

Lab 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

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.

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.

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.

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.

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.

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.

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.