Bedmaster High Availability Using Mirth Channels

Notice

Vocera Communications, Inc.
www.vocera.com
tel :: +1 408 882 5100
fax :: +1 408 882 5101

Last modified: 2018-11-27 07:54

VAM-225-Docs build 26

Contents

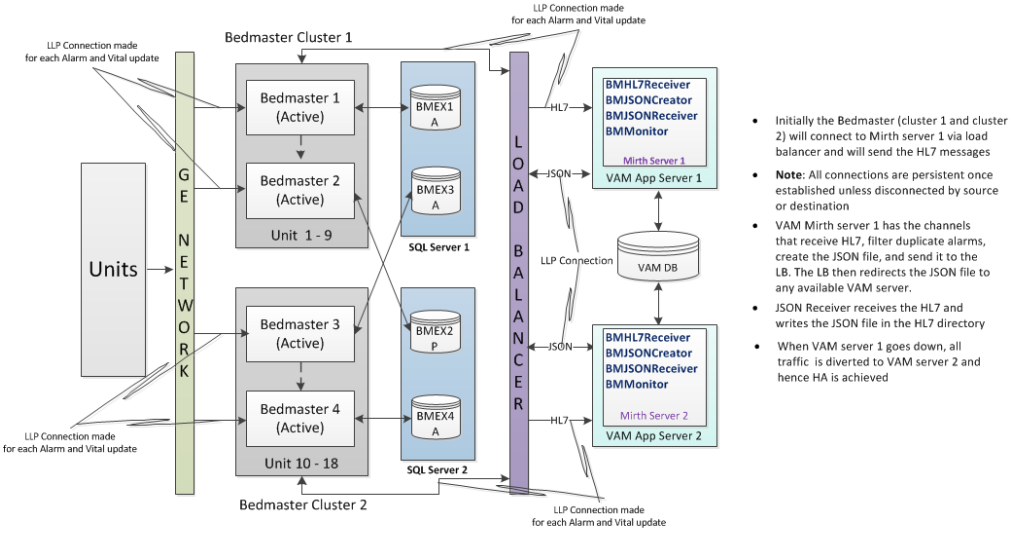
Using Bedmaster in a High Availability Environment

These sections describe how to deploy Bedmaster in a high-availability environment using Mirth channels and data deduplication logic.

In a high-availability environment, each deployed Bedmaster runs in active-active mode. This means that all alarms or vitals messages that are generated are duplicated. To handle these duplicated alarms, deduplication logic is added in the BMH7Receiver Mirth channel, which is one of the four Mirth channels needed in a high-availability environment.

Architecture Diagram

This diagram shows the typical layout in a Bedmaster high-availability environment. This shows the Mirth channels that are required.



Bedmaster HA with Load Balancer

The connection between the Bedmaster and the BMHL7Receiver channel is persistent. This means that HL7 messages are sent to VAM in the order specified in the load balancer.

The connection between the BMJSONCreator channel and the load balancer can be persistent or non-persistent. If the connection is persistent, the JSON messages are sent to only the VAM server to which it is connected, or in the order specified in the load balancer. If the connection is non-persistent, JSON messages are sent to any of the VAM servers.

Note: If the HL7 service is stopped on a Bedmaster server and then restarted, you may receive duplicate alarms.

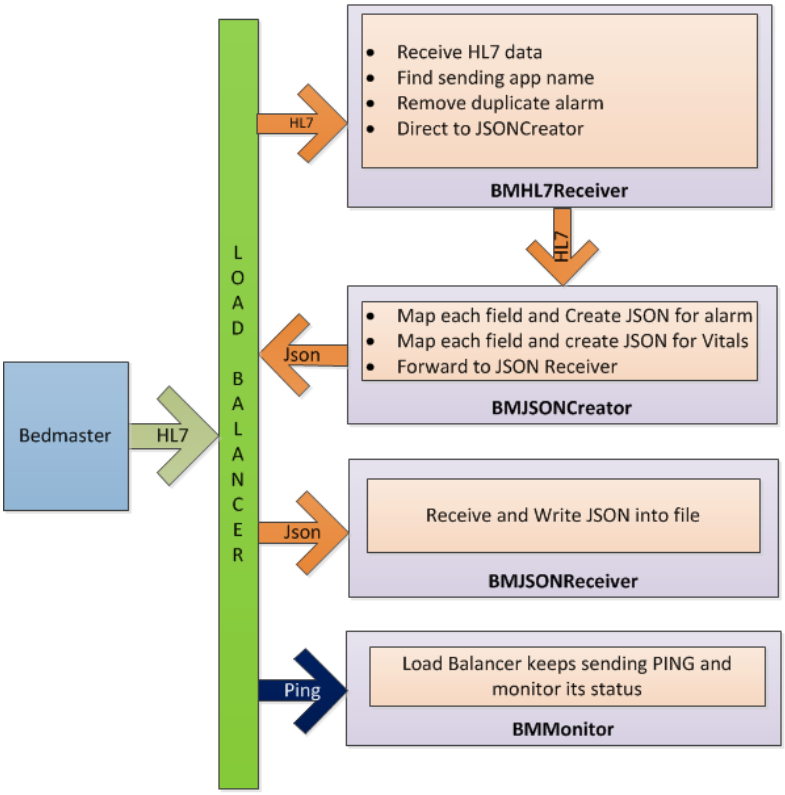
Mirth Channels

In a VAM load balancer environment, four channels are required to receive and process patient monitor HL7 data from Bedmaster.

- BMHL7Receiver
- BMJSONCreator
- BMJSONReceiver
- BMMonitor

In a load balancer environment, all four channels must be deployed on each VAM server.

This diagram shows the relationship between the Mirth channels, the load balancer, and the Bedmaster:



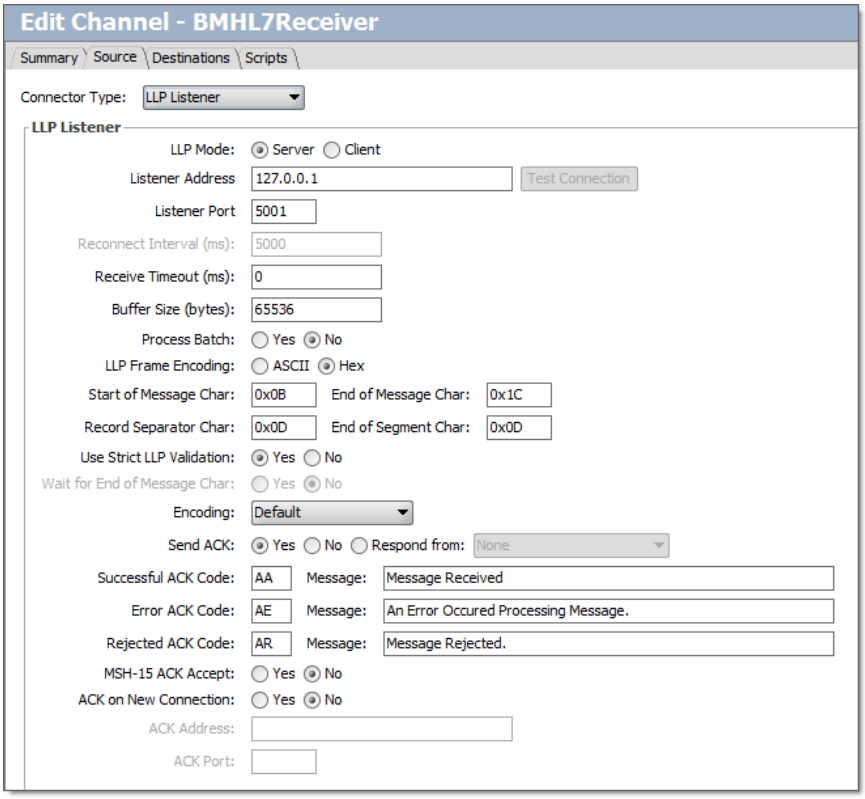
Mirth Channels required in a VAM server in LB Environment

Details on how these channels are configured can be found in the following sections. For information on how to deploy these channels after they have been created, see **Deploying a Channel** on page 9.

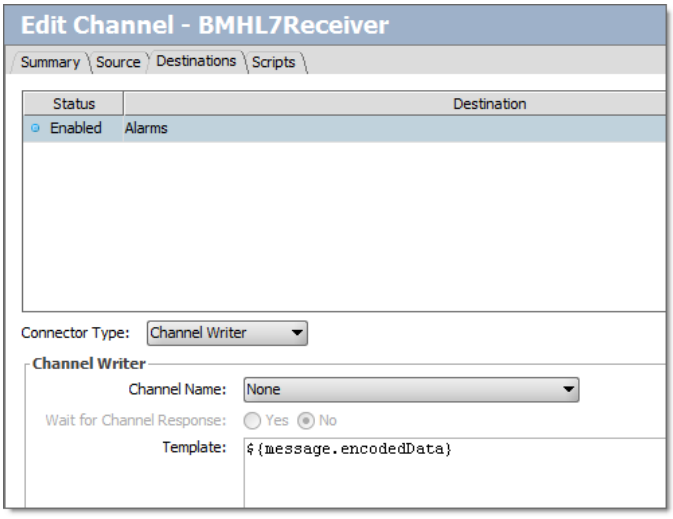
The BMHL7Receiver Channel

For the BMHL7Receiver channel, you must make these modifications.

Update the Source tab as shown:



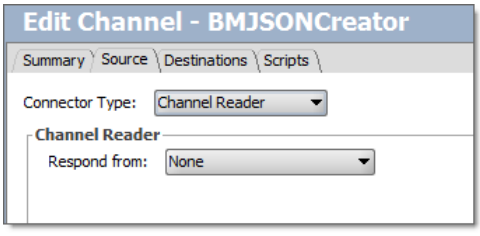
Update the Destination tab as shown:



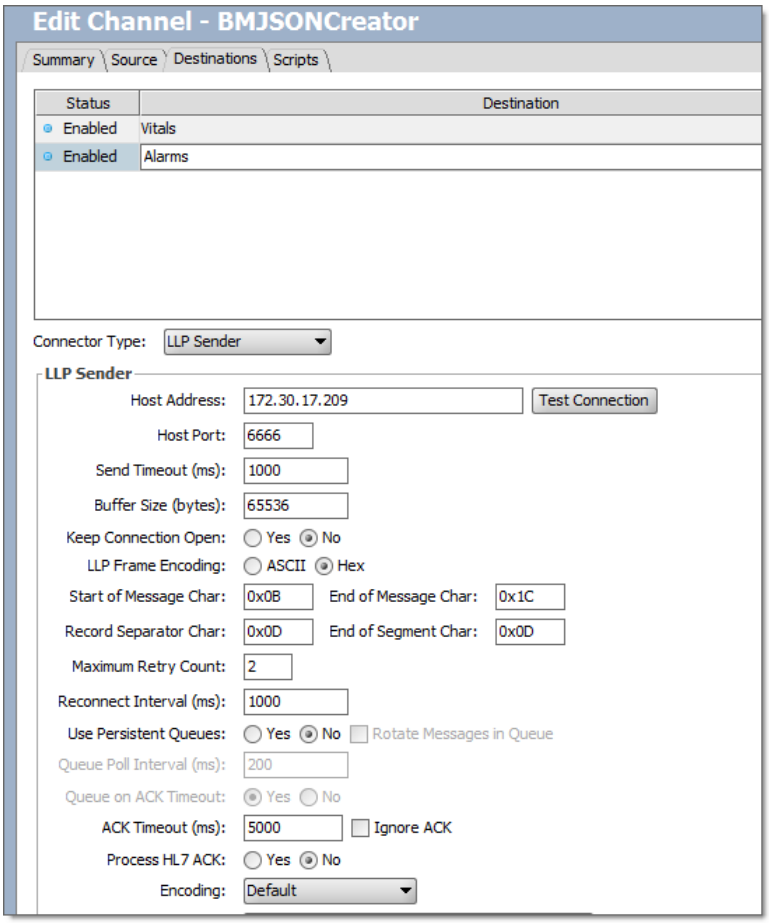
The BMJSONCreator Channel

For the BMJSONCreator channel, you must make these modifications.

Update the Source tab as shown:



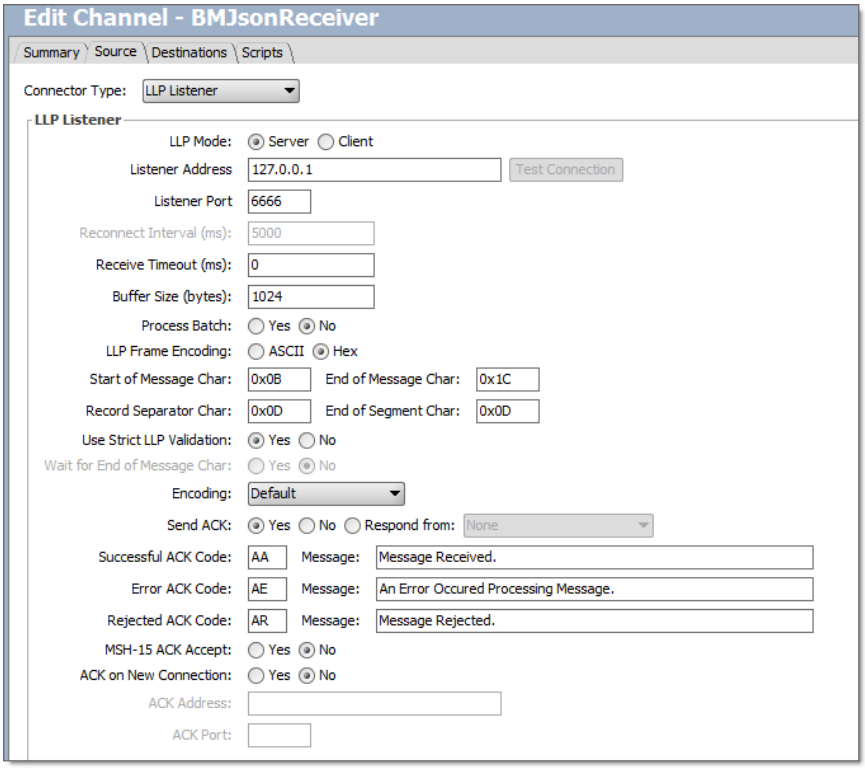
Update the Destination tab as shown:



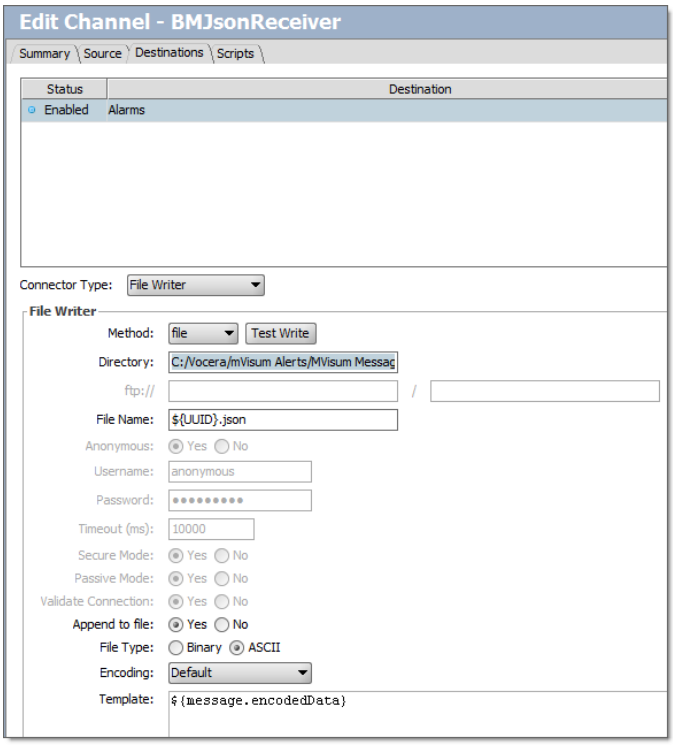
The BMJSONReceiver Channel

For the BMJSONReceiver channel, you must make these modifications.

Update the Source tab as shown:



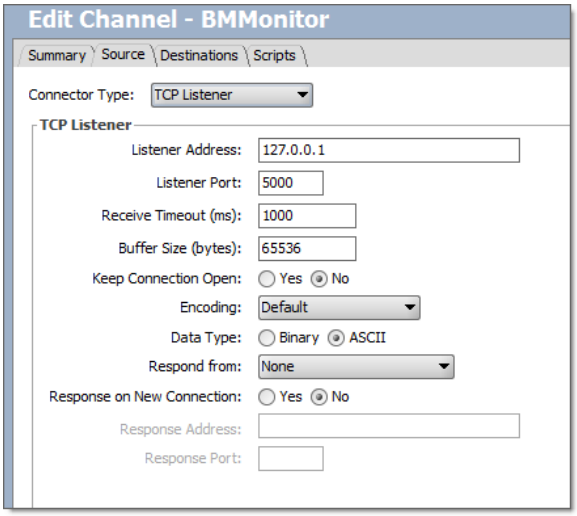
Update the Destination tab as shown:



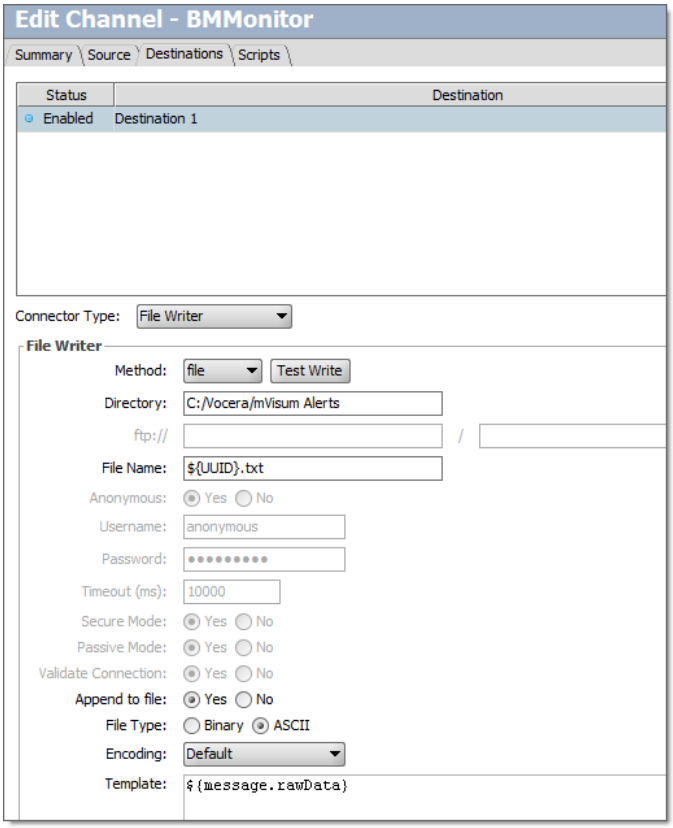
The BMMonitor Channel

For the BMMonitor channel, you must make these modifications.

Update the Source tab as shown:



Update the Destination tab as shown:



Deploying a Channel

After you have created a channel, you must deploy it.

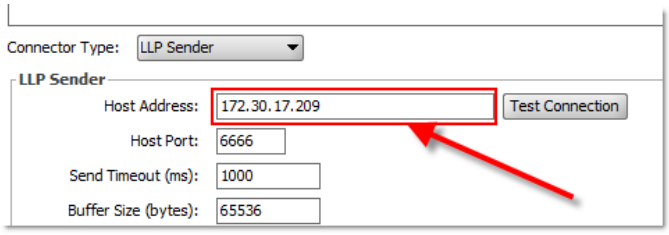
1. In the Mirth Connect Administrator, in the Mirth Connect pane at the top left of the screen, select Channels. A list of configured channels appears.
2. Right-click on the channel that you want to deploy. From the popup menu that appears, select Deploy Channel.

The Dashboard now displays the deployed channel.

Changing the Load Balancer IP Address

In the BMJSONCreator channel, the load balancer IP address is set to a default value. Follow these steps to change the load balancer IP address.

1. In the Mirth Connect Administrator, in the Morth Connect pane, select Channels. The list of channels is displayed.
2. Right-click BMJSONCreator and select Edit Channel.
3. In the Edit Channel screen, click the Destinations tab.
4. In the Host Address field, type the IP address of your load balancer.



5. Click Save Changes to save the changes.
6. Right-click on the channel and select Deploy Channel to redeploy the changed channel.

Changing the Connector Type to File Writer

If a File Writer connection is needed instead of an LLP Sender connection, you can update the BMJSONCreator channel to reflect this.

1. In the Mirth Connect Administrator, in the Morth Connect pane, select Channels. The list of channels is displayed.
2. Right-click BMJSONCreator and select Edit Channel.
3. In the Edit Channel screen, click the Destinations tab.
4. In the Connector Type dropdown list, select File Writer.

5. In the Directory field, type `<Vocera>\mVisumAlerts\MVisum Message Generator\HL7`, where `<Vocera>` is the folder in which VAM is installed. This is where the message generator obtains the JSON files.
6. In the File Name field, type `${UUID}.json`. This generates a unique ID for each file name, and ensures that the file extension is `.json`.
7. From the Append to file radio buttons, select Yes.
8. In the Template field, add the following:

```
{
  "VitalData": {
    "SendingApplication": "${SendingApplication}"
    ,"SendingFacility": "${SendingFacility}"
,"DateTimeOfMessage": "${DateTimeOfMessage}"
    ,"MessageControlID": "${MessageControlID}"
    ,"ProcessingID": "${ProcessingID}"
    ,"PatientIDExternalID": "${PatientIDExternalID}"
    ,"PatientInternalId": "${PatientInternalId}"
,"PatientGivenName": "${PatientGivenName}"
,"FamilyName": "${FamilyName}"
,"PatientClass": "${PatientClass}"
,"PointOfCare": "${PointOfCare}"
,"Bed": "${Bed}"
,"AdmissionType": "${AdmissionType}"
```

```
 ,"UniversalServiceIdentifier": "${UniversalServiceIdentifier}"
 ,"ObservationDateTime": "${ObservationDateTime}"
 ,"ObservationEndDateTime": "${ObservationEndDateTime}"
 ,"RelevantClinicalInformation": "${RelevantClinicalInformation}"
 ,"DiagnosticServiceSectionID": "${DiagnosticServiceSectionID}"
 ,"ParentResult": "${ParentResult}"
 ,"AdmitDateTime": "${AdmitDateTime}"
 ,"Vitals": ${Vitals}
  }
}

{
  "AlarmInfo": {
 "SendingApplication": "${SendingApplication}"
 ,"SendingFacility": "${SendingFacility}"
 ,"DateTimeOfMessage": "${DateTimeOfMessage}"
 ,"MessageControlID": "${MessageControlID}"
 ,"ProcessingID": "${ProcessingID}"
 ,"PatientInternalID": "${PatientInternalId}"
 ,"PatientGivenName": "${PatientGivenName}"
 ,"FamilyName": "${FamilyName}"
 ,"DateOfBirth": "${DateOfBirth}"
 ,"PatientClass": "${PatientClass}"
 ,"PointOfCare": "${PointOfCare}"
 ,"Bed": "${Bed}"
 ,"AdmissionType": "${AdmissionType}"
 ,"UniversalServiceIdentifier": "${UniversalServiceIdentifier}"
 ,"ObservationDateTime": "${ObservationDateTime}"
 ,"ObservationEndDateTime": "${ObservationEndDateTime}"
 ,"RelevantClinicalInformation": "${RelevantClinicalInformation}"
 ,"DiagnosticServiceSectionID": "${DiagnosticServiceSectionID}"
 ,"ParentResult": "${ParentResult}"
 ,"QuantityTiming": "${QuantityTiming}"
 ,"AlarmLevel": "${AlarmLevel}"
 ,"AlarmReason": "${AlarmReason}"
 ,"ObservResultStatus": "${ObservResultStatus}"
 ,"userDefinedAccessChecks": "${AttachmentDate}"
 ,"AttachmentType": "${AttachmentType}"
 ,"ControlId": "${ControlId}"
 ,"MessageType": "${MessageType}"
 ,"VersionId": "${VersionId}"
 ,"Waveform": "${Waveform}"
   ,"Vitals": ${Vitals}
  }
}
```

9. Click Save Changes to save the changes.
10. Right-click on the channel and select Deploy Channel to redeploy the changed channel.

Specifying the Sending Application Name

In the HL7 file, use the sending application name as shown in this table.

Bedmaster Server Name	Sending application name in HL7
Bedmaster 1 (active)	BedMasterEx1A
Bedmaster 2 (active)	BedMasterEx1P
Bedmaster 3 (active)	BedMasterEx2A
Bedmaster 4 (active)	BedMasterEx2P

Note: The sending application names must be configured in Bedmaster.

Default Ports

The following default ports are configured in the Mirth channels.

Channel	Port	Comment
BMHL7Receiver	5001	To receive HL7 data from Bedmaster
BMJSONCreator	6666	To send data to BMJSONReceiver via the load balancer
BMJSONReceiver	6666	To receive data from the BMJSONCreator channel
BMMonitor	5000	The load balancer sends pings to this port.

Using the Load Balancer

In the load balancer, these steps must be followed to achieve high availability.

1. Create a monitor with TCP and port 5000. One monitor must be created for each VAM server.
2. Create services for port 5001 (Bedmaster to load balancer) and the IP address of the VAM server. Create one service for each monitor.
3. Create a primary load balancer virtual service with port 5001. Add the primary VAM services to it. This sends all traffic to the primary server.
4. Set up a secondary load alancer virtual service that sends traffic to the secondary VAM server. This virtual server can be created without an IP address.
5. On the primary load balancer virtual service, under Protection Features, set the secondary load balancer service to be the backup virtual server.

   As long as the primary backend server is up, the primary load balancer virtual service sends traffic to it. If this server fails, traffic is passed to the secondary load balancer virtual service, and then to the secondary backend server.
6. Create services for port 6666 (BMJSONCreator to load balancer) and the IP address of the VAM server. Create one service for each monitor.
7. Create a virtual service with port 6666, and add both of the services that you have just created. Verify that the service order is correct (VAM 1, VAM 2).
8. Create a service with port 80 for HTTP or port 443 for HTTPS with the IP address of the VAM server. Bind the HTTP or HTTPS monitor (this monitor is present by default in NetScaler) to access the VAM Console. One service needs to be created for each node.
9. Create a virtual service with port 80 or 443 and bind the services that you have just created.

The total required is four monitors, six services, and three virtual servers.

For more details, consult the configuration document for your load balancer.