



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Genesys Knowledge Center Deployment Guide

Load-Balancing Configuration

# Load-Balancing Configuration

## Deploying a Cluster

### Important

Whenever you deploy a Knowledge Center Server instance, you must configure a Knowledge Center Cluster, even if you only plan on having one server.

Knowledge Center Cluster stores all of the settings and data that are shared by each of the Knowledge Center Server instances that reside within it. This makes it pretty easy to add additional servers as your knowledge needs grow.

Knowledge Center Cluster also serves as the entry point to all client requests sent to Knowledge Center Servers. The cluster application in Genesys Administrator needs to be configured to point to the host and port of the load balancer that will distribute these requests among your Knowledge Center Servers.

### Important

If you only have one server deployed in your cluster, you can configure the cluster application to point directly to the host and port of that server.

## Configuring Your Load-Balancer Solution

Let's take a look at how you might configure your load balancer to distribute requests between servers. This sample uses an Apache load balancer.

### Important

Genesys recommends that you use a round-robin approach to balancing.

### Important

If you need more information about load balancing in a Genesys environment, the

Genesys Web Engagement [Load Balancing](#) page provides some useful background information.

### Prerequisites

- Several Knowledge Center Servers should be installed. These servers will be used as cluster nodes (node1, node2, node3, and so on)
- You must have a Genesys Administrator application of type **Application Cluster**
- All Knowledge Center Server applications should be connected to the application cluster

### Start

1. Install the Apache HTTP Server (<http://httpd.apache.org/>). The port and host of the installed load balancer should be used in the **Application Cluster** application in Genesys Administrator.
2. Enable these modules (in the `./conf/https.conf` configuration file):
  - `LoadModule proxy_module modules/mod_proxy.so`
  - `LoadModule proxy_ajp_module modules/mod_proxy_ajp.so`
  - `LoadModule proxy_balancer_module modules/mod_proxy_balancer.so`
  - `LoadModule proxy_connect_module modules/mod_proxy_connect.so`
  - `LoadModule proxy_ftp_module modules/mod_proxy_ftp.so`
  - `LoadModule proxy_http_module modules/mod_proxy_http.so`
  - `LoadModule proxy_scgi_module modules/mod_proxy_scgi.so`
3. Configure your proxy settings ([http://httpd.apache.org/docs/2.2/mod/mod\\_proxy\\_balancer.html](http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html)):

```
# Proxy
# ProxyPass / balancer:/'knowledge_cluster'/
stickysession=JSESSIONID|jsessionid nofailover=Off
ProxyPass / balancer:/'knowledge_cluster'/
<Proxy balancer://test_cluster>
    BalancerMember http://host_node_1:port_node_1 route=node1
    BalancerMember http://host__node_2:port_node_1 route=node2
</Proxy>
ProxyRequests On
<Proxy *>
    AddDefaultCharset off
    Order deny,allow
    Allow from all
    #Allow from .example.com
</Proxy>
```

4. In each node in your Jetty server configuration, set `./etc/jetty.xml` like this:

```
<Set name="sessionIdManager">
  <New id="hashIdMgr" class=
"org.eclipse.jetty.server.session.HashSessionIdManager">
    <Set name="workerName">node1</Set>
  </New>
```

</Set>

5. Restart Apache, then restart all of your nodes
6. All requests that are sent to Apache will be distributed to your cluster nodes. The current configuration supports stickysession mode based on JSESSIONID ([http://httpd.apache.org/docs/2.4/mod/mod\\_proxy\\_balancer.html#stickyness\\_implementation](http://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html#stickyness_implementation))

### End

Here are couple of sample requests:

- Request to a specific node: [http://host\\_node\\_1:port\\_node\\_1/gks-sample-ui](http://host_node_1:port_node_1/gks-sample-ui)
- Request to the cluster, which will be distributed to any appropriate node: [http://host\\_load\\_balacer:port\\_load\\_balancer/gks-sample-ui](http://host_load_balacer:port_load_balancer/gks-sample-ui)