Database Access For Development Purposes

ver 2012-08-07

Contents

Overview

Nagios XI comes as a ready to go monitoring system delivered as a virtual machine (VM). I have spent a bit of time developing Configuration Wizards, Components and Dashlets for Nagios XI and during this time I have needed to look at the backend databases to understand where data is stored and how.

I am very much well a graphical user interface (GUI) kind of person, so when it comes to test and development, I will use a GUI if I can. The purpose of this document is to show you how you can access the backend databases from a Windows 7 PC using two different database management applications.

Specifically you need to perform some configuration steps for each database system to allow access from the Windows 7 PC.

The Databases

There are two different database systems running on Nagios XI.

MySQL

This database system is used for Nagios Core. The data that is stored in this database is for all of the monitoring configurations like host and service object definitions.

PostgreSQL

This database system is used for the Nagios XI configurations. Things like user accounts and dashlet settings are just a few examples of what is stored in this database.

Requirements

Nagios XI VM

This document is based on a running virtual machine of Nagios XI downloaded from the Nagios Enterprise website. It is recommended to be running the most up to date release version of Nagios XI. This guide may also be helpful for admins who have deployed Nagios XI manually in their environment, however keep in mind that your configuration might be slightly different to the Nagios XI VM so some steps may not be verbatim.

At the time of writing this document I was using a Nagios IX 2011R3.2 VM.

Applications

There are a couple of programs I use throughout this guide:

- Putty is used to remotely access your Nagios XI server to perform console commands ○ http://www.chiark.greenend.org.uk/~sgtatham/putty/
- MySQL Workbench is the GUI I use for connecting to the MySQL databases ○ http://www.mysql.com/products/workbench/ ○ Version 5.2.33 was used when I created this document

- pgAdmin III is the GUI I use for connecting to the PostgreSQL databases ○ http://www.pgadmin.org ○ Version 1.14.3 was used when I created this document
- nano is a text based editor that I use in the CentOS shell ○ I have steps below showing you how to make sure nano is installed

You will need to download these apps and install them before proceeding.

Putty

The first time you connect to the Nagios XI server it will present a warning dialog box about the Nagios XI host not being in the local cache, click Yes and you will not be prompted again.

I highly recommend that you use Putty for all of your shell commands. The reason why is that copy and paste is your friend … all the steps here can be copied to the Putty session to save you time and typing errors.

nano

I really like the text editor nano, it's much easier to use than vi so I always install it. Sometimes it is already installed in the Nagios XI VM so this step is not always necessary.

- Establish a Putty session to your Nagios XI VM
- Username: root
- Password: Your root Password (nagiosxi by default)
- Type yum install nano and press Enter
- Wait while nano is installed
- You can now run nano by typing nano and pressing Enter

This completes installing nano.

MySQL

Configuring MySQL

- Establish a Putty session to your Nagios XI VM
- Username: root
- Password: Your root Password (nagiosxi by default)

Firstly we need to edit the MySQL config and define the IP address of the Nagios XI VM. My Nagios XI VM uses the IP address 192.168.100.64 so this is what I will use in the steps below.

- Type nano /etc/my.cnf and press Enter ○ Under the [mysqld] section we need to add the following line bind-address=192.168.100.64
  - Press Ctrl + X ○ Type y ○ Press Enter
- Additionally we need to restart the mysqld service

- Type /etc/init.d/mysqld restart and press Enter

Now we must add a user account to the MySQL database to allow remote access. I will be using the username troy and the password troypassword .

- Type mysql -u root -p mysql and press Enter ○ When prompted for a password the default password is nagiosxi ○ Type CREATE USER 'troy'@'%' IDENTIFIED BY 'troypassword'; and press Enter ○ Type GRANT ALL PRIVILEGES ON *.* TO 'troy'@'%' WITH GRANT OPTION; and press Enter ○ Type CREATE USER 'troy'@'localhost' IDENTIFIED BY 'troy'; and press Enter
  - ○ Type GRANT ALL PRIVILEGES ON *.* TO 'troy'@'localhost' WITH GRANT OPTION; and press Enter ○ Type exit and press Enter

Last step is to allow the CentOS firewall inbound traffic on port 3306.

- Type /sbin/iptables -I INPUT -p tcp --destination-port 3306 -j ACCEPT and press Enter • Type service iptables save and press Enter

Connect To MySQL

- Open MySQL Workbench

First we must define a connection.

- Click the pull down menu Database and select Manage Connections ○ Click the New button
  - ○ Connection Name: My Nagios XI VM ○ Connection Method: Standard (TCP/IP) ○ Parameters tab
    - Hostname: 192.168.100.64
    - Port: 3306
    - Username: troy
    - click the Store in Vault ... button
- Password: troypassword
- Click OK
    - Default Schema: leave this blank ○ Click the Test Connection button
    - If everything is correct you will receive the message connection parameters are correct Click OK ○ Click the Close button

Now we can connect to the MySQL database on the Nagios XI VM.

- Click the pull down menu Database and select Query Database

○ Stored Connection: My Nagios XI VM ○ Click OK

- You are now connected to the MySQL database on your Nagios XI VM.

This completes the MySQL chapter.

PostgreSQL

Configuring PostgreSQL

- Establish a Putty session to your Nagios XI VM
- Username: root
- Password: Your root Password (nagiosxi by default)

Firstly we need to edit the PostgreSQL config and define the subnet that my Windows 7 PC is on, which is 10.254.1.0/24 in this example.

- Type nano /var/lib/pgsql/data/postgresql.conf and press Enter ○ Under the # CONNECTIONS AND AUTHENTICATION section we need to add the following line (press page down three times and you should be at that section) listen_addresses = '*'

  ○ Press Ctrl + X ○ Type y ○ Press Enter

- Type nano /var/lib/pgsql/data/pg_hba.conf and press Enter ○ Under the # IPv4 local connections: section we need to add the following line (press page down three times and you should be at that section) host all all 10.254.1.0/24 trust

  ○ Press Ctrl + X ○ Type y ○ Press Enter

- Additionally we need to restart the postgresql service
- Type /etc/init.d/postgresql restart and press Enter

Last step is to allow the CentOS firewall inbound traffic on port 5432.

- Type /sbin/iptables -I INPUT -p tcp --destination-port 5432 -j ACCEPT and press Enter • Type service iptables save and press Enter

Connect To PostgreSQL

- Open pgAdmin III

First we must define a connection.

- Click the pull down menu File and select Add Server ○ Properties tab

Name:	My Nagios XI VM
Host:	192.168.100.64
Port:	5432
Service:	leave this blank
Maintenance DB:	postgres
Username:	nagiosxi
Password:	nagiosxi

- This is the default password in the Nagios XI VM

Store password:	tick ○

Click OK

- You will now be connected and your connection will appear in the left pane under Object browser

Every other time you open pgAdmin III, double click on the My Nagios XI VM server under the Object browser in the left pane.

This completes the PostgreSQL chapter.