LogMatrix

NerveCenter

# NerveCenter High Availability Deployment Architecture

August 2011

NCHA6000-

## Copyright

## Disclaimers

LogMatrix, Inc. ("LogMatrix") makes no representations or warranties, either expressed or implied, by or with respect to anything in this manual, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

These applications are available through separate, individual licenses. Not every feature or application described herein is licensed to every customer. Please contact LogMatrix if you have licensing questions.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of LogMatrix. While every precaution has been taken in the preparation of this book, LogMatrix assumes no responsibility for errors or omissions. This publication and the features described herein are subject to change without notice.

The program and information contained herein are licensed only pursuant to a license agreement that contains use, reverse engineering, disclosure and other restrictions.

## Trademarks

LogMatrix is registered in the U.S. Patent and Trademark Office. NerveCenter and the LogMatrix Logo are trademarks of LogMatrix, Inc.

All other products or services mentioned in this manual may be covered by the trademarks, service marks, or product names as designated by the companies who market those products.

## Contacting LogMatrix

LogMatrix, Inc.
4 Mount Royal Ave, Suite 250
Marlborough, MA 01752

Phone 508-597-5300
Fax 774-348-4953

info@logmatrix.com

Website: www.logmatrix.com
Forum: http://community.logmatrix.com/LogMatrix
Blog: www.logmatrix.com/blog

# NerveCenter High Availability Deployment Architecture

The purpose of this document is to discuss potential scenarios for deploying NerveCenter servers in an environment where there is a need to meet certain High Availability (HA) requirements. A range of deployment scenarios is presented to suite a variety of HA levels.

This document is written assuming you are running NerveCenter Release 5.1 or later. If you are running an earlier release of NerveCenter, please contact LogMatrix Customer Support for more information on either an upgrade path or an alternative way to manage your current environment.

## Basic Architectures for NerveCenter High Availability

There are 3 basic architectures for NerveCenter servers in a High Availability role. Each option can be also customized to meet specific needs or requirements depending on specific High Availability goals:

**Cold Standby**

- Manually bring up the secondary server when the primary server fails

**Warm Standby (two options)**

- One Primary – One primary server is monitored by a secondary; if a primary failure is detected, the secondary server "turns on" node monitoring to take over for the primary.
- Multiple Primaries - There are 2 (or more) primaries, each monitoring a different set of nodes in the network. Each primary has the capacity to take over for any other primary. When one primary fails, another primary "turns on" monitoring of the nodes associated with the failed primary server.

**Dual Production**

- Two or more servers are configured identically; all servers are polling/monitoring the exact same networks/nodes. However, only one is primary for notification (i.e. sending alerts to a console, sending emails/text message, opening a trouble ticket). If the primary notification server fails, another server takes over the notification. (A variation on this would be full dual production including dual notification systems).

## Which Option is Right for You?

**Cold Standby**

The cold standby option is used most often in a disaster recovery scenario. For High Availability, it can be used when availability goals are not very strict. A manual database synchronization process would be required to ensure all nodes/models are synchronized with the primary. When a primary failure does occur, a manual "start up" of the secondary is required.

## Warm Standby

The warm standby option is the most common deployment scenario with NerveCenter customers.  With this architecture, an automated scheme for keeping the databases synchronized is used, or alternatively a change control process (manual) can be used when changes are less frequent.

With the warm standby scenario, the secondary server would have both the operation system and the NerveCenter application running.  However, the NerveCenter application on the secondary server would have network monitoring turned off – i.e. nodes unmanaged or models off.  Because the secondary server is always running, the warm standby option could be used where there are more stringent HA requirements, since less time would be lost during a switch over from a “running” server as compared to the cold standby option.

Although with the warm standby option the secondary server would not normally be monitoring network nodes, there would be several NerveCenter models running on the secondary server to monitor the primary server for a failure. This monitoring can take the form of an “I'm alive” trap sent from the primary to the secondary server on a periodic basis.  In addition, this monitoring would include the monitoring of the primary server system itself and other NerveCenter functions.

The warm standby option has the greatest flexibility to automate fail-over based on various failure scenarios (or “what ifs”) specific to your particular network environment. For example, an alternative architecture would be to have two or more primary systems.  The assumption here is the primaries are sharing the duties of monitoring the network, each system monitoring different sets of network nodes/systems.  However, all primary servers have access to all nodes/systems being monitored. Additionally, all primaries must have the capacity to take the “load” from another primary. All nodes or monitored elements need to be in sync on all primaries, but each primary only has its set of nodes “turned on” in normal operation. Each primary monitors the others for potential failure. When a failure is detected, the detecting “primary” will “turn on” the monitoring of the nodes the failed primary would normally monitor.

## Dual Production

The dual production option is used to achieve the highest level of High Availability.  In this option, two or more NerveCenter servers are configured identically and are polling/monitoring the same nodes. An automated synchronization scheme needs to be in place and each

NerveCenter will need to be a trap receiver of all managed nodes as well as have access/connectivity to all managed nodes. However, only one NerveCenter would be configured to send alert notifications (i.e. alerts to a console or auto trouble ticket generation). The NerveCenter servers that are not sending notifications would monitor the other NerveCenter servers for failure. If a failure is detected, the alternate NerveCenter server “turns on” notification to take over for the original notification server. Optionally, this architecture could be a full dual production environment where even upstream systems that receive notifications are duplicated.

The three HA options presented are summarized below showing pros and cons of each option.

| HA Option | Pros | Cons |
|---|---|---|
| Cold Standby | - Ease of deployment | - Will take more time to fail-over<br>- Not automated |
| Warm Standby | - Most flexible option for specific requirements<br>- Automated detection and fail-over<br>- Cost saving in the “shared” monitor version as no backup licenses are needed | - More setup required<br>- In the true warm standby version, will need backup licenses |
| Dual Production | - Highest HA option<br>- Automated fail-over of no actual fail-over required<br>- No “lost” polls or event detection<br>- If dual systems are physically different machines, can also meet Disaster Recovery needs | - Higher cost due to multiple production system<br>- More setup time |

## Configuration Data Synchronization

This refers to ensuring the models, nodes, and compiled MIBs remain in synchronization between primary and secondary NerveCenter systems.  If necessary, refer to the specific

NerveCenter documentation for the exact name/path to the files containing this information. In general these files would consist of the following:

- NerveCenter.ncdb
- NerveCenter.node
- nervectr.mib

There are various methods for achieving "synchronization" of this configuration data.

**Manual**

Based on your implemented change control process, as NerveCenter nodes are updated by your current process, perform the identical updates on the secondary NerveCenter server as well. This way, models are always created/modified manually via the client. In addition, when a model is deployed to production, also deploy the model (or changed model) to the secondary server as well.

**Automated**

Alternatively, the use of system admin tools or a scripted file transfer (i.e. SCP) can be used to automate the process of pushing the updated configuration files to the secondary system on a periodic basis. In addition, yet another option could be the use of available third party HA cluster products/concepts (e.g. Red Hat cluster or VMware) with a shared (or imaged) storage environment.

## Assistance from LogMatrix

Realizing that each network is unique with respect to HA requirements and limitations, please contact LogMatrix for assistance and guidance in designing and implementing your specific HA architecture for NerveCenter. LogMatrix can be reached at +1 (800) 892-3646 or info@LogMatrix.com.