CHAPTER 14: HIGH AVAILABILITY

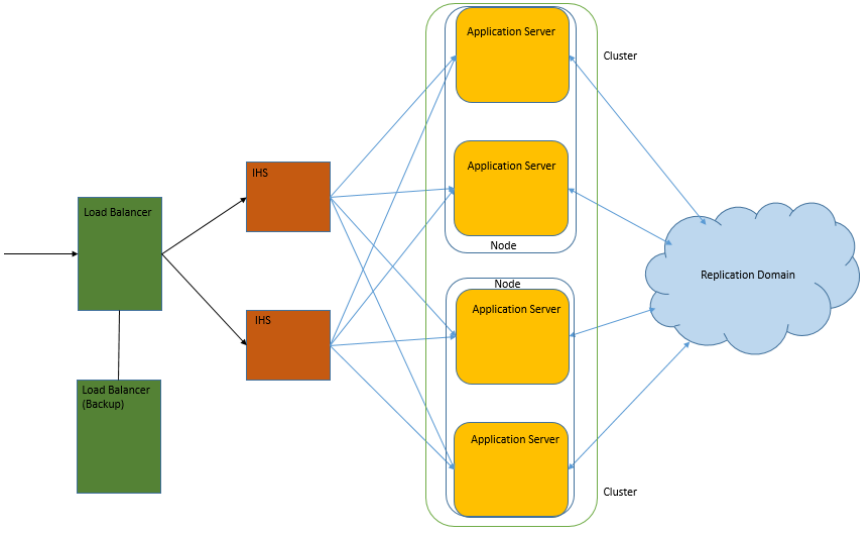
Theory

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:

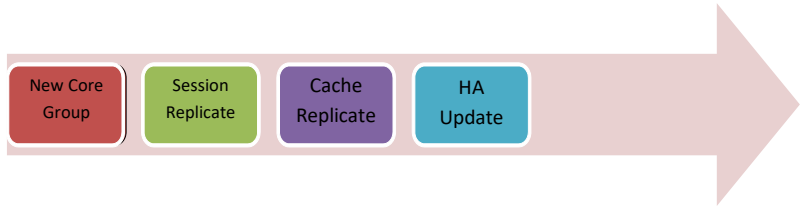
- Memory-to-memory replication is provided by Data Replication Service (DRS) which is part of WebSphere Application Server. It is used to replicate HTTP session data, EJB stateful session, and dynamic cache data among cluster members by using the transport channels to pass information between cluster 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

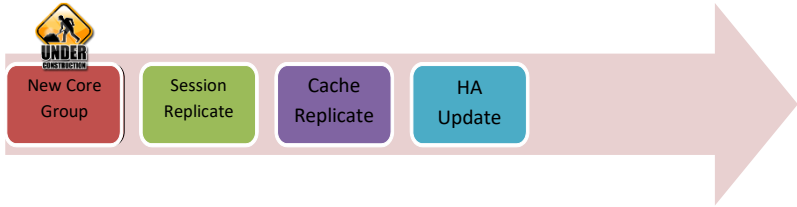
In this lab exercise, you will be able to configure high availability and fail over capabilities of WebSphere Application Server. In order to achieve this goal, you need to complete following tasks:

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

Lab Exercise 14: HIGH AVAILABILITY

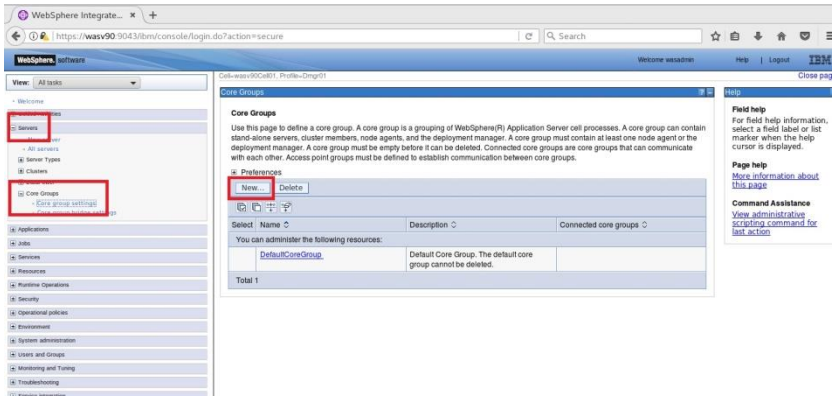


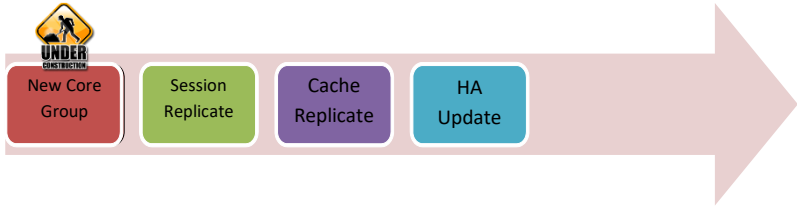
1. **Create a new core group**
2. **Configure session replication**
3. **Configure cache replication**
4. **Configure high availability application update**



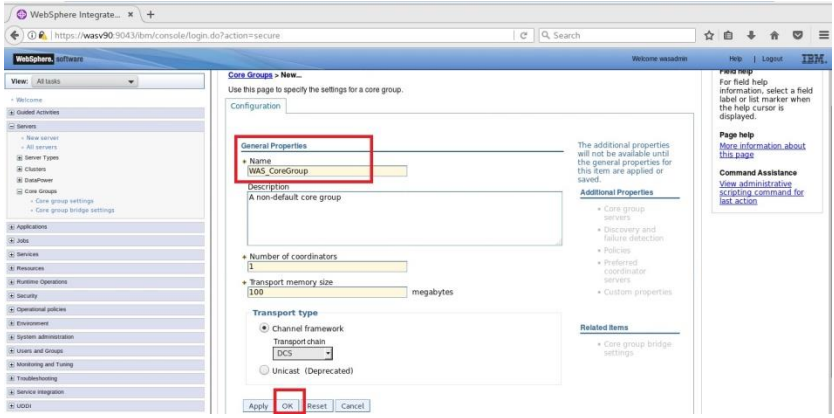
Task 1: Create a new core group

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

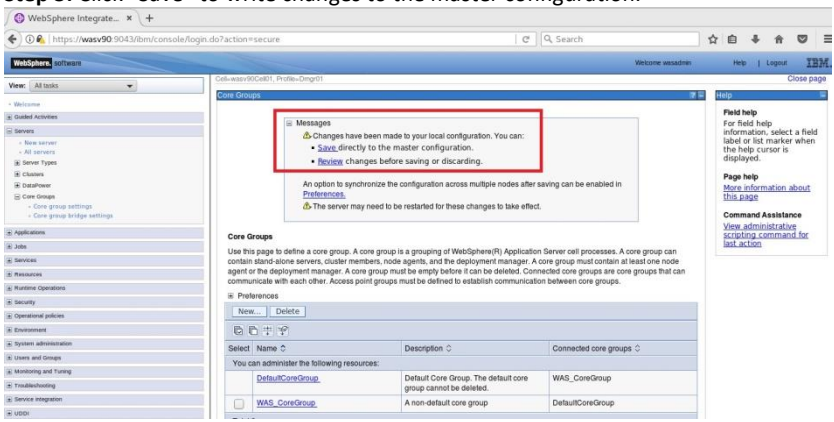


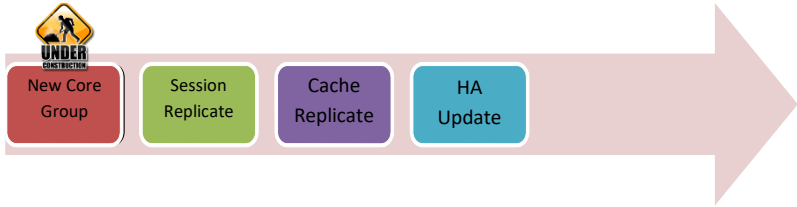


Step 2: Enter “WAS_CoreGroup” as name and click “OK”.

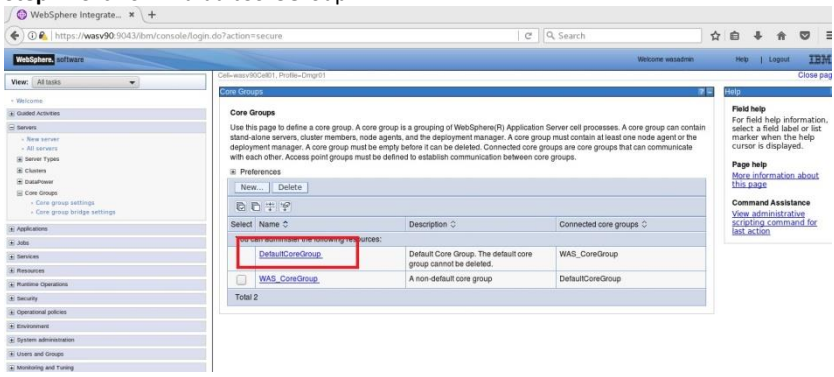


Step 3: Click “Save” to write changes to the master configuration.

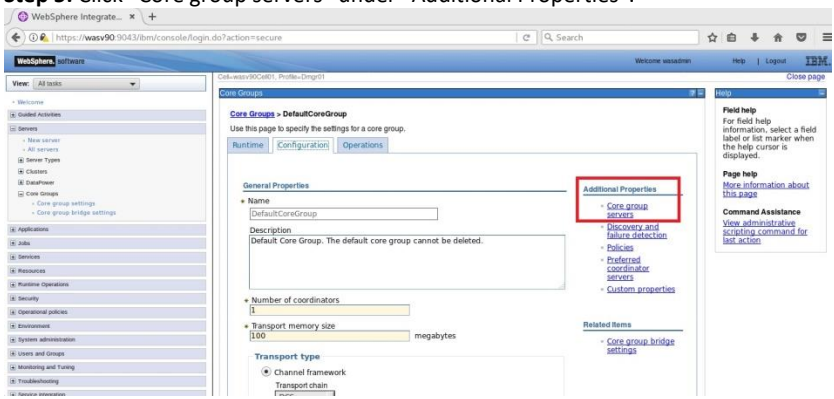


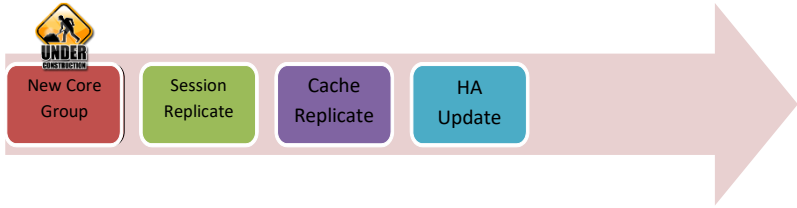


Step 4: Click on “DafaultCoreGroup”.

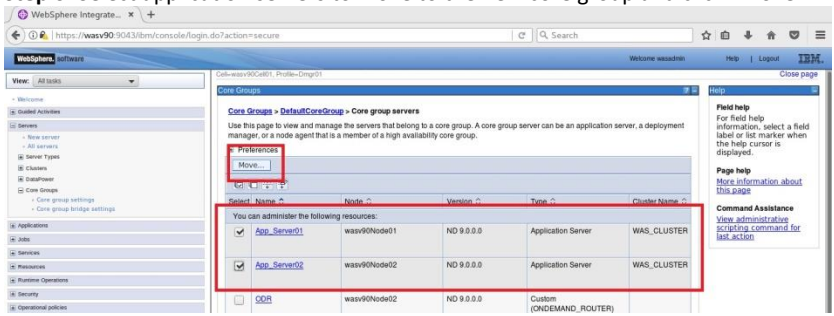


Step 5: Click “Core group servers” under “Additional Properties”.

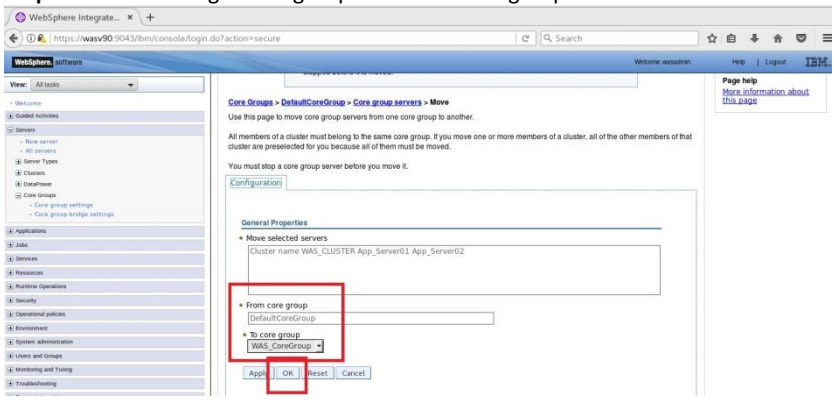


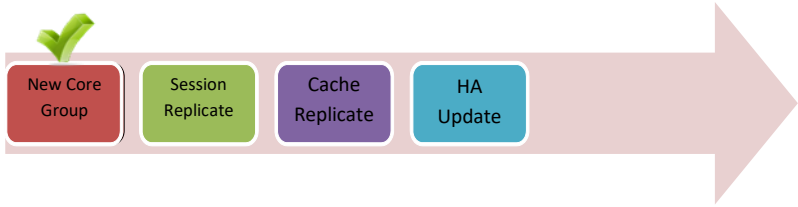


Step 6: Select application servers to move to the new core group and click “Move”.

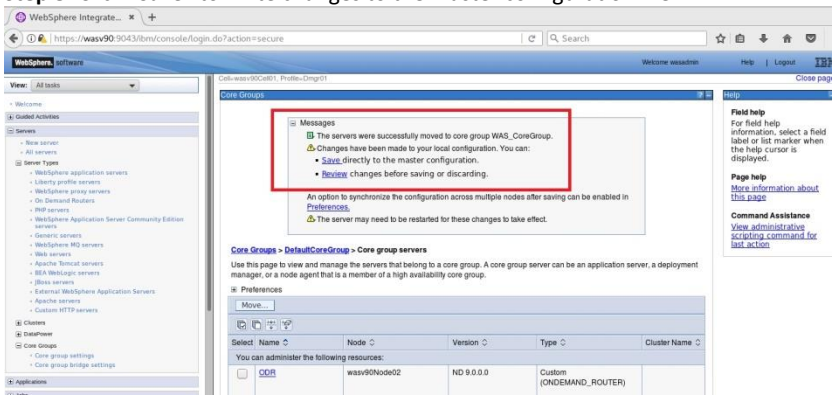


Step 7: Select the target core group under “To core group” and click “OK”.

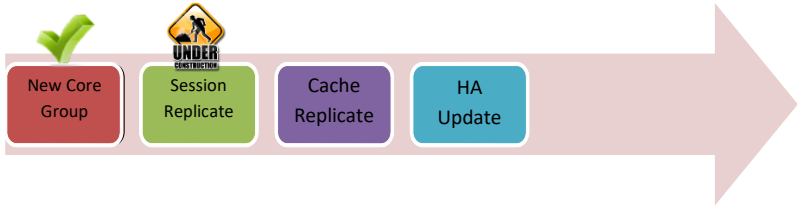




Step 8: Click “Save” to write changes to the master configuration file.

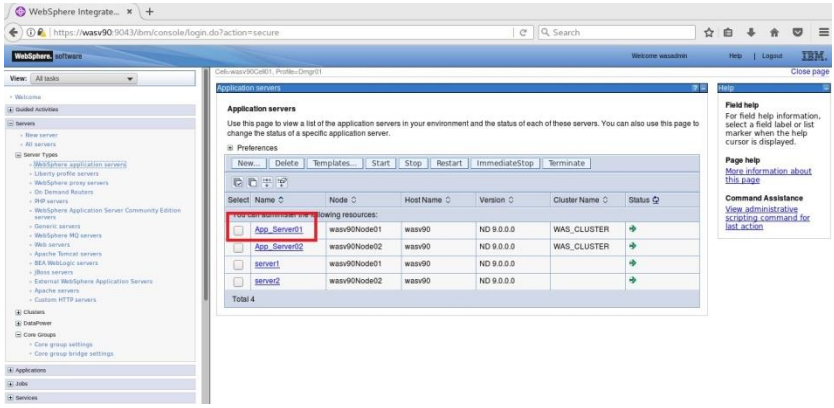


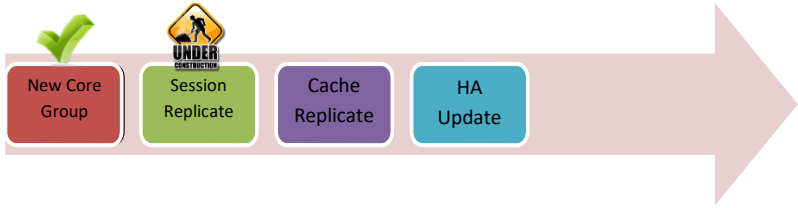
Task 1 is complete!



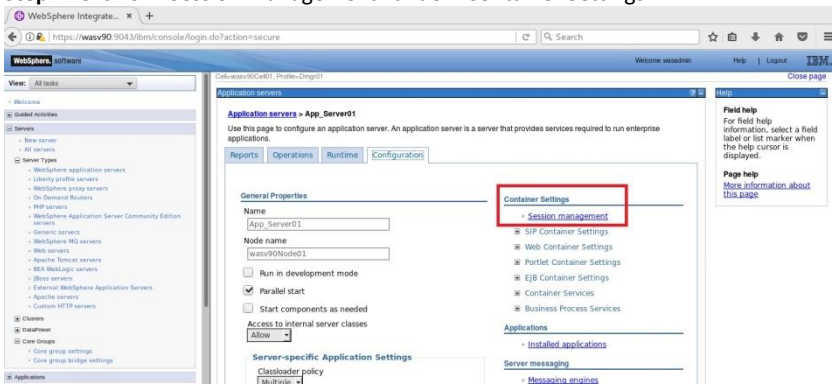
Task 2: Configure session replication

Step 1: Navigate to “Servers>Server Types>WebSphere application servers” and click on the application server name “App_Server01”.

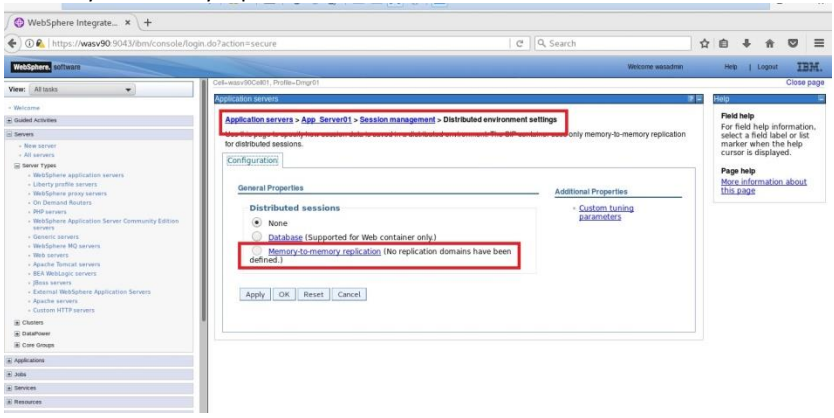


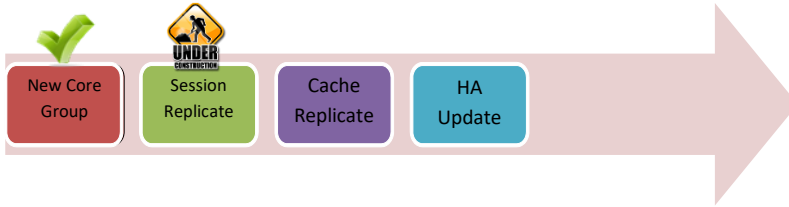


Step 2: Click on “Session management” under “Container Settings”.

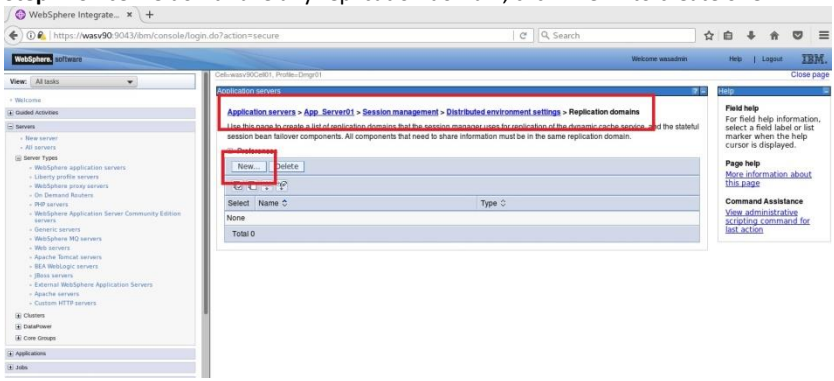


Step 3: Under Additional Properties, Distributed Environment Setting. Then click on “Memory-to-memory replication”.

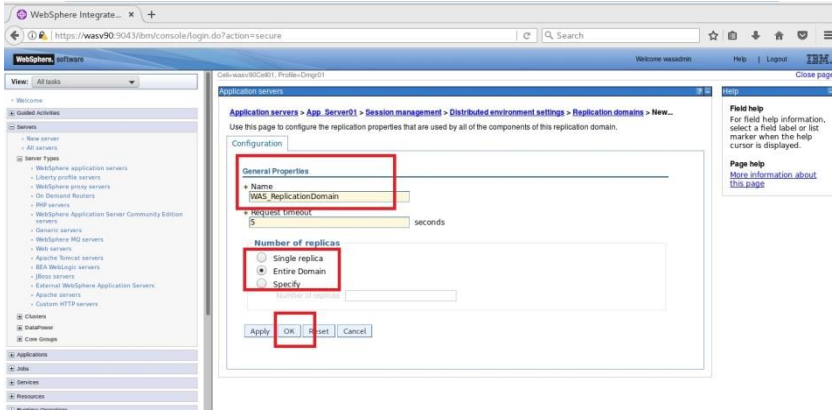


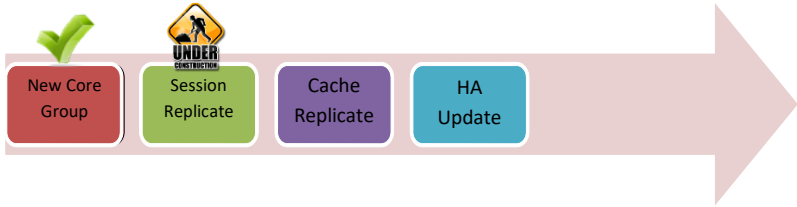


Step 4: Since we don't have any replication domain, click "New" to create one.

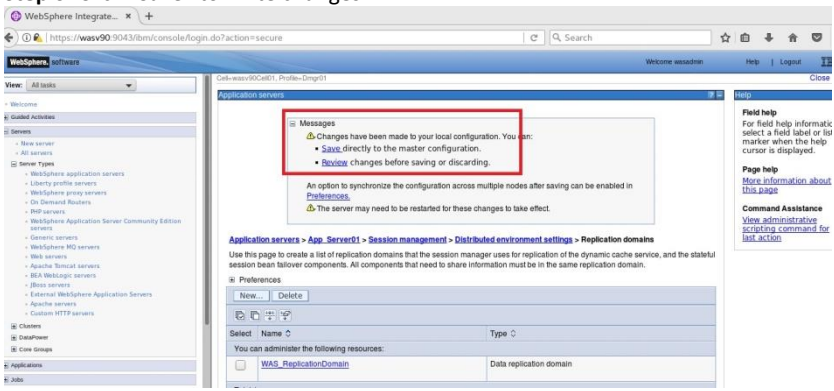


Step 5: Enter "WAS_ReplicationDomain" as name and select "Entire Domain" then click "OK".

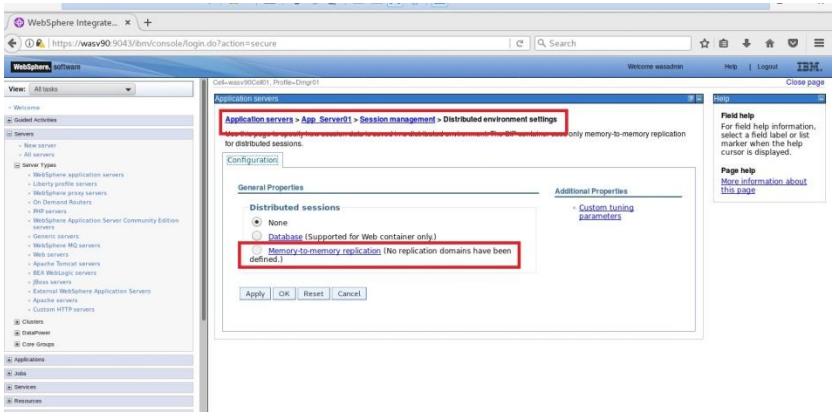


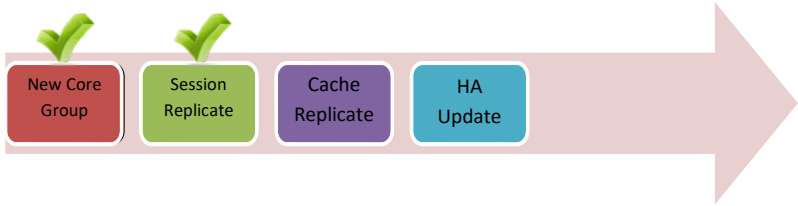


Step 6: Click “Save” to write changes.

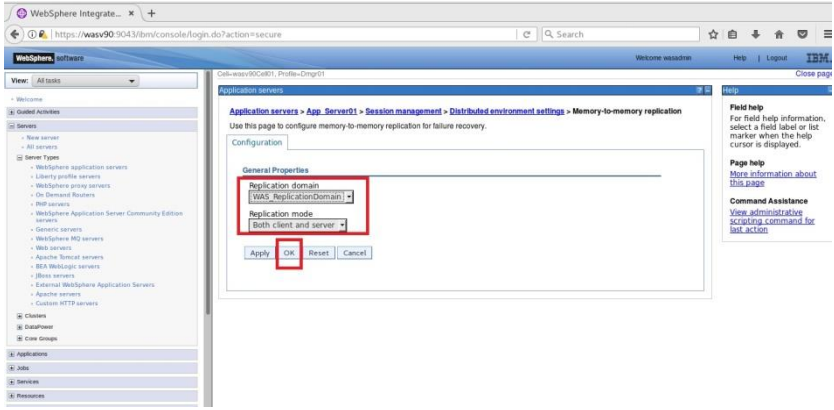


Step 7: Click on “Memory-to-memory replication”.

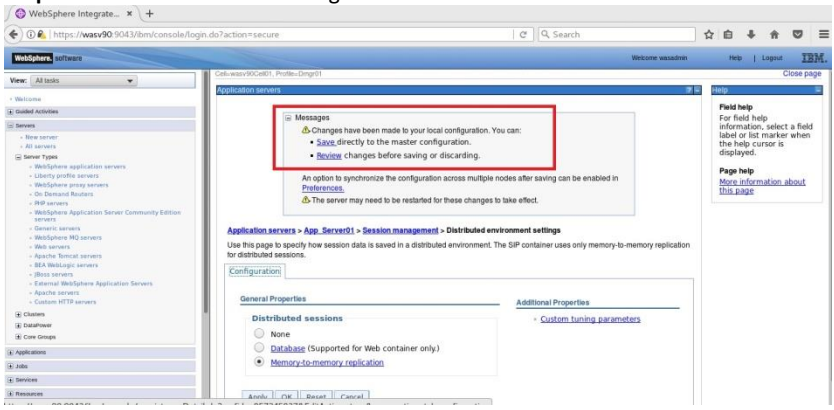




Step 8: Select “WAS_ReplicationDomain” as “Replication domain” and “Both client and server” as “Replication mode” and click “OK”.

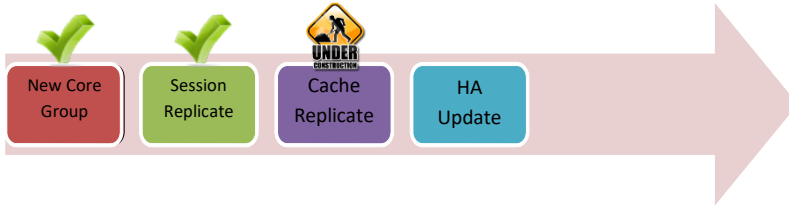


Step 9: Click “Save” to write changes to the master file.



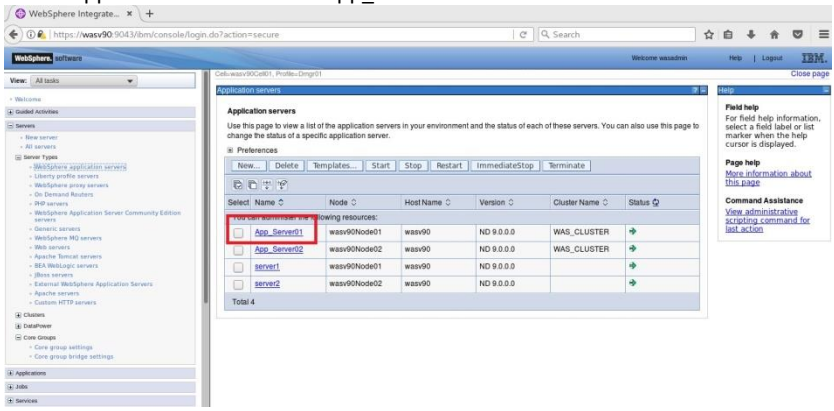
Step 10: Repeat the steps 7 to 9 for the application server “App_Server02”.

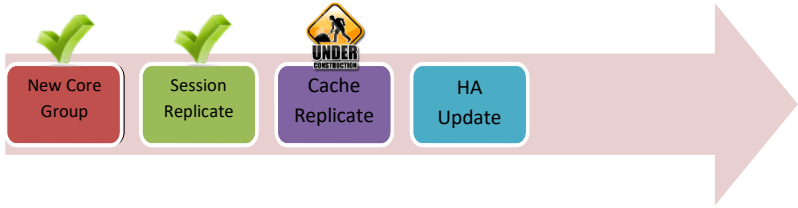
Task 2 is complete!



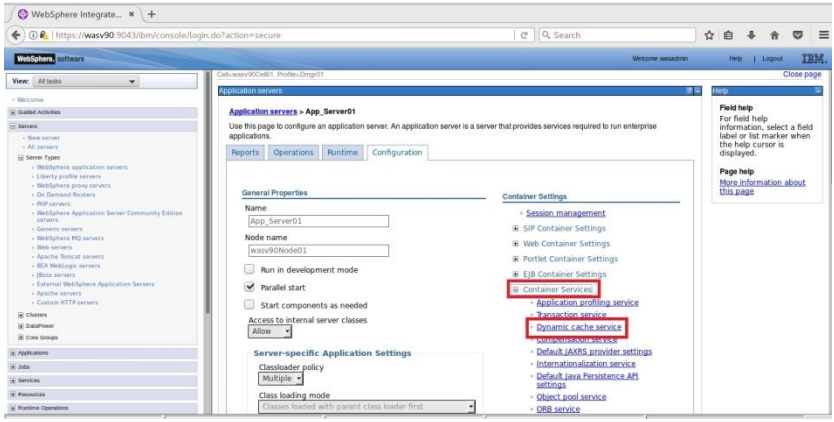
Task 3: Configure cache replication

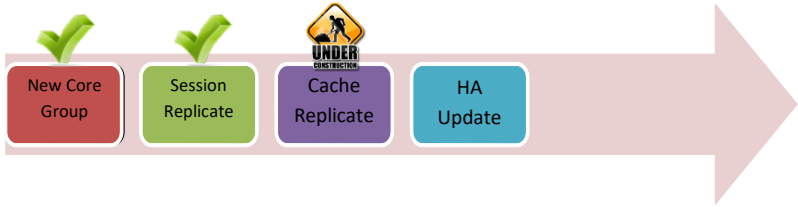
Step 1: Navigate to “Servers>Server Types>WebSphere application servers” and click on the application server name “App_Server01”.



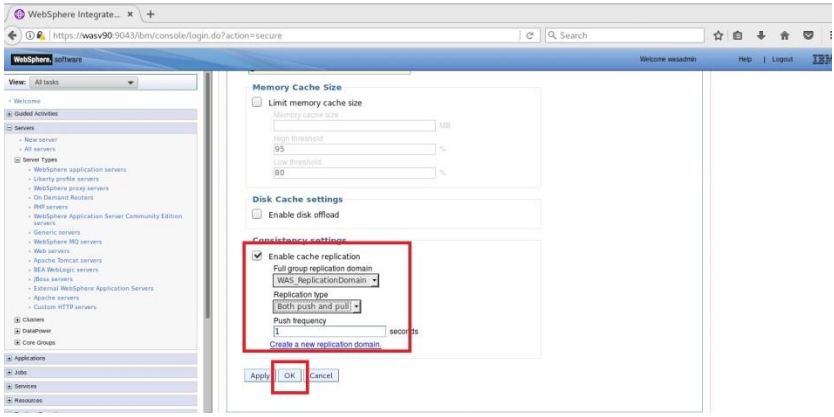


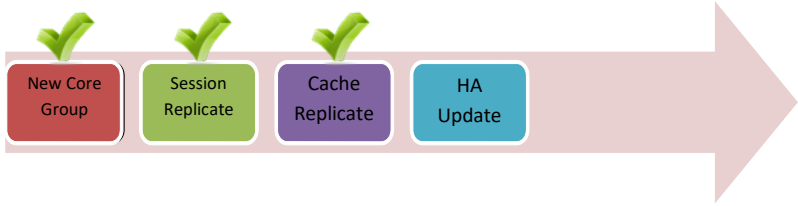
Step 2: Click “Dynamic cache service” under “Container Settings>Container Services”.



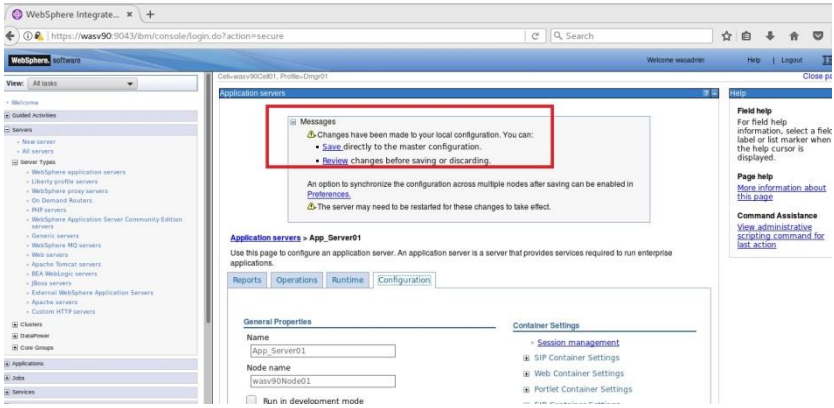


Step 3: Check “Enable cache replication” and select “WAS_ReplicationDomain” for “Full group replication name”, “Both push and pull” as “Replication type” and “1” for “Push frequency” then click “OK”.



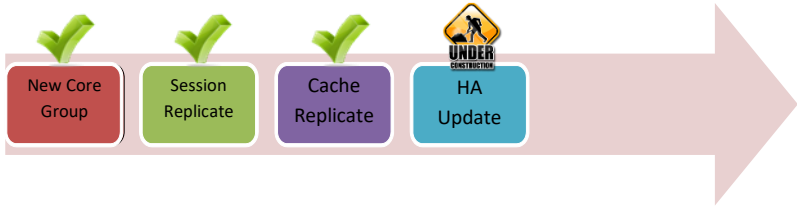


Step 4: Click “Save” to write changes directly to the master configuration file.



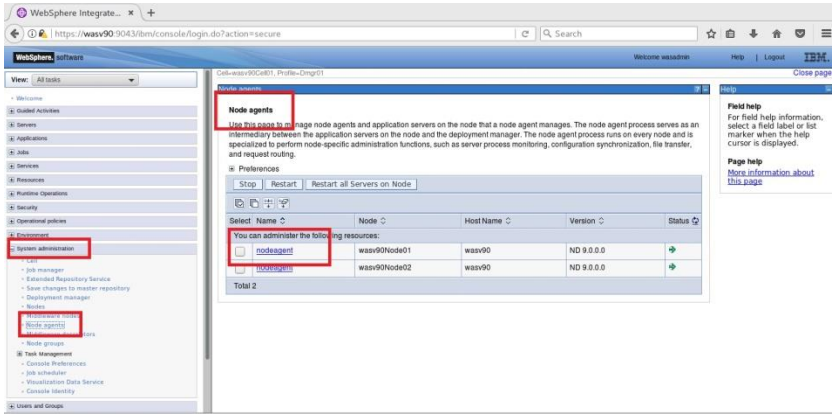
Step 5: Repeat the steps 1 to 4 for the application server “App_Server02”.

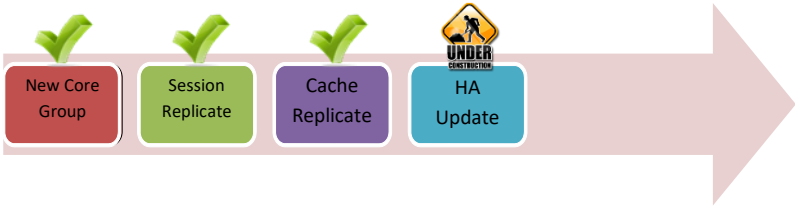
Task 3 is complete!



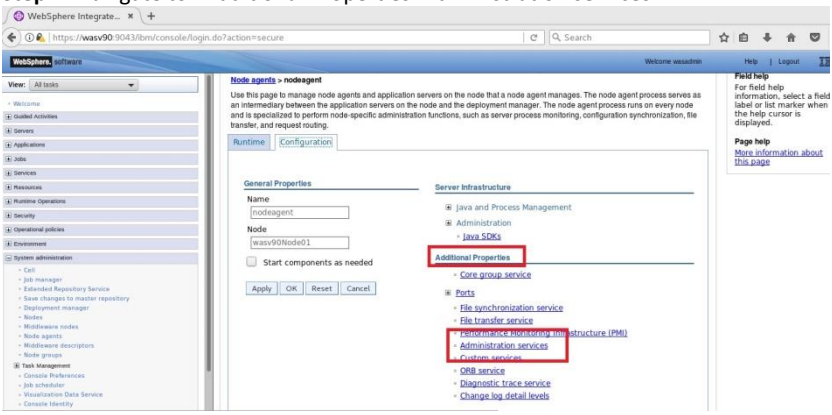
Task 4: Configure High Availability Application Update

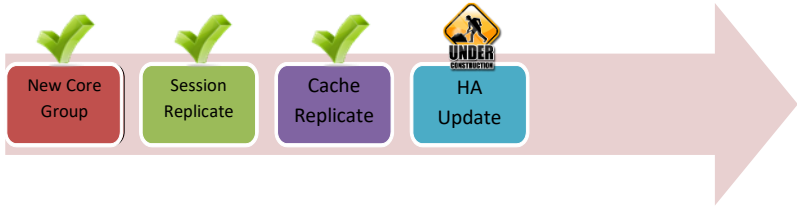
Step 1: Navigate to “System administration>Node agents” and click on “nodeagent” for the first node.



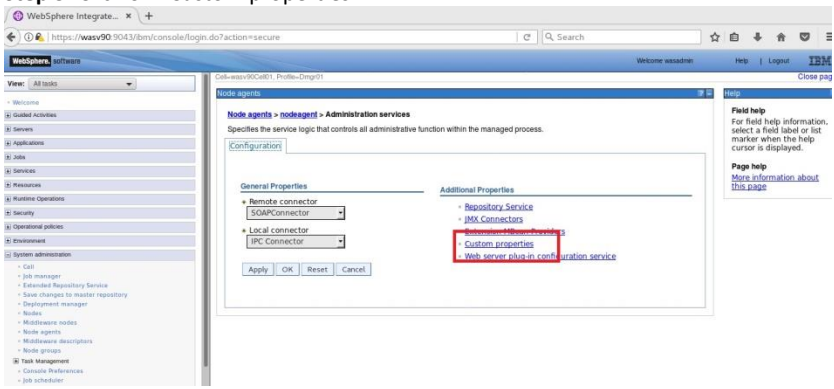


Step 2: Navigate to “Additional Properties>Administration services”.

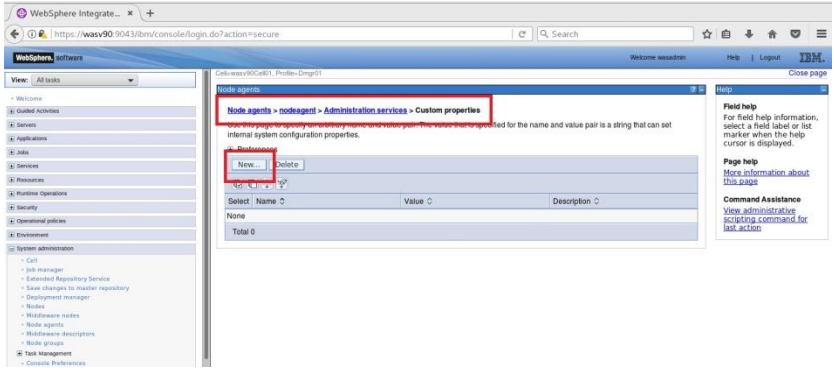


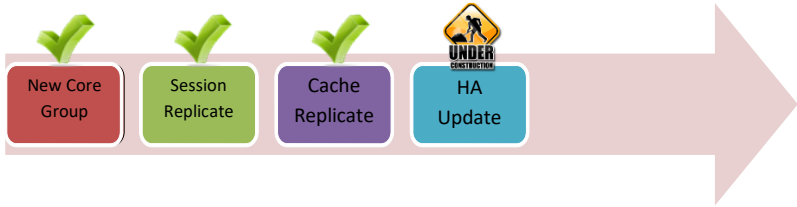


Step 3: Click on “Custom properties”.

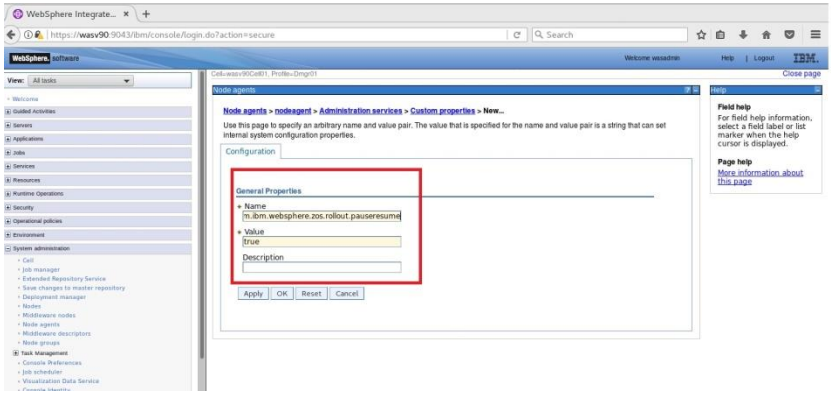


Step 4: Click “New” to add a custom property.

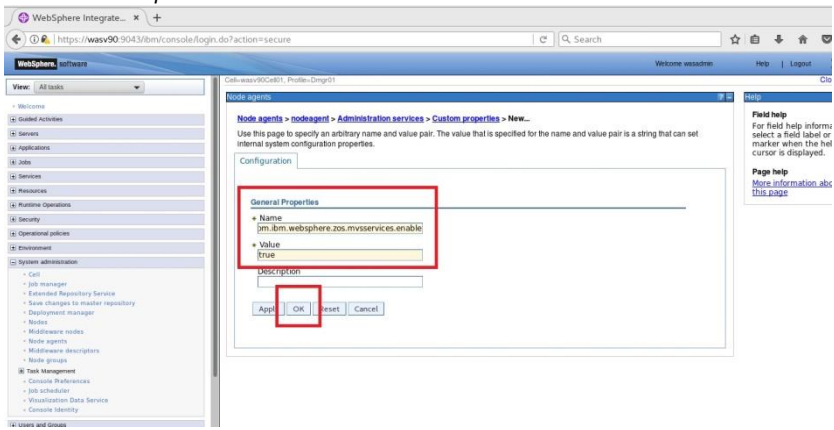


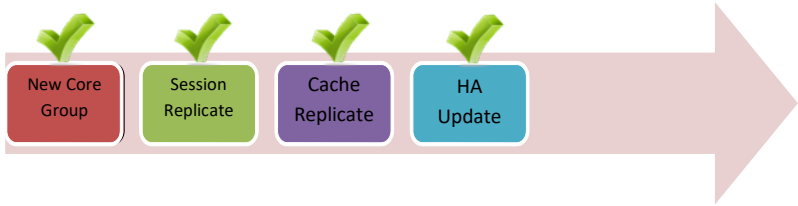


Step 5: Enter "*com.ibm.websphere.zos.rollout.pauseresume*" as "Name" and "true" as "Value" then click "OK".

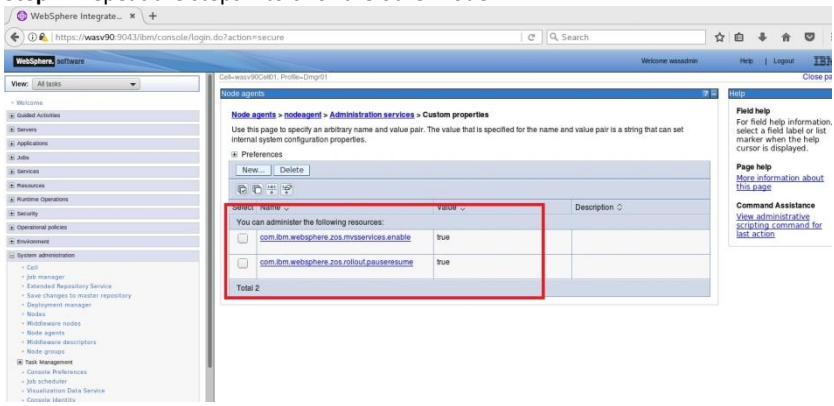


Step 6: Add another custom property that has name "*com.ibm.websphere.zos.mvsservices.enable*" and value "true" then click "OK".





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



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.productization.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://pic.dhe.ibm.com/infocenter/wxsinfo/v8r5/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.doc%2Ftxsdyncache.html

INDEX