Oracle® Communications Session Delivery Manager

High Availability Guide

Release 7.3

Formerly Net-Net Central

October 2013

ORACLE

Contents

About this Guide

The *Oracle Communications Session Delivery Manager High Availability Guide* describes High Availability (HA), which provides continuous management of your Session Delivery Management system.

1 High Availability

Overview

High Availability (HA) enables continuous service by masking both planned and unplanned downtime and preventing single points of failure. Services remain up over 99% of the time in a given period. An HA cluster is a network of tightly linked servers that are also known as members or nodes. Fail-over clusters pool together multiple members, each of which is a candidate server for your file systems, databases or applications. Each of these systems monitors the health of other systems in the cluster.

In the event of failure in one of the cluster members, the others take over the services of the failed node. When an interruption or failure occurs in a critical application, high availability clustering will combat this by switching the operations of this application to one of the other computers or nodes within that cluster.

Net-Net Central HA Cluster

Net-Net Central HA ensures that data and business processes are partitioned evenly across a cluster of NNC servers (members or nodes) without compromising availability. It ensures reliable access to NNC services at all times. Net-Net Central HA cluster of redundant nodes provides continuous service even if system components fail. In the cluster, the members work together to ensure that data remains continuously available. When a fault or potential interruption does occur, the HA cluster works to quickly and seamlessly prevent a complete system failure. Because there is not one single point of failure within the cluster, operations can continue without any noticeable interruptions to the user.

While Net-Net Central is based on a modular architecture that allows for customization, it is recommended that customers requiring a clustering solution run default configurations.

Terminology

Net-Net Central HA uses the following terminology:

- HA**:** Ability of a system to provide reliable access to services to its users. Highly available services remain up over 99% of the time in a given period.
- Co-located HA: System provides HA and high scalability using components that are geographically co-located. Further details are provided below.
- Planned downtime: Planned maintenance window that could be disruptive to system operation. Planned downtime events can include patches or re-configure components such as adding new members to a cluster or allowing members to rejoin a cluster.
- Unplanned downtime: Events that were not planned such as power outages, network, or hardware failures.
- Election: When the database master fails or becomes partitioned from the rest of the nodes in the cluster, an election is automatically held to elect a new database master. In a cluster composed of three or more nodes an election is won by the node with the simple majority of votes. Also, the node that wins an election is always the one with the most recent log records. In the case of a tie, the node with the highest priority

wins. In a two node cluster, an election can be won with a single vote. The replica can be elected master if the master fails.

HA Process

Net-Net Central (NNC) clustering offers reliable access to Net-Net Central services at all times. The cluster resilience in failure conditions provides the High Availability. The Net-Net Central cluster accomplishes this by categorizing data into different distribution mechanisms. The following summarizes these distribution mechanisms:

- Device configuration: The device is considered as the master database for configuration data. Any member of a cluster can go to the device and retrieve the most recent configuration. This process is called “on-demand” loading of configuration data. Not only is the configuration on each member synchronized to the same version, this has the advantage of removing the need to replicate large datasets between members.
- Message driven data: Some data sets that can be subject to network latency such as fault management events, polling statistics and audit trails are distributed via a Message-Oriented Middleware (MOM). The MOM provides asynchronous processing that allows the NNC systems to scale both vertically and horizontally. The resilience of the MOM is maintained via guaranteed deliveries and durable subscribers. The data is generally stored on the local host machine in a dedicated local database.
- Database replication: Sensitive data sets such as LCV, user security, and device management are transactional and need to be available across the cluster are maintained by a database replication group. The database replication group maintains one master database in the cluster at all times while all other members are replicas. Retrieval of datasets is done on the local host machines. However, transactional modifications to the data are done on the master database, which then replicates the transaction to the replicas. Replication keeps the cluster database synchronized. If the master database fails, the remaining replicas elect a new master database.
- Push-pull file transfer: Large datasets such as route sets, that are maintained in the local database or file system are transferred around the cluster via push pull mechanisms. Host members use the MOM to publish events indicating that datasets are available. The other members use SFTP to pull or push the information from or to other members in the cluster.

The Net-Net Central HA process also guarantees execution of submitted tasks and ensures that data distribution is centralized. In order to ensure no single point of failure, a NNC server runs a load balancer, a front-end server, and a back-end server.

- Load balancer: Entrance point to Net-Net Central services. The load balancer provides SSL security (HTTPS), as well balancing loads among all front end servers contained in the cluster. All cluster members will run a load balancer to ensure that there is no single point of failure. Access to NNC services is not denied as long as one NNC server is running.
- Front end server: Responsible for maintaining the client interaction support. The front end server manages sessions and performs authentication and authorization. By default the front end server targets a local back end server.
- Back end server: Runs the services required to support any functionality provided by Net-Net Central. For example, the back end server can provide route management, fault management, or configuration management functionality.

  The back end server also hosts the embedded database service and the message service. These services are responsible for maintaining the distributed data flow and provides for redundant failover capabilities.

- Embedded database service: Framework (Berkley DB XML) that provides the database XML support for the cluster. It is an in-memory database that supports replication and seamless negotiation of allocating a master database from among all the XML database members in the cluster. The database service (DBS) provides the database functionality in the back end server. Two database instances exist on any host, one is a local database XML and the other as part of a replication group.
- Message service: Message Oriented Middleware (MOM) used in the back end server to support distributed message queues and topics. MOM is supported by the distributed event and data service (DEDS) that allows components to publish and subscribe to message topics and queues.

Co-location

All host members of a cluster must reside at the same geographical location, as well as the same IP network.

Co-located clusters must also follow these requirements:

- No firewalls can exist between the members of the cluster.
- Firewalls can exist between client browsers and the cluster members so long as the ports specified in the installation guide are exposed.
- Firewalls can exist between Net-Net SBCs and the cluster members so long as the ports specified in the installation guide are exposed.

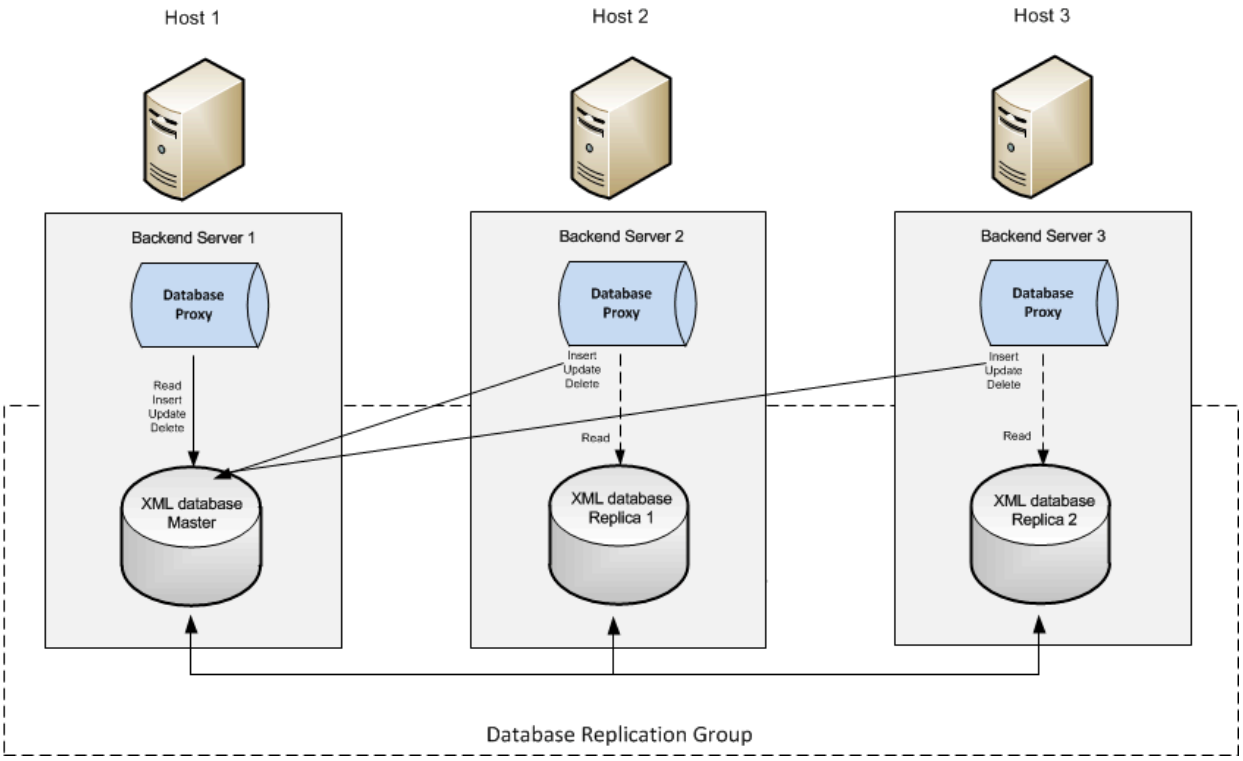
Single Master with Multiple Replica Strategy

NNC clustering offers two distinct types of clustering configurations these are:

- Multi-member cluster
- Two-member cluster

Multi-Member Clustering

In this strategy the NNC cluster contains three or more servers. Acme Packet recommends that for best HA performance and maintenance a three member cluster is configured as shown in the diagram below.



As illustrated in the diagram above, the cluster consists of three host systems. The database replication mechanism shown demonstrates the following points:

- Each host is running a backend server with an embedded-in-memory Berkeley XML database.
- In the replicated group, there is only one master database and multiple replicas.
- The master database is responsible for distribution of transactional modifications to the other replicas in the cluster.
- All back end server components interact with the database via a database proxy.
- The database proxy determines if the request for service is a transactional modification or a request for data retrieval. All data retrievals are done on the local database irrespective if it is a replica or master database. Requests for transactional modifications (inserts, updates or deletes) are forwarded from the database proxy to the master database in the cluster.

- The master database will guarantee the transactions on a quorum basis in a cluster. This means that in a cluster containing three or more members, the majority of the members need to reply that they have received the replicated datasets before the master returns success on the transaction. For example in a three member cluster the master needs to have a reply from at least one member before declaring success on the replication.
- User transactional latency is accounted for by detection of the late arrival of replicated data. Best effort replication is provided, which can mean the call might return before the dataset appears on the replicated databases. The database transactional layer offers additional support with latency in replicated data.

  For example, a user on Host 3 starts a local transaction with the database proxy to insert content into the database. The database proxy in turn starts a transaction with the master database on Host 1. Each transaction that is started with the master database has a transactional id (txId) associated with it. The master database uses best effort in ensuring that the datasets are replicated to the other members of its replication group.

  However, if the best effort takes longer than N seconds and the master database has received replies from quorum (the other replicas); the master database returns success. Returning success guarantees replication will occur at some point. The database Proxy on host 3 waits until the required txId appears in its local replicated database before returning success on the transaction to the user on host 3. This guarantees that the content inserted on the Master database has reached the replicated database. Users that initiate transactions are guaranteed to see the outcome of those transactions in their local database independent of which host the original transaction was initiated.

HA for Multi-Member Cluster

In a multi member cluster, high availability is provided for as follows:

- Database replication group:
  - Maintains one master in the entire cluster with multiple replicas.
  - On master database failure, re-election among the replicated database occurs and a new master database is elected.
  - For three or more member cluster, transactions are deemed successful only if quorum of replies from replicas is achieved. This guarantees that the data exists on more than the master database after the transaction completes. If quorum is not met the transaction will fail.
- Messaging event and data distribution:
  - The MOM distributes the messages in the cluster based on a store and forward process. The MOM guarantees message delivery by storing the message in a local database first before declaring that the message was properly processed.
  - There is no master in the MOM cluster, all MOM brokers participating in a cluster ensure that messages are synchronized in the cluster.
  - Durable subscribers ensure that even if a member leaves the cluster and reenters within a 24 hour period, missed messages are re-delivered.
  - Tasks entered in a queue are processed even if the host where the task was originally submitted goes down.
- On-demand device configuration loading:
  - The device is the master database for configuration data.
  - No replication of large configurations is required.

– Failure of a member in the host does not effect on-demand retrieval of configuration from devices serviced by other members.

Database Group Replication Data Content

Data is not compromised and is guaranteed to be replicated throughout the cluster, which ensures that failover is seamless. The data includes:

- User management data, user credentials for authentication, and user access control lists (ACL) for authorization
- User configuration modifications made per device, known as the Local Configuration View (LCV). When users make modifications to the configuration on a targeted device, only the modification changes are maintained in the LCV. This smaller dataset is replicated, reducing the latency introduced when replicating larger datasets.
- Distributed locking data. This data indicates when an NNC lock has been placed on a targeted device for a specific operation. A distributed lock ensures that no two operations can occur concurrently on the same device.
- Net-Net SBC details created when a device is added to Net-Net Central for management. This dataset provides the information required for Net-Net Central to communicate with the targeted SBC as well as provide relevant hardware, firmware, configuration and reachability status.
- NNC configuration details. This dataset provides the user customized required configuration details that inform NNC services on how to function.
- Fault sequence number, which is a global unique ID required to give distributed items uniqueness.

Database Group Replication HA Behavior in Multi-Member Cluster

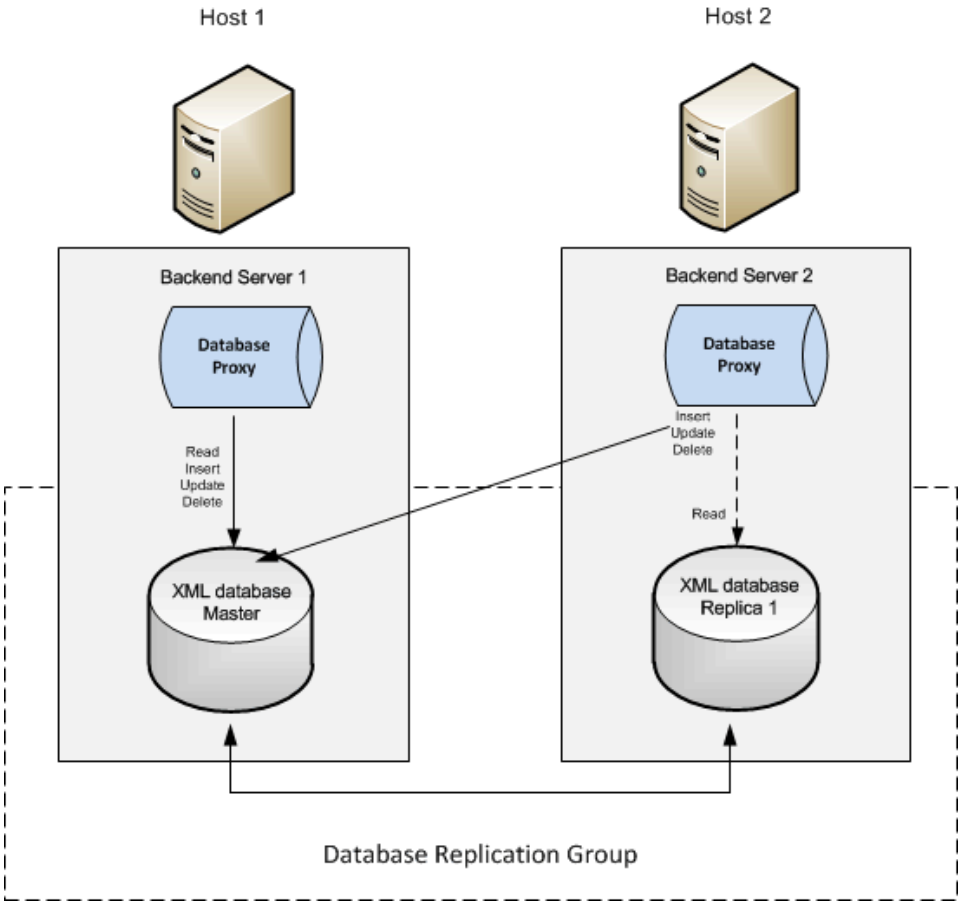
The following table lists the activities involved in database HA behavior in a multi-member cluster.

Activity	Description
Server startup	When an NNC server is started it joins the cluster as a replica and an election is held if there is currently no master. If the member ends up as a replica, then it is synchronized with the master during the initialization phase of the database service startup.
Master member failure	When the master fails or becomes partitioned from the rest of the members in the cluster, an election is automatically held by the remaining replicas to elect a new master.
Transactions (Quorum)	Transactions return successfully if a majority of the members in the cluster have replied that they received the replicated datasets. If quorum in replies from replicas is not achieved in a specific time period, the transaction fails.
Network partition	When a network partition exists between the members, only the members that can communicate with a majority of the members may elect a new master database. Members that can not communicate with a majority will enter READ-ONLY mode. Upon re-establishing network connectivity, re-elections take place and a master is elected while the other members revert to replicas.

Activity	Description
Elections	In a cluster of three or more members, an election is won by the member with the simple majority of votes. The member that wins an election is always the one with the most recent log records. In the case of a tie, the member with the highest priority wins.
Rejoining a cluster after a graceful shutdown	When a server is restarted within several hours after it shuts down, the behavior is similar to the server startup activity.
Rejoining cluster after shutdown for an extended period	Perform a hot backup on the host running the master database before restarting a server that has been down for a long time. This avoids the potentially high cost synchronizing the server with the master during startup.

Two-Member Clustering

In this strategy the NNC cluster contains only two members in the cluster as shown in the diagram below.



As illustrated in the diagram above, the cluster in this case consists of only two host machines. Considering for the moment only the database replication mechanism the following points need to be appreciated:

- Most of the items described for a three member cluster pertain here as well.

- The most important difference is that with two members the quorum policy cannot be adopted since if a network partition exists between the host machines transactions would consistently fail.
- In order to ensure transactions are serviced if a network partition exists both member will act as master databases. When the networks is re-established the members will elect which one will shut down, leaving the other to maintain services.

HA for Two Member Cluster

In a two member cluster, high availability is provided for as follows:

- Database replication group:
  - Maintains one master in the cluster with one replica.
  - On master database failure, the remaining replica becomes master.
  - If a network partition exists both members become masters. On re-establishing network connectivity one of the member will shutdown.
- Messaging event and data distribution:
  - Remains the same as described for multi member cluster
- On-demand device configuration loading:
  - Remains the same as described for multi member cluster

Database Group Replication HA Behavior in a Two Member Cluster

The following table lists the activities involved in database replication HA behavior in a two member cluster setup.

Server startup	When an NNC server is started it joins the cluster as a replica and election is held if there is currently no master. If the member ends up as a replica, then it is synchronized with the master during the initialization phase of the database service startup.
Master member failure	When the master fails the remaining replica becomes the new master.
Transactions (Quorum)	Transactions return successfully if a majority of the members in the cluster have replied that they received the replicated datasets. If quorum in replies from replicas is not achieved in a specific time period, the transaction fails.
Network partition	When the master fails the remaining replica becomes the new master.
Elections	An election can be won with a single vote. This allows the replica to be elected master in the case the master fails.

Recovery after a network partition	In a two node cluster it's possible for the network connection between the master and replica to be partitioned or become unresponsive due to network latency. In this situation an election is held and both nodes are elected and act as masters. While in this state, write transactions can occur at both sites. As a result, special handling is required after the partition is resolved and the system recovers from a two master configuration to a single master configuration: • Before the partition is resolved both nodes will be in the role of master. • After the partition is resolved an election is automatically held to elect a master. • When the election is complete the node that wins remains the master and the other will become the replica. • The node that loses the election and becomes the replica after previously operating as master is immediately shut down. The node that loses the election after a partition is resolved is shut down because the two nodes might not be in synch. All Write transactions that occurred while the nodes were partitioned cannot be automatically resolved. Only Write transactions that occurred on the node that is elected master after the partition are accepted. Because Write transactions might have also occurred on the other node, it is shutdown and must be manually synchronized with the master before it is restarted.
Rejoining a cluster after graceful shutdown	An election can be won with a single vote. This allows the replica to be elected master in the case the master fails.
Rejoining cluster after shutdown for extended period	Perform a hot backup on the host running the master database before restarting a server that has been down for a long time. This avoids the potentially high cost synchronizing the server with the master during startup.

Data Synchronization Scenarios

The following sections describes different data synchronization scenarios.

Adding Member to a Cluster

Database backups from a standalone system cannot be restored to a cluster node. You must first configure the node as a member of the cluster before creating and restoring backed up databases. See *The Net-Net Central Installation Guide* for details.

> **Note:** Acme Packet reccomends adding a new member to a cluster during planned downtime.

You can add a member to an existing cluster using the following steps:

To add a member to a cluster:

1. Install Net-Net Central on the server you plan to add to the cluster.
2. Run the setup.sh script and add all previous members in a cluster to this new member's neighbor list.
3. Log into the running cluster bring up the HM console and record which server is running the master database.
4. Stop the replica members first, followed by the master.
5. Backup the databases on the host where the master database was running using the backupdbcold.sh script.

   Note: If the member being added to the cluster contained a database at any point in time, run the bin/reinitialize.sh script.

6. Restore the database on the new member that is joining the cluster.
7. Run the setup.sh script on all other members of the cluster and add the new member to their neighbor list.
8. Start the master server first, and then the replica servers.

Member of a Cluster Fails or is Shutdown

The following describes automatic sequences that take place upon member failure or shutdown.

- Berkley DB XML: if the member leaving is running the master database, the remaining replicas carry out an election and elect a new master.
- MOM: maintain topic messages for any durable subscribers registered on the host that left. The messages would be maintained for a 24 hour period before truncation clean up occurred.
- Services that share common task processing or locks ensure that any locks acquired by the services on the member that left the cluster are cleaned up. Submitted tasks will still be processed by the remaining cluster members. Examples of these tasks include:
  - Save and Activate
  - Poller
- Clean up of any locks maintained for device synchronization. The service acquiring a lock generally provides an age-out time. If the original service cannot clean up the lock, it automatically gets aged out.

- Tasks initiated by the member that left the cluster would be re-submitted,
- Load balancers on the active hosts in the cluster bypass the front end server that has left the cluster. Clients are redirected to a valid running host.
- Health monitoring service: Health heartbeat monitor determining that a member's heartbeat has failed publishes a failed membership message to the cluster to inform all registered subscribers that a member has left the cluster.

Member Rejoining a Cluster

When a member that was previously part of a cluster rejoins that cluster, data resynchronization is required for the following databases:

- DB XML local master
- DB XML belonging to a replication group.

Types of members rejoining a cluster include those effected by the following:

- Temporary communication outage: All cluster members are running but network issues temporarily cause members to drop connections to each other.

  The tolerance window for communication outage is 8 seconds.

  Because of dynamic reconnection features, a member can temporarily leave the cluster for any reason. As long as the member NNC server has not been out of contact from the cluster for too long rejoining and re-synchronization is automatic. The Berkley DB XML will carry out re-elections and replicate datasets to replicas. The MOM brokers send messages missed by other brokers.
- NNC member fails or is stopped. A server fails outright and is down for some time until the administrator executes the startnnc.sh script. Determining how long the member has been out of contact is important:
  - If the member does not rejoin during the 24 hours of message storage, missed messages might not be sent to the joining broker. This server should be considered out of the cluster for extended period of time and needs to be treated as a member that needs to be re-synchronized before re-entering the cluster. If this is the case the following procedure is needed to re-join a member that has been out of the cluster for an extended period of time.

    1. Log into the running cluster, bring up the HM console and record which server is running the master database.
    2. Backup the databases on the host where the master database was running.

    Note: If the member being added to the cluster contained a database at any point in time, run the bin/reinitialize.sh script.

    3. Restore the database on the member that is re-joining the cluster.
    4. Start all NNC servers in the cluster that includes the new member
  - The existing master database automatically detects a replica trying to rejoin and performs an internal initialization. Because internal initialization requires transfer of log file records that were missed to the client, it can take a lengthier period of time and could require database handles to be reopened in the replica's applications.

High Availability Scenarios

The following instructions are for different high availability scenarios. They contain a summary of steps, refer to the *Net-Net Central Installation Guide* for details.

Setup a Cluster

To setup a cluster for the first time:

1. Run setup.sh on each server in the cluster.
2. Choose the Custom setup option.
3. Apply the new license.
4. Configure HTTP/HTTPS.
5. Configure sudo password.

Entering Member Information

1. Choose Net-Net Central cluster management option.
2. Choose Configure and manage members in cluster.
3. Enter Yes to continue.
4. Choose Add a new member.
5. Enter Yes to continue.
6. Enter the member IP address.
7. Repeat steps 4 through 6 to add additional members to the cluster.
8. Choose Apply new cluster configuration when done entering new members to the cluster.
9. Enter Yes to continue.
10. Choose Quit out of cluster configuration.
11. Enter Yes to continue.

Note: The status of the device can take two polling cycles to update if you are adding a large number of devices to a cluster.

Configuring SFTP Properties

1. Choose Yes when prompted whether the machine will be a member of a cluster.
2. Enter Yes to continue.
3. Enter the username used to SFTP files off the machine.
4. Enter the password for the username.
5. Choose Quit setup to finish.

Start the Servers

Start the cluster members at the same time, or one at a time. The first server that is started becomes the master server for the cluster.

Shutdown a Cluster

To shutdown a cluster:

In order to avoid unnecessary database elections, first shutdown the servers running the replica databases, and then shutdown the server running the master.

1. Login to the NNC GUI client and access the Health Monitor to identify the server with the master database.
2. Identify the master database in one of two ways:
   - In the Heartbeat console, the server that is running the master database has (master) next to the IP address.
   - Search the logs/DbService.log file for a line that looks like this:

     2011-06-14 16:52:16,266 INFO [com.acmepacket.ems.server.services.database.ReplicatedXMLDatabaseManagerImpl] - Method: [setCurrentState] Thread: [Thread-16:74]

     Msg:[Current State = unknown, New State = master]

     2011-06-14 16:52:16,295 INFO [com.acmepacket.ems.server.services.database.ReplicatedXMLDatabaseManagerImpl] - Method: [initDbEnv] Thread: [DatabaseService:20]

     Msg:[Replicated Database environment initialization complete. Role = master, IPAddress = 172.30.80.19]
3. Login to the other servers that are running replica databases first and run the shutdownncc.sh script on each.
4. Once these servers have shutdown, login to the server running the master database and run the shutdownnc.sh script.

Retiring Cluster Member

To retire a member from a cluster during planned downtime:

1. Shutdown the node to be removed from the cluster.
2. Uninstall Net-Net Central from that node.
3. Shutdown all remaining nodes starting with the replicas, followed by the master.
4. Run setup.sh
5. Choose the Custom setup option.
6. Choose Net-Net Central cluster management.
7. Enter Yes to continue.
8. Choose Configure and manage members in a cluster.
9. Enter Yes to continue.
10. Choose Remove all remote members to remove the first node from the cluster configuration.
11. Enter Yes to continue.
12. Choose Proceed with removing all remote members.
13. One by one, add all members to the cluster, excluding the retired member.
14. Enter Yes to continue.
15. Remove all of the files from the db/replicated directory except for DB_CONFIG.
16. Startup the master node, followed by the replicas.

After startup is complete, ensure the replications between the two nodes is in synch.

Repeat steps for each remaining node in the cluster.

Network Partition in Two Node Cluster

You are running a two node cluster when connectivity between the two nodes is lost. Each node elects itself as master. At this point, it is possible to lose any writes that occur to one of the databases. When the network connectivity is re-established automatically, the node that loses the election to be the master shuts down. The writes that occurred to that node are lost.

1. To re-establish the cluster, perform a hot backup on the node that is still operational.
2. Run the bin/reinitialize.sh script on the shutdown node.
3. Restore the database backup from the master onto the shutdown node.
4. Startup the shutdown node.

Network Partition in a Three Node Cluster

You are running a three or more node cluster. The network connectivity between two or more of the nodes is lost. The partition that contains a majority of the nodes elects a master from among those nodes. Any partition containing a number of servers less than the majority transitions to a READ-ONLY mode. Database updates are not permitted until the partition is resolved or the cluster has been reconfigured. A four node cluster that is partitioned into two 2 node clusters will contain two clusters running in READ-ONLY mode. When network connectivity is re-established, one node is elected master and the cluster resumes normal operation.