Clustering Configuration Reference

You configure clustering in the `<PRODUCT_HOME>/repository/conf/axis2/axis2.xml` file for the load balancer and each node in the cluster. The following sections describe how to configure various aspects of the cluster.

- Clustering agent
- Group management
- Static members
- Dynamic members

Clustering agent

The clustering agent is responsible for initializing the nodes in a cluster and for all clustering-related functionality. The default clustering agent is from Hazelcast Community Edition (org.wso2.carbon.core.clustering.hazelcast.HazelcastClusteringAgent).

The following are the parameters that are used in the default clustering agent implementation:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AvoidInitiation</td>
<td>Prevents the cluster from being initialized automatically when the AxisConfiguration is built. By default, this parameter is set to true, because Carbon initializes the cluster after the server and transports have been fully started.</td>
</tr>
</tbody>
</table>
| membershipScheme| The membership scheme that will be used to manage membership of nodes in a cluster. You can set this parameter to `multicast` or `wka`:  
  - `multicast` - Membership is automatically discovered using multicasting. To use this approach, multicasting must be allowed in the network and all members in the cluster must use the same multicast address (as defined by the `mcastAddress` parameter) and multicast port (as defined by the `mcastPort` parameter).  
  - `wka` - Membership is discovered with the help of one or more nodes running at a Well-Known Address. New members joining a cluster first connect to and register with a well-known node and get the membership list from it. When new members join, one of the well-known nodes will notify the others in the group. When a member leaves the cluster or is deemed to have left the cluster, it will be detected by the Group Membership Service (GMS) using a TCP ping mechanism. WKA-based membership is necessary when multicast-based membership discovery is not possible. For example, on Amazon EC2, multicasting is not allowed, and there is no control over the IP address assigned to EC2 instances. In this case, use WKA-based membership discovery. |
| domain         | The clustering domain/group. Note that the words "domain" and "group" are used synonymously. There will not be any interference between nodes in different groups. Messages received from members outside the group will generally be ignored. However, special messages, such as cluster management messages or membership messages from members outside the group, will be allowed. |
| mcastAddress   | The multicast address to be used. This parameter will only be taken into consideration when `membershipScheme` is set to `multicast`.                                                                                 |
| mcastPort      | The multicast port to be used. This parameter will only be taken into consideration when `membershipScheme` is set to `multicast`.                                                                           |
| mcastTTL       | Controls the scope of multicasts by defining the default time-to-live for multicast packets that are sent out. Defaults to 32 with a maximum of 255.                                                            |
### mcastTimedefaults

Specifies the time in seconds that a node should wait for a valid multicast response from another node running in the network before declaring itself as master node and creating its own cluster. This applies only to the startup of nodes where no master has been assigned yet. For example, if you set this property to 60 seconds, until a master is selected each node will wait 60 seconds before continuing. Therefore, a high value can negatively impact performance, but if the value is set too low, nodes might give up too early and will create their own cluster. Defaults to 2 seconds.

### memberDropTimeout

The time interval in milliseconds to wait for a member to respond before it is deemed to have left the group and is dropped.

### mcastBindAddress

The IP address of the network interface to which the multicasting is bound. This parameter will only be taken into consideration when `membershipScheme` is set to `multicast`. Note that this can be different from the `localMemberHost` parameter, so the cluster can listen for point-to-point messages on the `localMemberHost` network interface while listening for multicast messages on the network interface bound to the `mcastBindAddress`.

### localMemberHost

The host name or IP address of this member. This is the IP address advertised by this member when it joins the group and sends messages. This should be set to a valid value other than `local host` or `127.0.0.1`. In most cases, you can set it to the IP address bound to the network interface that is used for communicating with members in the group.

### localMemberPort

The TCP port used by this member and through which other members will contact this member.

### properties

Name-value pairs specific to this member. When a member joins a group, these properties are bound to this member so that other members in the group can detect these properties. The properties `backendServerURL` and `mgtConsoleURL` are used for cluster management. The `backendServerURL` property is used to connect to the back-end management Web services of this member using the front end of the cluster manager. Similarly, if this member wants to expose its own management console, it can do so by providing the `mgtConsoleURL` property. The `backendServerURL` property is specified using the `<property name="backendServerURL" value="https://${hostName}:${httpsPort}/services/"/>` entry, where `${hostName}` represents the hostName member property and `${httpsPort}` represents the httpsPort member property, both of which are implicit member properties. (The `httpPort` property is another implicit member property.) Properties that have been previously declared can be used in subsequent property definitions. For example, you can define a new property as follows: `<property name="foo" value="${backendServerURL}/foo"/>`

### Group management

If you are using the WSO2 ELB to front a cluster of service-based products (such as WSO2 ESB, Data Services Server, and Application Server), enable the `groupManagement` element when a member is deployed as a cluster manager. A group management agent, which is an instance of the `org.wso2.carbon.core.clustering.hazelcast.HazelcastGroupManagementAgent` interface, must be specified for each group (applicationDomain) that is being managed. The following example configuration shows how cluster management has been enabled for group1 and group2:
<groupManagement enable="true">
  <applicationDomain name="group1"
    description="This is the first group"
    agent="org.wso2.carbon.core.clustering.hazelcast.HazelcastGroupManagementAgent"
    subDomain="worker"
    port="2222" />
  <applicationDomain name="group2"
    description="This is the second group"
    agent="org.wso2.carbon.core.clustering.hazelcast.HazelcastGroupManagementAgent"
    subDomain="worker"
    port="2233" />
</groupManagement>

Static members

The members element is used for specifying static or well-known members. You must specify the host name and primary port of each member. The following sample member configuration shows two static members with different IP addresses and ports:

<members>
  <member>
    <hostName>10.100.1.202</hostName>
    <port>4000</port>
  </member>
  <member>
    <hostName>10.100.1.154</hostName>
    <port>4001</port>
  </member>
</members>

Dynamic members

The members element can also be used to specify a range of well-known members. You must specify a range of IP addresses as the host name and primary port of each member. The following sample member configuration shows two dynamic members with different IP addresses and ports:

<members>
  <member>
    <hostName>10.100.1.202-209</hostName>
    <port>4000</port>
  </member>
  <member>
    <hostName>10.100.1.154-164</hostName>
    <port>4001</port>
  </member>
</members>