr

## Application servers

? -

### Application servers > server1 > Session management > [Distributed environment settings](#) > Memory-to-memory replication

Use this page to configure memory-to-memory replication for failure recovery.

Configuration

#### General Properties

##### Replication domain

PlantsCluster

##### Replication mode

Both client and server

[Apply](#) [OK](#) [Reset](#) [Cancel](#)

Cell=wasnd-node01Cell01, Profile=Dmgr

**Application servers**

**Application servers > server1 > Session management > Distributed environment settings**

Use this page to specify how session data is saved in a distributed environment. The SIP container uses only memory-to-memory replication for distributed sessions.

Configuration

**General Properties**

**Distributed sessions**

- None
- [Database](#) (Supported for Web container only.)
- [Memory-to-memory replication](#)

**Additional Properties**

- [Custom tuning parameters](#)

**Buttons:** [Apply](#) [OK](#) [Reset](#) [Cancel](#)

Cell=wasnd-node01Cell01, Profile=Dmgr

**Application servers**

**Application servers > server1**

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

Reports Operations Runtime Configuration

**General Properties**

Name:

Node name:

Run in development mode

Parallel start

**Container Settings**

- [Session management](#)
- [SIP Container Settings](#)
- [Web Container Settings](#)
- [Portlet Container Settings](#)
- [EJB Container Settings](#)
- [EJB container](#)

Cell=wasnd-node01Cell01, Profile=Dmgr

**Application servers**

**Application servers > server1 > EJB container**

Specifies that an EJB container is a component of a J2EE application server that provides runtime services to EJB modules that can be deployed within it.

Configuration

**General Properties**

\* Passivation directory:

Inactive pool cleanup interval:  milliseconds

Default data source JNDI name:

Enable stateful session bean failover using [memory-to-memory replication](#) (Replication domains are defined, but the memory to memory settings have not been selected.)

**Additional Properties**

- [EJB cache settings](#)
- [EJB timer service settings](#)
- [EJB asynchronous method invocation settings](#)

**Buttons:** [Apply](#) [OK](#) [Reset](#) [Cancel](#)

**Application servers****[Application servers](#) > [server1](#) > [EJB container](#) > [Memory-to-memory replication](#)**

Use this page to configure memory-to-memory replication for failure recovery.

**Configuration****General Properties**

Replication domain

Replication mode

**Application servers****[Application servers](#) > [server2](#) > [Session management](#) > [Distributed environment settings](#)**

Use this page to specify how session data is saved in a distributed environment. The SIP container uses only memory-to-memory replication for distributed sessions.

**Configuration****General Properties****Distributed sessions**

- [None](#)
- [Database](#) (Supported for Web container only.)
- [Memory-to-memory replication](#)

**Additional Properties**[Custom tuning parameters](#)

Application servers > [server2](#) > EJB container

Specifies that an EJB container is a component of a J2EE application server that provides runtime services to EJB modules that can be deployed within it.

Configuration

**General Properties**

\* Passivation directory  
\${USER\_INSTALL\_ROOT}/temp

Inactive pool cleanup interval  
30000 milliseconds

Default data source JNDI name  
(none)

Enable stateful session bean failover using [memory-to-memory replication](#)

**Additional Properties**

- [EJB cache settings](#)
- [EJB timer service settings](#)
- [EJB asynchronous method invocation settings](#)

Apply OK Reset Cancel

## **Section 7: Test the application for session failover**

In this section, you test the failover of the session information. Although the PlantsByWebSphere application was not designed to fail over to the shopping cart, you can store content in the session object. After that is done, you stop the application server that is holding session information to demonstrate that the information does indeed fail over to the other cluster member.