

Clustering Information

This guide describes the clustering model through which Clearspace run over a group of servers.

The clustering module allows Clearspace to work over a group of application servers, providing increased speed and fault-tolerance. Clusters are auto-configuring, have no single point of failure, and provide nearly linear scalability as additional servers are added.

At the heart of the clustering module is a distributed cache powered by [Oracle Coherence](#). This allows application state to be shared among cluster members. Servers can join and leave the cluster at will, and cluster node failures are transparently handled.

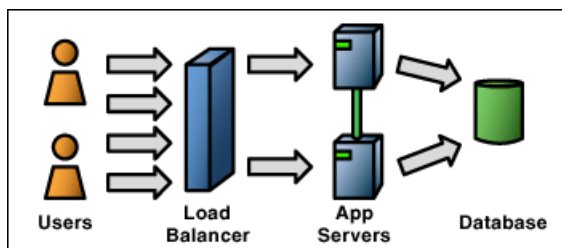
[Cluster Architecture](#) (page 1)

[Setting Up a Cluster](#) (page 2)

[Clustering FAQ](#) (page 2)

Cluster Architecture

The best clustering configuration will depend on your traffic needs, existing infrastructure, and budget. One possible clustering configuration is pictured below:



There are three layers to this setup:

- **Load-balancer** Sitting between web surfers and the application servers is a load-balancing box. Every Clearspace user has a session (represented by a unique cookie value) which allows stateful data to be maintained while they are using Clearspace. Each session is created on a particular application server. The load-balancer must be "session-aware," meaning that it inspects the cookie value and always sends a given user's requests to the same application server during a given session. Without session-aware load balancing, the load-balancer could send requests to any application server in the cluster.
- **Application Servers** In the middle-tier, multiple application servers are set up with Clearspace and the clustering feature is enabled. Caches between the Clearspace instances are automatically synchronized. If a particular application server fails, the load-balancer detects this and removes the server from the cluster.
- **Database** All Clearspace instances in a cluster must share the same database. Depending on your needs, you can use a single database server, or use a cluster of database servers for increased performance and reliability. Consult your database vendor for more information about database clustering.

Setting Up a Cluster

Your license determines whether or not clustering is enabled and how many nodes are supported. To check on the number of clustered servers your license allows, see the license information after logging into the Clearspace administration tool.

Note: Your cluster's load balancer must have sticky sessions turned on to ensure that session affinity is preserved.

Next, navigate to the cache panel. Toggle clustering to "on" and then save settings (note that this option will only be displayed if your license allows it). The server will automatically discover and join the cluster. You'll then see additional options in the cache panel to monitor cluster activity. Note: When enabling clustering on the first clustered node, there will be a 30-second delay when it starts. This is a necessary component of the clustering protocol.

You should also set the system time on each cluster member to match other cluster member's settings as closely as possible, or use a centralized time server.

Clustering FAQ

Do all cluster members need to be on the same local network?

By default, the clustering module uses IP multicast to discover the cluster. This requires all cluster members to be on the same local network or for multicast traffic to be tunneled across multiple networks. It is possible to configure Coherence to avoid the use of IP multicast. See the [Coherence documentation](#) for more information.

Is it possible to have more than one cluster per physical network?

Yes, this is possible. Coherence uses IP multicast to discover other cluster members. So, to isolate different clusters on the same physical network, you simply need to tell each cluster to use different multicast IPs or ports. There are two ways to do this:

1. Edit the tangosol-coherence-override.xml file in the clearspace-<version>.jar file and change the following values:

```
<multicast-listener>
  <address>224.2.2.245</address>
  <port>22245</port>
</multicast-listener>
```

2. Set Java system properties on each app-server (for example, in the app-server startup script) for the values `tangosol.coherence.clusteraddress` or `tangosol.coherence.clusterport`.

Should there be a separate jiveHome directory for each deployed WAR file?

Yes, each WAR should have a corresponding jiveHome directory.

How do attachments work in a cluster?

You must store attachment data in the database. They will be cached locally on each node for fast delivery.

Does searching work when clustering is enabled?

Yes, searching works without needing to make any special configuration changes. Each cluster member will maintain its own search index and update it with the latest data (when auto-indexing is enabled).

Do gateways work with clustering?

Yes, gateways will work transparently in a cluster.

How do config files work in a cluster?

All configuration data (except bootstrap information such as database connection information) is stored in the database. Changing configuration settings on one cluster member will automatically update them on all other cluster members.

How do I set Coherence operational configuration parameters such as multicast port or address?

The default operational configuration parameters are specified in several places. Inside the coherence JAR itself are the files `tangosol-coherence.xml`, and `tangosol-coherence-override-prod.xml`. Together, these two files contain the default settings for running coherence in production. Within the Clearspace JAR is the file `tangosol-coherence-override.xml`, which is where any changes to the settings in the previous two files should be made. The elements within these files are documented [here](#).

As a developer, how do I add a cache of my own?

All caches are defined in the file `WEB-INF/classes/coherence-cache-config.xml`. The format of this file is defined and [extensively documented](#) by Oracle.

Note: There are a few Jive-specific initialization parameters used within our version of this file which are ignored by Coherence. They are named 'jive-small-size', 'jive-medium-size', and 'jive-large-size'. The values of these parameters are sizes in kilobytes, and are used to display cache size presets within the administration console.