

## CHAPTER 14: HIGH AVAILABILITY

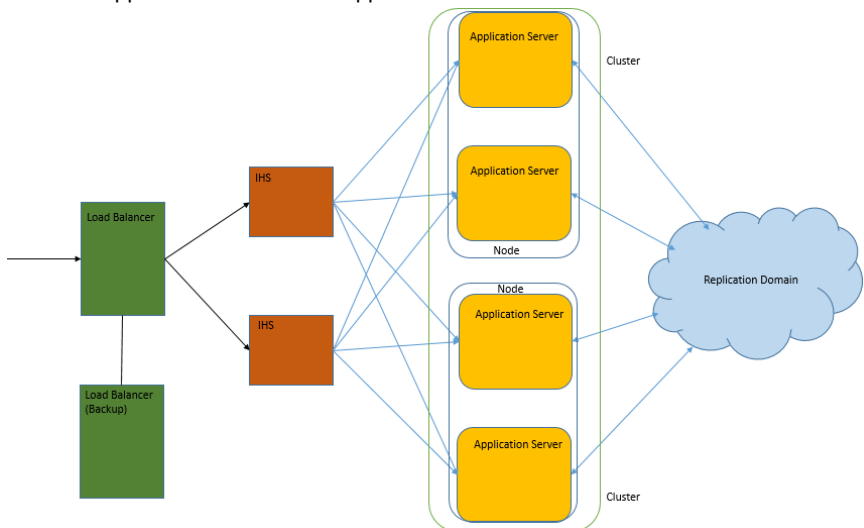
### Theory

High Availability is the ability to tolerate a certain amount of failures without any interruption to the service. It is not a consideration point only for WebSphere Application Server and it can be achieved by only having more redundancy in your architecture. WebSphere Application Server Network Deployment gives you different options to have a highly available environment for your business critical applications.

In order to avoid single point of failure (SPOF), WebSphere Application Server provides vertical and horizontal scaling across different systems. Adding multiple load balancers and web servers can increase the redundancy of the environment.

In a WebSphere Application Server environment, there are multiple areas that are critical to data availability:

- Databases: For read-only data, having multiple copies of synchronized databases behind a load balancer can be a solution. For read/write data access, it can be better to have a hardware cluster for the database node.
- HTTP session state: The session manager creates HTTP sessions and manages the life cycles of HTTP sessions that are associated with the application.
- EJB session state: EJB session state caching that can be replicated across a cluster and persisted to a variety of data stores for failover purposes.
- EJB persistence: It enables to monitor dynamically of deployed applications to add JPA support.



In WebSphere Application Server high availability manager (HA Manager) provides singleton processes to have process high availability. A singleton process can exist in only one location at any given instance, or multiple instances of this function operate independently of one another.

A core groups is the high available part of a WebSphere Application Server cell. Each HA Manager creates connectivity with all the other HA Manager instances in the same core group. HA Manager periodically runs number of tasks in background to provide following services:

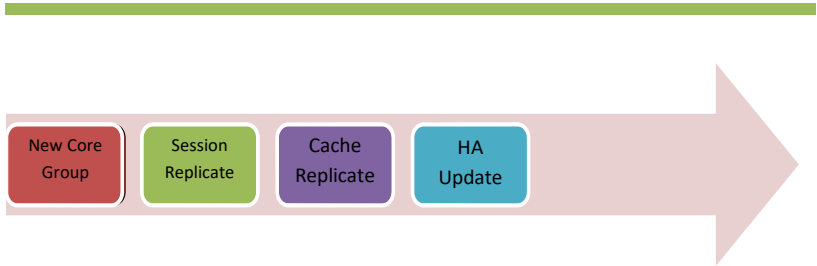
- Memory-to-memory replication is provided by Data Replication Service (DRS) which is part of WebSphere Application Server. It is used to replicate HTTP session data, EJB stateful session, and dynamic cache data among cluster members by using the transport channels to pass information between cluster members.
- Singleton failover is a cluster based service that includes transaction managers for cluster members and the default messaging provider, service integration bus.
- Workload management routing has following types:
  - Routing of the default messaging bus. (SIB)
  - Routing of HTTP requests through WebSphere Application Server proxy server
  - Routing of Web Services Addressing requests through WebSphere Application Server proxy server
  - Routing of Session Initiation Protocol requests
- On-demand configuration routing is used for WebSphere Application Server proxy server routing.

## AIM

In this lab exercise, you will be able to configure high availability and fail over capabilities of WebSphere Application Server. In order to achieve this goal, you need to complete following tasks:

- Create a new core group
- Configure session replication
- Configure cache replication
- Configure high availability application update

## Lab Exercise 14: HIGH AVAILABILITY



1. **Create a new core group**
2. **Configure session replication**
3. **Configure cache replication**
4. **Configure high availability application update**



New Core Group

Session Replicate

Cache Replicate

HA Update

## Task 1: Create a new core group

**Step 1:** Navigate to “Servers>Core Groups>Core Group Settings” and click “New”.

WebSphere Integrated Solutions console screenshot showing the 'Core Groups' page. The left sidebar shows the navigation tree with 'Servers' and 'Core Groups' highlighted. The main content area shows the 'Core Groups' page with a 'New' button highlighted in a red box. The page title is 'Core Groups' and the breadcrumb is 'Servers > Core Groups > Core Group Settings'. The page content includes a description of core groups and a table of existing core groups.

**Core Groups**

Use this page to define a core group. A core group is a grouping of WebSphere(R) Application Server cell processes. A core group can contain stand-alone servers, cluster members, node agents, and the deployment manager. A core group must contain at least one node agent or the deployment manager. A core group must be empty before it can be deleted. Connected core groups are core groups that can communicate with each other. Access point groups must be defined to establish communication between core groups.

**Preferences**

**New** **Delete**

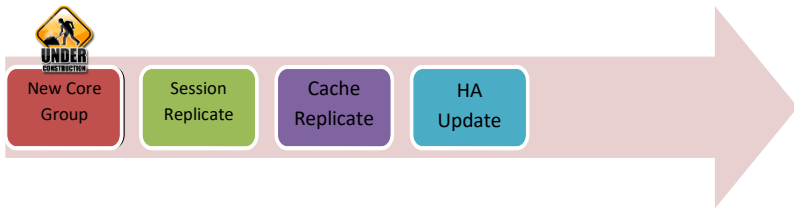
Select	Name	Description	Connected core groups
<input type="checkbox"/>	DefaultCoreGroup	Default Core Group. The default core group cannot be deleted.	

Total 1

**Field help**  
For field help information, select a field label or list marker when the help cursor is displayed.

**Page help**  
[More information about this page](#)

**Command Assistance**  
[View administrative scanning command for last action](#)



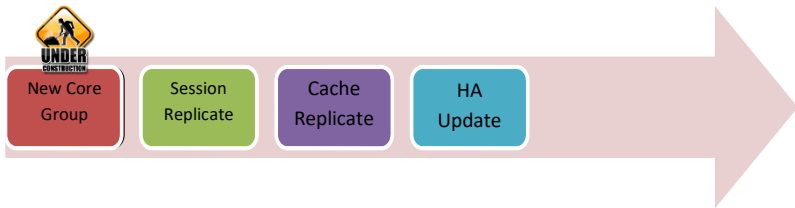
**Step 2:** Enter “WAS\_CoreGroup” as name and click “OK”.

The screenshot shows the 'Core Groups > New...' configuration page in the WebSphere Integration Developer console. The 'General Properties' section is highlighted with a red box, and the 'Name' field is set to 'WAS\_CoreGroup'. The 'Description' field contains 'A non-default core group'. The 'Number of coordinators' is set to '1', and the 'Transport memory size' is set to '100 megabytes'. The 'Transport type' is set to 'Channel framework', and the 'Transport chain' is set to 'DCS'. The 'OK' button is highlighted with a red box.

**Step 3:** Click “Save” to write changes to the master configuration.

The screenshot shows the 'Core Groups' page in the WebSphere Integration Developer console. A message box is highlighted with a red box, indicating that changes have been made to the local configuration and can be saved directly to the master configuration. The message box contains the following text: "Changes have been made to your local configuration. You can: • Save directly to the master configuration. • Review changes before saving or discarding." Below the message box, there is a section for 'Core Groups' with a table listing the resources.

Select	Name	Description	Connected core groups
<input checked="" type="checkbox"/>	DefaultCoreGroup	Default Core Group. The default core group cannot be deleted.	WAS_CoreGroup
<input type="checkbox"/>	WAS_CoreGroup	A non-default core group	DefaultCoreGroup



Step 4: Click on “DafaultCoreGroup”.

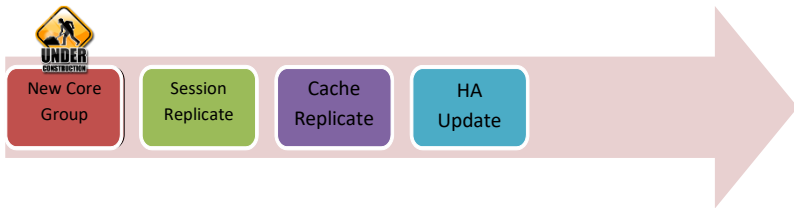
The screenshot shows the WebSphere Integrated Solutions console. The left-hand navigation pane is expanded to "Core Groups". The main content area displays the "Core Groups" page. A table lists the core groups:

Select	Name	Description	Connected core groups
<input type="checkbox"/>	DefaultCoreGroup	Default Core Group. The default core group cannot be deleted.	WAS_CoreGroup
<input type="checkbox"/>	WAS_CoreGroup	A non-default core group	DefaultCoreGroup

The "DefaultCoreGroup" row is highlighted with a red box. Below the table, it says "Total 2".

Step 5: Click “Core group servers” under “Additional Properties”.

The screenshot shows the WebSphere Integrated Solutions console. The left-hand navigation pane is expanded to "Core Groups". The main content area displays the "Core Groups" page. The "DefaultCoreGroup" row is selected. The "Additional Properties" tab is selected, and the "Core group servers" link is highlighted with a red box.



**Step 6:** Select application servers to move to the new core group and click “Move”.

Select	Name	Node	Version	Type	Cluster Name
<input checked="" type="checkbox"/>	App_Server01	wasv9Node01	ND 9.0.0.0	Application Server	WAS_CLUSTER
<input checked="" type="checkbox"/>	App_Server02	wasv9Node02	ND 9.0.0.0	Application Server	WAS_CLUSTER
<input type="checkbox"/>	CDR	wasv9Node02	ND 9.0.0.0	Custom (ONDEMAND, ROUTER)	

**Step 7:** Select the target core group under “To core group” and click “OK”.

**Configuration**

**General Properties**

**Move selected servers**

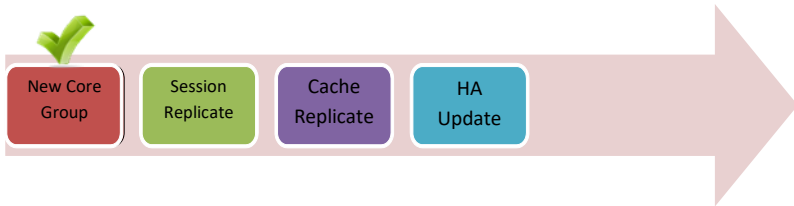
Cluster name WAS\_CLUSTER App\_Server01 App\_Server02

From core group: DefaultCoreGroup

To core group: **WAS\_CoreGroup**

App OK Reset Cancel





**Step 8:** Click “Save” to write changes to the master configuration file.

**Messages**

The servers were successfully moved to core group WAS\_CoreGroup.  
 Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

An option to synchronize the configuration across multiple nodes after saving can be enabled in [Preferences](#).  
 The server may need to be restarted for these changes to take effect.

**Core Groups - DefaultCoreGroup - Core group servers**

Use this page to view and manage the servers that belong to a core group. A core group server can be an application server, a deployment manager, or a node agent that is a member of a high availability core group.

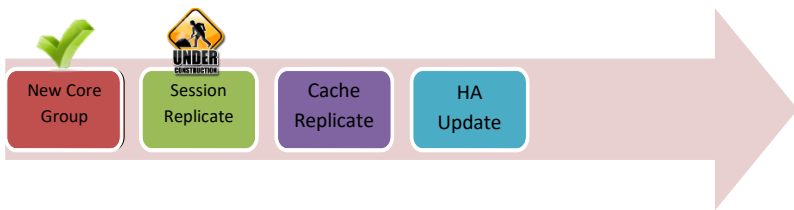
**Preferences**

Move...

Select	Name	Node	Version	Type	Cluster Name
<input type="checkbox"/>	CDR	wasv90Node02	ND 9.0.0.0	Custom (ONDEMAND_ROUTER)	

You can administer the following resources:

**Task 1 is complete!**



## Task 2: Configure session replication

**Step 1:** Navigate to “Servers>Server Types>WebSphere application servers” and click on the application server name “App\_Server01”.

**WebSphere Integrated Solutions console**

View: All tasks

Server Types

- WebSphere application servers
- Liberty profile servers
- WebSphere proxy servers
- On Demand Routers
- Web servers
- WebSphere Application Server Community Edition servers
- Servlet servers
- WebSphere MQ servers
- Web services
- Apache Tomcat servers
- WSA WebLogic servers
- Blaze servers
- External WebSphere Application Servers
- Apache servers
- Custom HTTP servers

**Application servers**

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.

Preferences

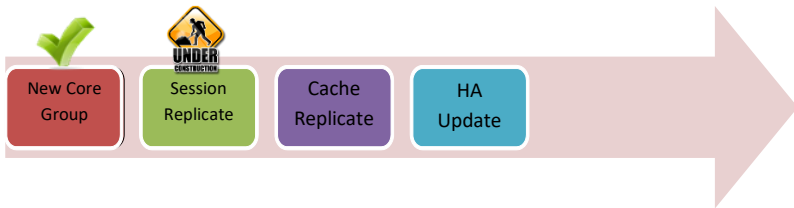
Select	Name	Node	Host Name	Version	Cluster Name	Status
<input checked="" type="checkbox"/>	App_Server01	wasv90Node01	wasv90	ND 9.0.0.0	WAS_CLUSTER	➔
<input type="checkbox"/>	App_Server02	wasv90Node02	wasv90	ND 9.0.0.0	WAS_CLUSTER	➔
<input type="checkbox"/>	server1	wasv90Node01	wasv90	ND 9.0.0.0		➔
<input type="checkbox"/>	server2	wasv90Node02	wasv90	ND 9.0.0.0		➔

Total 4

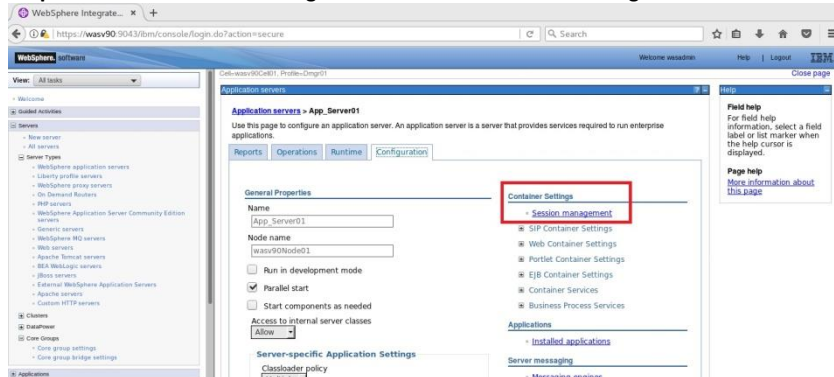
**Field help**  
For field help information, select a field label or list marker when the help cursor is displayed.

**Page help**  
More information about this page

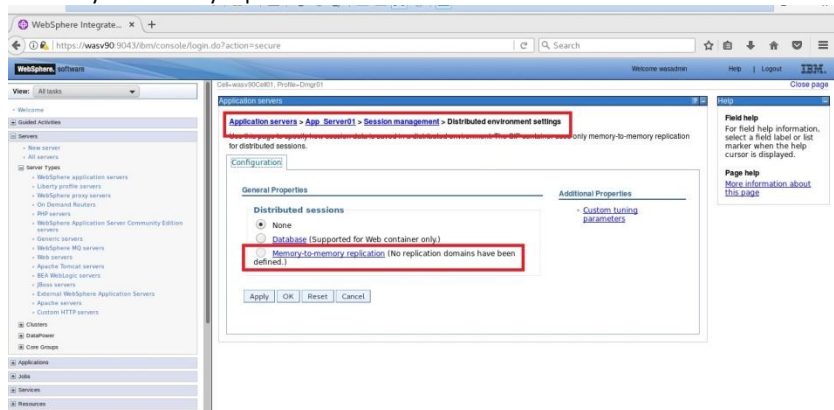
**Command Assistance**  
View administrative scripting commands for list action

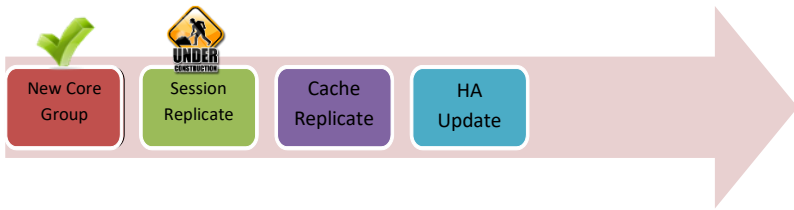


**Step 2:** Click on “Session management” under “Container Settings”.

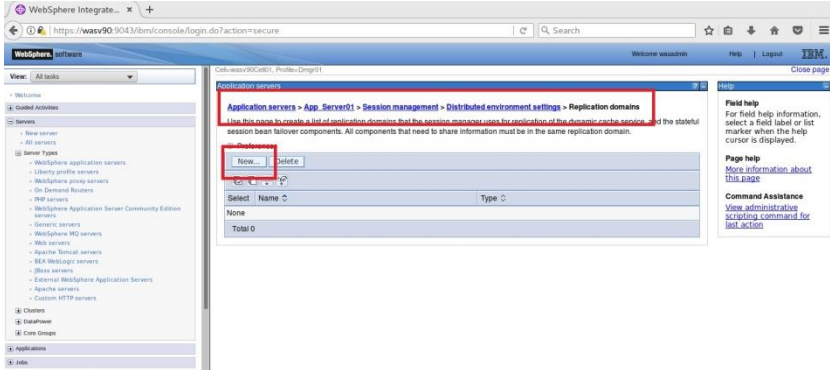


**Step 3:** Under Additional Properties, Distributed Environment Setting. Then click on “Memory-to-memory replication”.

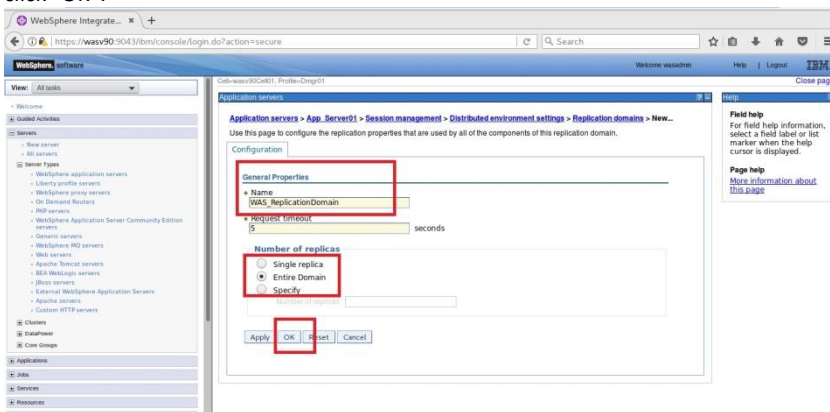


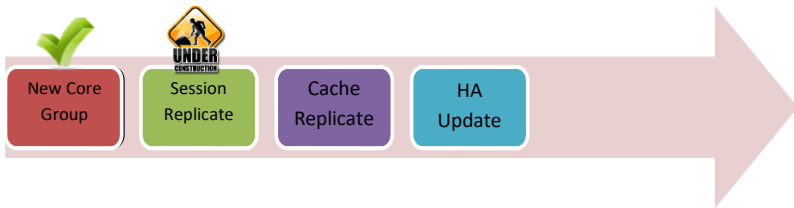


**Step 4:** Since we don't have any replication domain, click "New" to create one.

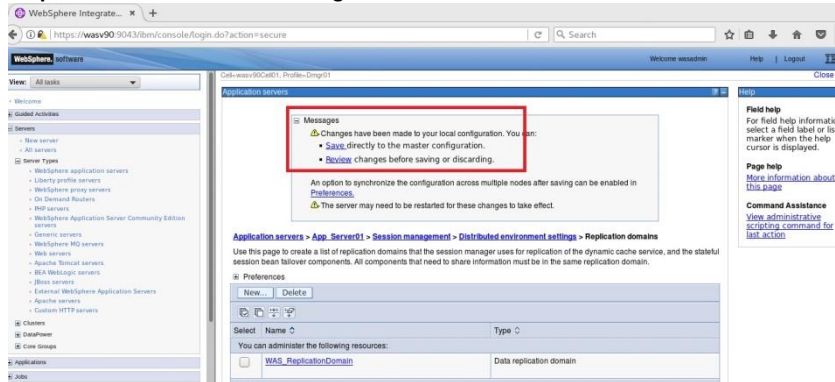


**Step 5:** Enter "WAS\_ReplicationDomain" as name and select "Entire Domain" then click "OK".

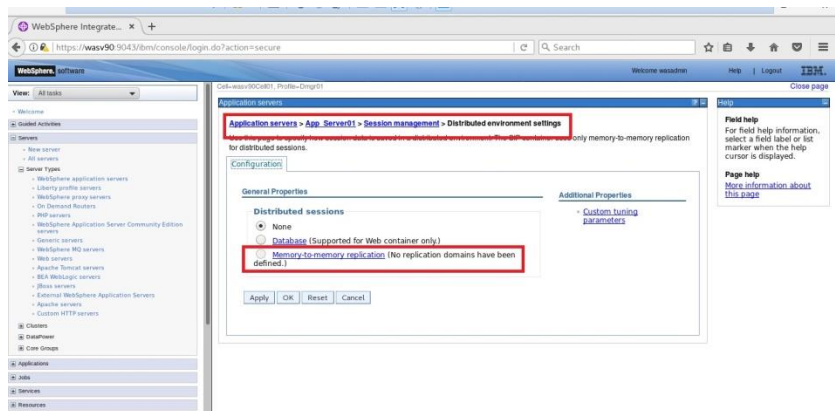


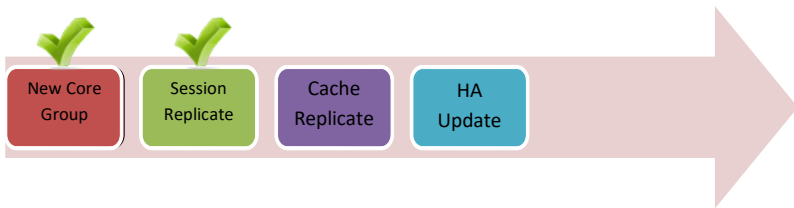


**Step 6:** Click “Save” to write changes.

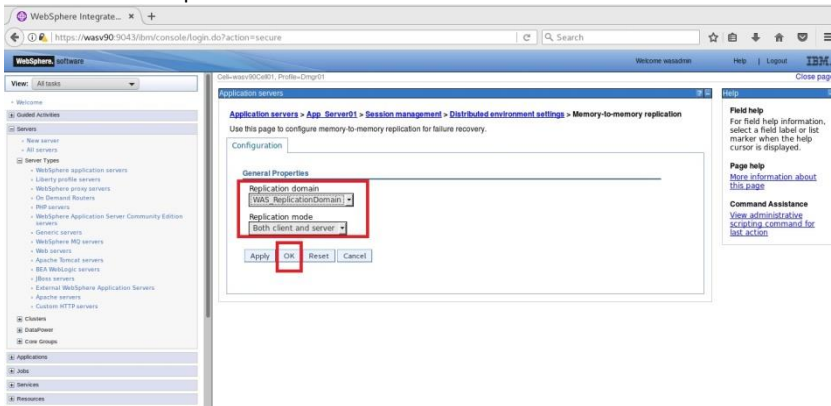


**Step 7:** Click on “Memory-to-memory replication”.

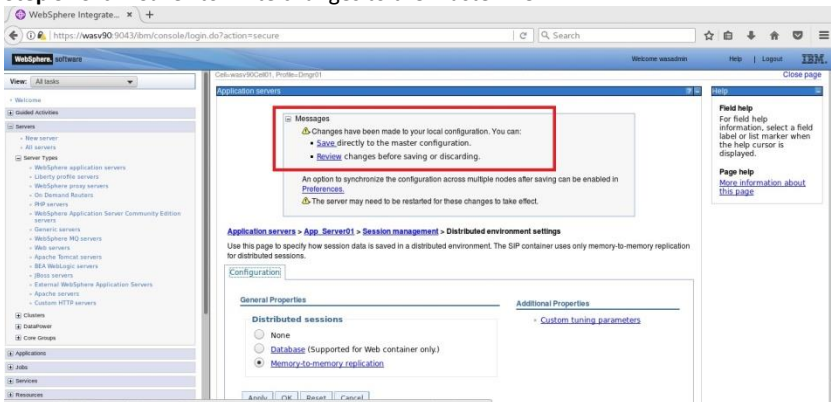




**Step 8:** Select “WAS\_ReplicationDomain” as “Replication domain” and “Both client and server” as “Replication mode” and click “OK”.

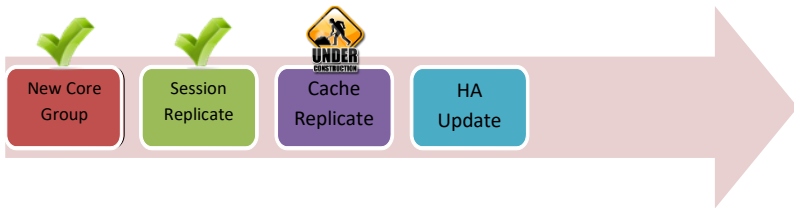


**Step 9:** Click “Save” to write changes to the master file.



**Step 10:** Repeat the steps 7 to 9 for the application server “App\_Server02”.

**Task 2 is complete!**



### Task 3: Configure cache replication

**Step 1:** Navigate to “Servers>Server Types>WebSphere application servers” and click on the application server name “App\_Server01”.

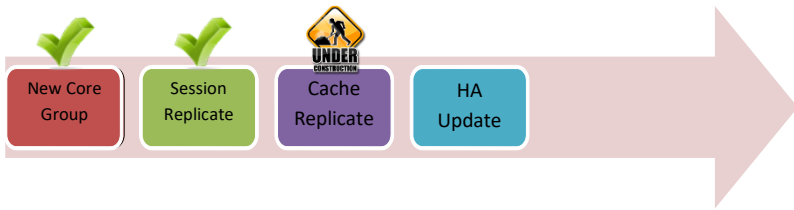
WebSphere Integrated Solutions console

Application servers

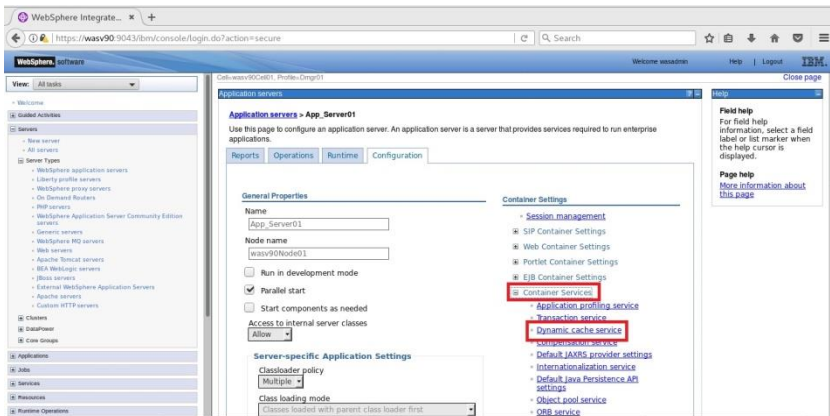
Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.

Select	Name	Node	Host Name	Version	Cluster Name	Status
<input checked="" type="checkbox"/>	App_Server01	wasv90Node01	wasv90	ND 9.0.0.0	WAS_CLUSTER	→
<input type="checkbox"/>	App_Server02	wasv90Node02	wasv90	ND 9.0.0.0	WAS_CLUSTER	→
<input type="checkbox"/>	server1	wasv90Node01	wasv90	ND 9.0.0.0		→
<input type="checkbox"/>	server2	wasv90Node02	wasv90	ND 9.0.0.0		→

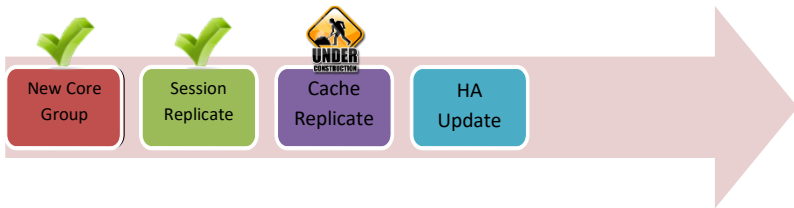
Total: 4



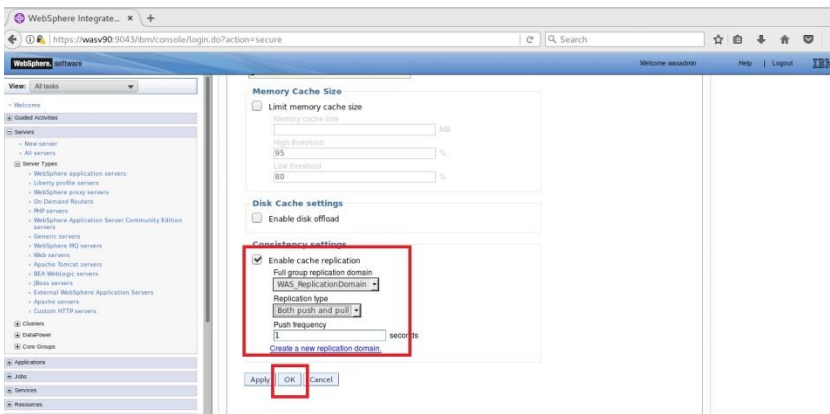
**Step 2:** Click “Dynamic cache service” under “Container Settings>Container Services”.

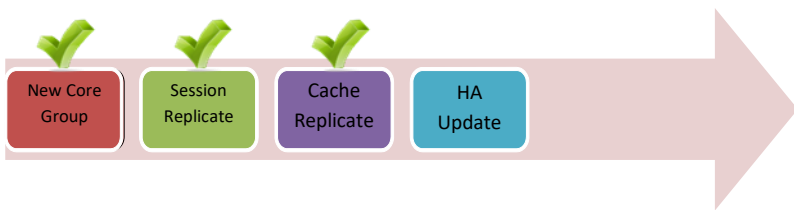




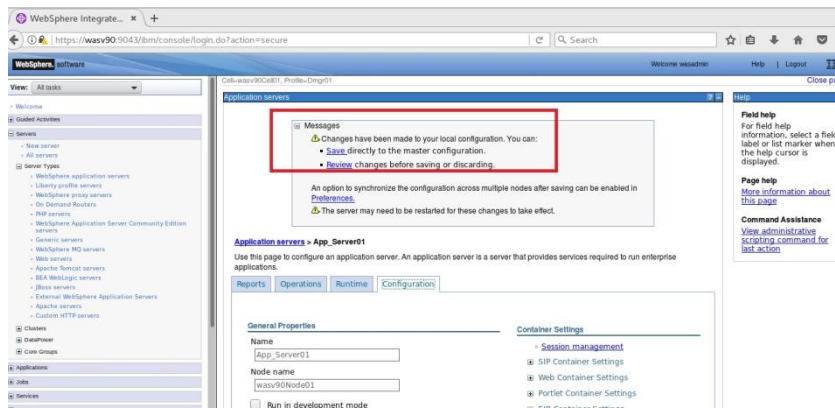


**Step 3:** Check “Enable cache replication” and select “WAS\_ReplicationDomain” for “Full group replication name”, “Both push and pull” as “Replication type” and “1” for “Push frequency” then click “OK”.



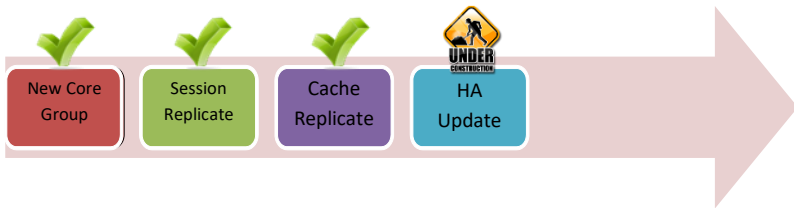


**Step 4:** Click “Save” to write changes directly to the master configuration file.



**Step 5:** Repeat the steps 1 to 4 for the application server “App\_Server02”.

**Task 3 is complete!**



## Task 4: Configure High Availability Application Update

**Step 1:** Navigate to “System administration>Node agents” and click on “nodeagent” for the first node.

**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.

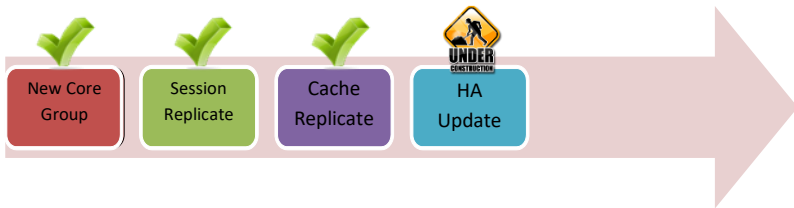
**Preferences**

Stop | Restart | Restart all Servers on Node

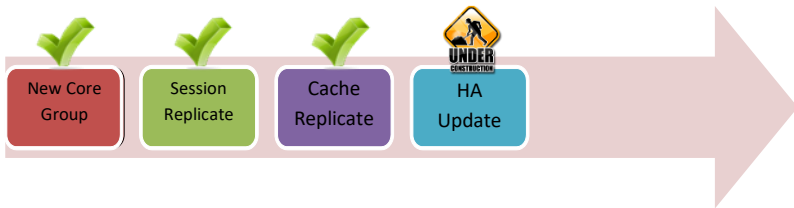
Select: Name Node Host Name Version Status

You can administer the following resources:

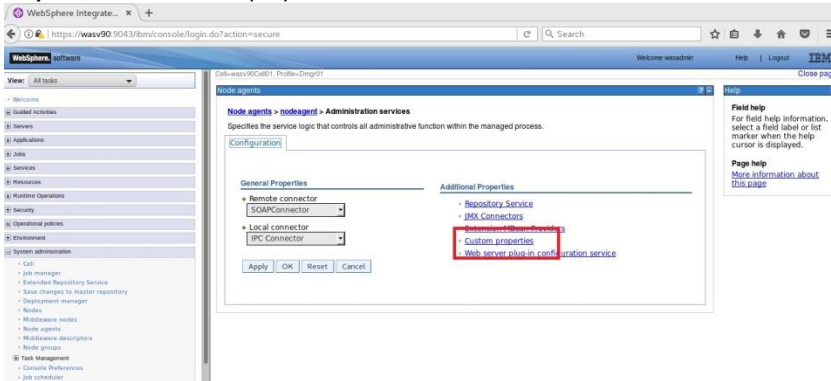
Select	Name	Node	Host Name	Version	Status
<input checked="" type="checkbox"/>	nodeagent	wasv90Node01	wasv90	ND 9.0.0.0	➔
<input type="checkbox"/>	nodeagent	wasv90Node02	wasv90	ND 9.0.0.0	➔
Total 2					



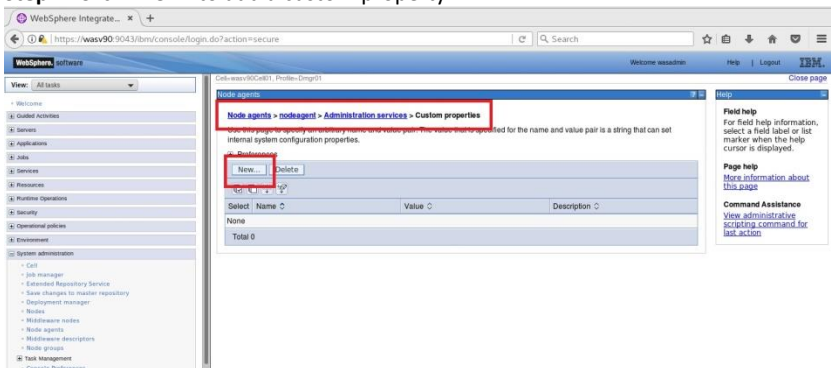
**Step 2:** Navigate to “Additional Properties>Administration services”.

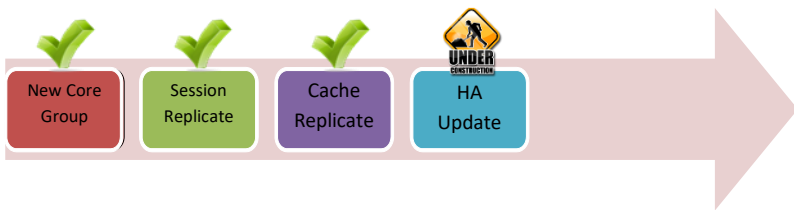


### Step 3: Click on “Custom properties”.



### Step 4: Click “New” to add a custom property.



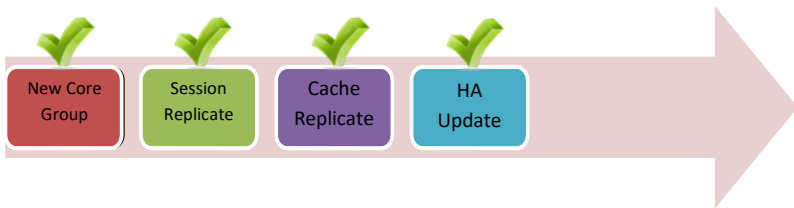


**Step 5:** Enter “*com.ibm.websphere.zos.rollout.pauseresume*” as “Name” and “true” as “Value” then click “OK”.

The screenshot shows the WebSphere Integration console interface. The left sidebar contains a navigation tree with categories like 'Welcome', 'Guided Activities', 'Servers', 'Applications', 'Jobs', 'Services', 'Resources', 'Runtime Operations', 'Security', 'Operational policies', 'Environment', 'System administration', and 'Task Management'. The main content area is titled 'Node agents > nodeagent > Administration services > Custom properties > New...'. It includes a description: 'Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.' Below this is a 'Configuration' section with a 'General Properties' tab. The 'Name' field contains 'com.ibm.websphere.zos.rollout.pauseresume' and the 'Value' field contains 'true'. The 'OK' button is highlighted with a red box.

**Step 6:** Add another custom property that has name “*com.ibm.websphere.zos.mvsservices.enable*” and value “true” then click “OK”.

The screenshot shows the WebSphere Integration console interface, similar to the previous one. The main content area is titled 'Node agents > nodeagent > Administration services > Custom properties > New...'. The 'Name' field contains 'com.ibm.websphere.zos.mvsservices.enable' and the 'Value' field contains 'true'. The 'OK' button is highlighted with a red box.



**Step 7:** Repeat the steps 1 to 6 for the other node.

WebSphere Integrated Solutions console

Node agents

Node agents > nodeagent > Administration services > Custom properties

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.

Preferences

New... Delete

Select	Name	Value	Description
<input type="checkbox"/>	com.ibm.websphere.zos.mvservices.enable	true	
<input type="checkbox"/>	com.ibm.websphere.zos.rtfoud.pauseresume	true	

Total 2

**Task 4 is complete!**

## **SUMMARY**

High Availability is the ability to tolerate a certain amount of failures without any interruption to the service. WebSphere Application Server Network Deployment gives you different options to have a highly available environment for your business critical applications. In order to avoid single point of failure (SPOF), WebSphere Application Server provides vertical and horizontal scaling across different systems. WebSphere Application Server high availability manager (HA Manager) provides singleton processes to have process high availability.



## REFERENCES

- [http://www-01.ibm.com/support/knowledgecenter/SSZJPZ\\_8.5.0/com.ibm.swg.im.iis.production.iisinfsv.ha.install.doc/topics/wsisinst\\_ha\\_wasclusteringtop.html?lang=en](http://www-01.ibm.com/support/knowledgecenter/SSZJPZ_8.5.0/com.ibm.swg.im.iis.production.iisinfsv.ha.install.doc/topics/wsisinst_ha_wasclusteringtop.html?lang=en)
- [http://publib.boulder.ibm.com/infocenter/wsdoc400/v6r0/index.jsp?topic=/com.ibm.websphere.iseries.doc/info/ae/ae/cprs\\_memory2memory.html](http://publib.boulder.ibm.com/infocenter/wsdoc400/v6r0/index.jsp?topic=/com.ibm.websphere.iseries.doc/info/ae/ae/cprs_memory2memory.html)
- <http://pic.dhe.ibm.com/infocenter/wxsinfo/v8r5/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.doc%2Ftxsdyncache.html>

## INDEX

core group .....	455
data availability.....	454
DRS.....	455
EJB persistence .....	454
EJB session state .....	454
High Availability .....	454
high availability manager .....	455
HTTP session state .....	454
load balancers.....	454
Memory-to-memory replication.....	455
redundancy .....	454
Singleton .....	455
SPOF.....	454
Workload management .....	455



