Contents

Enabling Email Notifications for Secure API Manager 1.1

Enabling Email Notifications to Be Sent When Any Node in the Database Service Cluster Is Down

December 2019

Secure API Manager contains a Database Service component that hosts multiple databases that store configuration information, user accounts, and information about the APIs. This information must be synchronized between the nodes when you cluster the Database Service component.

Secure API Manager uses SymmetricDS to synchronize the information between the Database Service nodes. When you cluster the Database Service, one of the nodes contains the configuration information. The node that contains the configuration information is referred to as the **master node**.

If the master node goes down, it is important that you receive a notification so you can make another node the master node and there is no information lost. The other components continue to work if the Database Service component goes down. Do not run in this mode for long or you will eventually lose data.

This document provides information about how to configure and implement the SQL file to send email notifications when any of the Database Service nodes are down and not responsive.

For information on how to promote another node to be master, see “Preparing Secure API Manager for a Disaster” in the *NetIQ Secure API Manager 1.1 Installation Guide*.

To enable email notifications to be sent when any of the Database Service nodes go down, you must download an SQL file, edit the file, and then import the SQL file to the Database Service component. You must understand SQL databases to perform the following tasks or have your database administrator perform the tasks.

Legal Notice

Downloading the SQL File

To enable email notifications to be sent when a node in the Database Service cluster is down, you must download the product ZIP file through the Customer Center. The product ZIP file contains the zipped `email-notification.sql` file. The Customer Center is where you download the products you have purchased.

You must have purchased Secure API Manager to access the ZIP file in the Customer Center. For more information, see How to Buy.

To download the SQL file:

1. Log in to the Customer Center.
2. Click **Software**.
3. On the **Entitled Software** tab, click the appropriate version of Secure API Manager for your environment.
4. Click the zipped product file to download the file.
5. Extract the product file and locate the `email-template.zip` file. This file contains both the SQL script file and this Technical Reference.
6. Copy the `email-template.zip` file to a workstation that can access the Database Service nodes.
7. Extract the `email-template.zip` file.
8. Use the information in the following sections to configure the `email-notification.sql` file for your environment.

Understanding the SQL File

To enable notifications to be sent when any of the Database Service nodes go down, you must edit the `email-notification.sql` file. The file contains configuration information about the four database tables that SymmetricDS uses to send notifications. SymmetricDS uses the four database tables to identify what to monitor, whom to notify, and how to send the notification.

NOTE: There are entries in each table: `NODE_GROUP_ID` and `EXTERNAL_ID`. Specify the value of `ALL` for each of these entries to include all of the nodes in the Database Service cluster instead of specifying each node separately.

Use the following information to understand what the four tables do and what columns you must change to be able to send email notifications. You must understand SQL databases and the formatting of the SQL file to update the Database Service nodes with the correct information. The information below does not cover all available options for these tables. For more information, see the SymmetricDS Documentation (https://www.symmetricds.org/docs/overview) website.

- “Understanding the SYM_MONITOR Table” on page 3
- “Understanding the SYM_MONITOR_EVENT Table” on page 4
- “Understanding the SYM_NOTIFICATION Table” on page 4
- “Understanding the SYM_PARAMETER Table” on page 5

Understanding the SYM_MONITOR Table

The SYM_MONITOR table defines what SymmetricDS monitors in the database. The following information defines the available columns for this table.

MONITOR_ID

Specify a unique ID for the monitor.

NODE_GROUP_ID

Specify `ALL` or specify the node group, if appropriate.

EXTERNAL_ID

Specify `ALL` or specify the external ID of the node.

CREATE_TIME

View the time stamp of when SymmetricDS created this entry.

LAST_UPDATE_TIME

View the time stamp of when SymmetricDS last updated this entry.

TYPE

Specify the item you want to check or monitor and then specify a numeric value that is equal to or greater than the threshold of that item. SymmetricDS creates an event when the threshold is met. The following are examples of the items you can check or monitor:

cpu

Percentage from 0 to 100 of CPU usage for the server process.

disk

Percentage from 0 to 100 of disk usage (`tmp` folder staging area) available to the server process.

memory

Percentage from 0 to 100 of memory usage (tenured heap pool) available to the server process.

batchError

Number of incoming and outgoing batches in error.

batchUnsent

Number of outgoing batches waiting to be sent.

dataUnrouted

Number of change capture rows that are waiting to be batched and sent.

dataGaps

Number of active data gaps that are being checked during routing for data to commit.

offlineNodes

Number of nodes that are offline based on the last heartbeat time. The `console.report.as.offline.minutes` parameter controls how many minutes might pass before a node is considered offline.

THRESHOLD

Specify an integer that is a metric of the **TYPE** at which, when reached, SymmetricDS records an event.

RUN_PERIOD

Specify the time in seconds between runs of the monitor. For example: `300`

RUN_COUNT

Specify the average the value across a number of runs before checking threshold.

SEVERITY_LEVEL

Specify an integer that SymmetricDS uses to differentiate which monitors, for their respective thresholds, are severe enough to trigger the specified notifications. Example: `3`

ENABLED

Adding the ENABLED column enables the monitor in the SYS_MONITOR table.

Understanding the SYM_MONITOR_EVENT Table

SymmetricDS automatically populates the SYM_MONITOR_EVENT table when the threshold of a monitor is met or exceeded. You do not have to configure any options for the SYM_MONITOR_EVENT table. You can use this table as a log of the events that have occurred.

Understanding the SYM_NOTIFICATION Table

In the SYM_NOTIFICATION table, you define which monitor events SymmetricDS sends out in a notification. You also define how SymmetricDS sends the notifications. You either define to send the notifications via SMTP or you can define to send the monitor events to a log file. The following information defines the available columns for this table.

NOTIFICATION_ID

Specify `ALL` or specify the node group, if appropriate.

NODE_GROUP_ID

Specify `ALL` or specify the node group, if appropriate.

EXTERNAL_ID

Specify `ALL` or specify the external ID of the node.

EXPRESSION

Specify a comma-separate list of the email addresses for the recipients of the notification email.

CREATE_TIME

View the time stamp of when SymmetricDS created this entry.

LAST_UPDATE_TIME

View the time stamp of when SymmetricDS last updated this entry.

SEVERITY_LEVEL

Specify the corresponding (or greater) severity level for which this notification should monitor the SYM_MONITOR_EVENT table. If an event with the appropriate severity level is found, SymmetricDS sends the event via this notification.

TYPE

Specify either `'email'` (using SMTP parameters defined in SYM_PARAMETERS) or `'log'`.

ENABLED

Adding the ENABLED column enables the notification for the monitor.

Understanding the SYM_PARAMETER Table

The SYM_PARAMETER table contains the parameters that you use to define the SMTP server you want to use to send emails. The following information defines the available columns for this table.

EXTERNAL_ID

Specify `ALL` or specify the external ID of the node.

NODE_GROUP_ID

Specify `ALL` or specify the node group, if appropriate.

PARAM_KEY and PARAM_VALUE

Specify the PARAM_KEY and PARAM_VALUE that are appropriate for your environment. You must enter both values together. All of the values of the PARAM_KEY entries are suggested values; they do not auto-populate when creating a record. Here is a list of the PARAM_KEY and PARAM_VALUES you use to define the SMTP server or a log file:

smtp.allow.untrusted.cert

This value determines whether or not to accept an untrusted certificate for SSL/TLS when connecting to the mail server. Default: `false`

smtp.auth

This value determines whether or not to authenticate with the mail server. Default: `false`

smtp.from

This value contains the email address to use in the "from" header when sending an email.

Default: `symmetricds@localhost`

smtp.host

This value contains the host name of the mail server. Default: `localhost`

smtp.password

This value contains the password when SymmetricDS authenticates with the mail server. There is no default value. You must specify the password for the SMTP server.

smtp.port

This value contains the port number of the mail server. Default: `25`

smtp.starttls

This value determines whether or not to use TLS after connecting to the mail server. Default: `false`

smtp.transport

This value contains the transport type to use when connecting to the mail server, either `smtp` or `smtps`. Default: `smtp`

smtp.user

This value contains the user name to use when authenticating with the mail server. There is not a default value. You must specify a user name.

Configuring and Implementing the SQL File

1. Open the `email-notification.sql` file in a text editor.
2. Use the information about the tables to populate the file with the information that is appropriate for your environment. For more information, see "Understanding the SQL File" on page 2.
3. Save the `email-notification.sql` file.
4. Use an SQL client and log in to the appliance using the database user name and password you creating when deploying the first Database Service node. It is the same user name and password that you use to add a new node to the Secure API Manager system.
5. Run the `email-notification.sql` file against the primary database of the Database Service cluster.

Testing the Notifications

After you have configured the `email-notification.sql` file and have imported the file to the Database Service, you can test the email notifications.

1. Power down the master node.
2. Check the email you configured to receive the notifications.
3. Power on the master node.