



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 Interaction Recording Solution Guide

Setting up the Load Balancer in a Multi-Tenant Environment

4/2/2025

Setting up the Load Balancer in a Multi-Tenant Environment

See also: [Setting up the Load Balancer in a Single-Tenant Environment](#)

Important

- The load balancer used for RWS must be configured with sufficient capacity to accommodate one persistent connection from each logged in agent with SR Service in addition to other RWS requests.
- Currently, Genesys does not provide instructions on how to set up load balancer for the Voice Processor. You can configure your own load balancing solution for multiple Voice Processor instances, if required.
- The architecture for load balancer cluster is supported on Red Hat Enterprise Linux 6 for HTTPD 2.2 only.

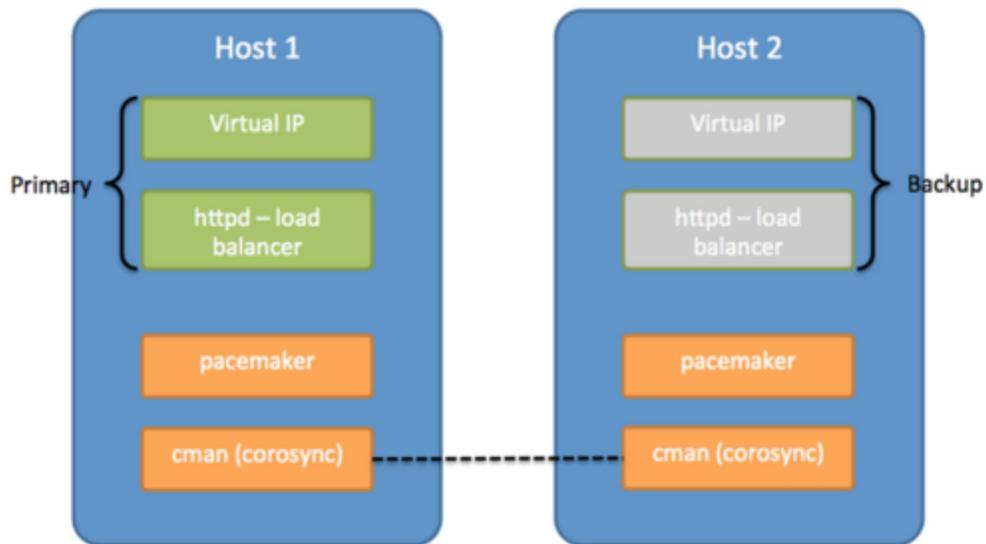
Red Hat Enterprise Linux 6 for HTTPD 2.2

Overview and Architecture

The solution uses a common Linux HA framework from <http://clusterlabs.org>. There are two components involved in this solution:

- Cman uses corosync internally to provide a platform for membership, messaging, and quorum among the hosts.
- Pacemaker is a cluster resource manager that controls where resources (processes) are executed. Pacemaker works with the processes like Apache httpd using resource agents to provide controls of the process such as start/stop/status.

The following diagram shows a primary/backup design to associate a single virtual IP address with httpd. Whenever the primary host fails, the virtual IP address and the httpd process can be automatically fail over to the backup host.



As a simple two host primary/backup solution, the hosts must be deployed on the same subnet that allows UDP multicast. This solution provides the same reliability as a network that hosts the two machines handling the virtual IP address.

Deploying the Load Balancer

Important

For load balancers used for Recording Processors, warm standby functionality must be disabled.

Prerequisites

- Red Hat Enterprise Linux 6 with the High Availability Add-On, for HTTPD 2.2

Tip

Network Manager can be enabled as part of the OS installation. To disable Network Manager, see [Red Hat documentation](#).

Installing the OS

Install the required software using the following command:

```
yum -y install httpd pacemaker cman pcs ccs resource-agents
```

Setting up the HTTP Load Balancer

Please note that any URL setup for the various GIR components described in the [Multi-Tenant Deployment](#) should now point to the respective loadbalancer URLs, such as

- RPS URL: <loadbalancer URL>/t1/rp/api
- htcc.baseurl should point to the RWS loadbalancer URL: <loadbalancer URL>/t1
- rcs.base_uri should point to <loadbalancer URL>/t1/rcs

Important

Only GIR releases post-8.5.210.02 with WDE support multi-tenancy.

On both servers, create the following files:

- Create /etc/httpd/conf.d/serverstatus.conf, and add the following text:

```
<Location /server-status>  
  SetHandler server-status  
  Order deny,allow  
  Deny from all  
  Allow from 127.0.0.1  
</Location>
```

For each tenant, create a separate /etc/httpd/conf.d/loadbalancer_tenantN.conf file. Use the Include directive within the main /etc/httpd/conf/httpd.conf to include each tenant configuration:

```
Include /etc/httpd/conf.d/loadbalancer_tenantN.conf
```

In addition, provide each tenant with a separate balancer rule, ProxyPass and the following URI conventions:

- Interaction Recording Web Services
 - http://loadbalancer/t1/api
 - http://loadbalancer/t1/internal-api
- Recording Processor
 - http://loadbalancer/t1/rp
- Recording Crypto Server
 - http://loadbalancer/t1/rcs
- Interaction Receiver

- `http://loadbalancer/t1/interactionreceiver`
- WebDAV Server
 - `http://loadbalancer/t1/recordings`
- For each tenant, create `/etc/httpd/conf.d/loadbalancer_tenantN.conf`, and add the following text:

Important

If your existing configuration already includes the loadbalancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the `_` (underscore) characters. For example, if the application name is RCS 1, set the **route** value to `RCS_1`.

loadbalancer_tenantN.conf

```
# Interaction Recording Web Services for tenant 1
<Proxy balancer://tlrws>
BalancerMember http://tlrws1:8080 route=T1RWS1
BalancerMember http://tlrws2:8080 route=T1RWS2
BalancerMember http://tlrws3:8080 route=T1RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://tlrws/api
ProxyPass /t1/internal-api balancer://tlrws/internal-api

# RP for tenant 1
<Proxy balancer://tlrp>
BalancerMember http://tlrp1:8889
BalancerMember http://tlrp2:8889
</Proxy>
ProxyPass /t1/rp/api balancer://tlrp/api

# RCS for tenant 1
<Proxy balancer://tlrcs>
BalancerMember http://tlrcs1:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://tlrcs2:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
ProxyPassReverseCookiePath "/rcs" "/t1/rcs"
</Proxy>
ProxyPass /t1/rcs balancer://tlrcs/rcs

# Interaction Receiver for tenant 1
```

```
<Proxy balancer://t1sm>
BalancerMember http://t1lr1
BalancerMember http://t1lr2
</Proxy>
ProxyPass /t1/interactionreceiver balancer://t1sm/interactionreceiver

# WebDAV for tenant 1
<Proxy balancer://t1webdav>
BalancerMember http://t1webdav1
BalancerMember http://t1webdav2 status=H
</Proxy>
ProxyPass /t1/recordings balancer://t1webdav/recordings
ProxyPass /t1/dest2 balancer://t1webdav/dest2
```

Configuring TLS for the HTTP Load Balancer

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: /etc/pki/tls/certs/localhost.crt
- Key: /etc/pki/tls/private/localhost.key

2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **/etc/httpd/conf.d/ssl.conf** file and modify the following lines:

- SSLCertificateFile /etc/pki/tls/certs/localhost.crt
- SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

3. To enable https for the proxy, edit the **/etc/httpd/conf.d/ssl.conf** file and add the following option:
SSLProxyEngine on

4. Direct the load balancer to the proper https locations. For example:

```
<Proxy balancer://t1rws>
BalancerMember https://t1rws1:8080 route=T1RWS1
BalancerMember https://t1rws2:8080 route=T1RWS2
BalancerMember https://t1rws3:8080 route=T1RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://t1rws/api
ProxyPass /t1/internal-api balancer://t1rws/internal-api
```

Setting Up Pacemaker and Cman

Important

Perform the following commands using the **root** user.

Disable Autostart for Httpd

Pacemaker manages the startup of httpd. Disable httpd from chkconfig services using the following command:

```
chkconfig httpd off
```

Setting Up the Hosts File

Make sure there is a hostname for both servers and that the hostname is resolvable on both hosts, either using DNS or /etc/hosts file. ip1 and ip2 are used as the hostnames thereafter.

```
# /etc/hosts
# ... keep the existing lines, and only append new lines below
192.168.33.18 ip1
192.168.33.19 ip2
```

Setting Up the Cluster

Run the following command on each host to create the cluster configuration:

```
ccs -f /etc/cluster/cluster.conf --createcluster webcluster
ccs -f /etc/cluster/cluster.conf --addnode ip1
ccs -f /etc/cluster/cluster.conf --addnode ip2
ccs -f /etc/cluster/cluster.conf --addfencedev pcmk agent=fence_pcmk
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect ip1
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect ip2
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk ip1 pcmk-redirect port=ip1
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk ip2 pcmk-redirect port=ip2
ccs -f /etc/cluster/cluster.conf --setcman two_node=1 expected_votes=1
echo "CMAN_QUORUM_TIMEOUT=0" >> /etc/sysconfig/cman
```

Start the Service

Start the cman and pacemaker services on each host using the following command:

```
service cman start
service pacemaker start
chkconfig --level 345 cman on
chkconfig --level 345 pacemaker on
```

(Optional) Setting Up UDP Unicast

This solution relies on UDP multicast to work, but can also work with UDP unicast. Edit the **/etc/cluster/cluster.conf** file and insert an attribute to the <cman> tag as follows:

```
...
<cman transport="udpu" two_node="1" expected_votes="1"/>
...
```

Restart both servers for the changes to take effect.

Setting Cluster Defaults

Run the following on one of the servers.

```
pcs property set stonith-enabled=false
pcs property set no-quorum-policy=ignore
pcs resource defaults migration-threshold=1
```

Configure the Virtual IP Address and Apache httpd

Run the following on one of the servers.

For the first command below, `nic=eth0` refers to the network interface that brings up the virtual IP address. Change `eth0` to the active network interface your environment uses.

Change `<Virtual IP>` in the first command below to your virtual IP assigned to this load balancer pair.

```
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=<Virtual IP> nic=eth0 cidr_netmask=32
op monitor interval=30s
pcs resource create webserver ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf
statusurl="http://localhost/server-status" op monitor interval=30s
pcs resource meta webserver migration-threshold=10
pcs constraint colocation add webserver virtual_ip INFINITY
pcs constraint order virtual_ip then webserver
```

Maintaining Pacemaker

The following commands help you with the maintenance operations for pacemaker.

To check the status of the cluster:

```
pcs status
```

To clear resource errors (for example, because of incorrect configuration):

```
pcs resource cleanup <resourcename>
```

A resource name is either `virtual_ip` or `web server` (for example, `pcs resource cleanup webserver`).

To check the status of the resources in the cluster:

```
crm_mon -o -1
```

Red Hat Enterprise Linux 8 for HTTPD 2.4

Deploying the Load Balancer

Important

For load balancers used for Recording Processors, warm standby functionality must be disabled.

Prerequisites

- Red Hat Enterprise Linux 8 with the High Availability Add-On, for HTTPD 2.4

Tip

Network Manager can be enabled as part of the OS installation. To disable Network Manager, see [Red Hat documentation](#).

Installing the OS

Install the required software using the following command:

```
yum -y install httpd
```

Setting up the HTTP Load Balancer

Please note that any URL setup for the various GIR components described in the [Multi-Tenant Deployment](#) should now point to the respective loadbalancer URLs, such as

- RPS URL: <loadbalancer URL>/t1/rp/api
- htcc.baseurl should point to the RWS loadbalancer URL: <loadbalancer URL>/t1
- rcs.base_uri should point to <loadbalancer URL>/t1/rcs

Important

Only GIR releases post-8.5.210.02 with WDE support multi-tenancy.

On both servers, create the following files:

- Create /etc/httpd/conf.d/serverstatus.conf, and add the following text:

```
<Location /server-status>  
  SetHandler server-status
```

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

For each tenant, create a separate `/etc/httpd/conf.d/loadbalancer_tenantN.conf` file. Use the `Include` directive within the main `/etc/httpd/conf/httpd.conf` to include each tenant configuration:

```
Include /etc/httpd/conf.d/loadbalancer_tenantN.conf
```

In addition, provide each tenant with a separate balancer rule, ProxyPass and the following URI conventions:

- Interaction Recording Web Services
 - `http://loadbalancer/t1/api`
 - `http://loadbalancer/t1/internal-api`
- Recording Processor
 - `http://loadbalancer/t1/rp`
- Recording Crypto Server
 - `http://loadbalancer/t1/rcs`
- Interaction Receiver
 - `http://loadbalancer/t1/interactionreceiver`
- WebDAV Server
 - `http://loadbalancer/t1/recordings`
- For each tenant, create `/etc/httpd/conf.d/loadbalancer_tenantN.conf`, and add the following text:

Important

If your existing configuration already includes the loadbalancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the `_` (underscore) characters. For example, if the application name is `RCS 1`, set the **route** value to `RCS_1`.

loadbalancer_tenantN.conf

```
# Interaction Recording Web Services for tenant 1
<Proxy balancer://tlrws>
BalancerMember http://tlrws1:8080 route=T1RWS1
BalancerMember http://tlrws2:8080 route=T1RWS2
BalancerMember http://tlrws3:8080 route=T1RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://tlrws/api
ProxyPass /t1/internal-api balancer://tlrws/internal-api

# RP for tenant 1
<Proxy balancer://tlrp>
BalancerMember http://tlrp1:8889
BalancerMember http://tlrp2:8889
</Proxy>
ProxyPass /t1/rp/api balancer://tlrp/api

# RCS for tenant 1
<Proxy balancer://tlrcs>
BalancerMember http://tlrcs1:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://tlrcs2:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
ProxyPassReverseCookiePath "/rcs" "/t1/rcs"
</Proxy>
ProxyPass /t1/rcs balancer://tlrcs/rcs

# Interaction Receiver for tenant 1
<Proxy balancer://tlism>
BalancerMember http://tlir1
BalancerMember http://tlir2
</Proxy>
ProxyPass /t1/interactionreceiver balancer://tlism/interactionreceiver

# WebDAV for tenant 1
<Proxy balancer://tlwebdav>
BalancerMember http://tlwebdav1
BalancerMember http://tlwebdav2 status=H
</Proxy>
ProxyPass /t1/recordings balancer://tlwebdav/recordings
ProxyPass /t1/dest2 balancer://tlwebdav/dest2
```

Configuring TLS for the HTTP Load Balancer

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: /etc/pki/tls/certs/localhost.crt

- Key: /etc/pki/tls/private/localhost.key
2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **/etc/httpd/conf.d/ssl.conf** file and modify the following lines:
 - SSLCertificateFile /etc/pki/tls/certs/localhost.crt
 - SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
 3. To enable https for the proxy, edit the **/etc/httpd/conf.d/ssl.conf** file and add the following option:
SSLProxyEngine on
 4. Direct the load balancer to the proper https locations. For example:

```
<Proxy balancer://t1rws>
BalancerMember https://t1rws1:8080 route=T1RWS1
BalancerMember https://t1rws2:8080 route=T1RWS2
BalancerMember https://t1rws3:8080 route=T1RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://t1rws/api
ProxyPass /t1/internal-api balancer://t1rws/internal-api
```