PANDORAFMS
ENTERPRISE

Pandora FMS
Administrator Manual
Cassandra DB monitoring

Table of Contents

1. Introduction....3
2. Compatibility Matrix....3
3. Modules generated by the plugin....3
4. Requisites....5
5. Specific Cassandra Configuration....6
6. File Configuration....7
7. Plugin Execution....7

1. Introduction

The main objective of this document is the description of the CassandraDB database monitoring procedures on Linux-based systems. We have chosen a set of “base” modules based on our experience in system monitoring and on the needs of some of our clients.

To extract the information we use:

- The basic configuration files of CassandraDB: Installation of cqlsh client is required. Correct version of Python must be installed. Cqlshlib is a necessary library for cqlsh.
- The general log files for Cassandra
- It uses the software already installed in the system (ps, gawk, grep, etc), for the monitoring done by the plugin without having to install libraries or third party tools.
- It uses Nodetool to monitor the general performance of the database. Nodetool needs at least version 7 of java to work correctly.

2. Compatibility Matrix

The compatibility matrix for CassandraDB monitoring is:

Systems where it has been tested	• CassandraDB v2.0.0

Systems where it should work	• CassandraDB v2.0.0 or above

3. Modules generated by the plugin

This plugin returns 32 modules by instance. All of them display valuable status information. You can fix thresholds manually to determine whether something is in a warning/critical or operative condition.

- Cassandra_Process: Displays information on the status of the cassandra process.

- Cassandra_CPU_Usage: Displays the cpu use for the cassandra process in %.
- Cassandra_Memory_Use: Displays the memory use for the cassandra process in %.
- Cassandra_Error_Log_Messages: Displays the amount of error messages in the cassandra log.
- Cassandra_Warning_Log_Messages: Displays the amount of warning messages in the cassandra log.
- Cassandra_Network_Connections: Displays the amount of network connections by clients.
- Cassandra_Thrift_Server_Status: Displays the status of the thrift server.
- Cassandra_Cluster_Status : Displays the status of the main Cluster.
- Cassandra_Key_Cache_Size: Displays the size of the key cache in kbs.
- Cassandra_Active_Commands: Displays the amount of active commands or task being executed.
- Cassandra_Pending_Commands: Displays the amount of pending commands.
- Cassandra_Completed_Commands: Displays the amount of completed commands.
- Cassandra_Active_Responses: Displays the amount of active responses to commands executed by cassandra.
- Cassandra_Pending_Responses: Displays the amount of pending responses to commands.
- Cassandra_Completed_Responses: Displays the amount of completed responses to commands.
- Cassandra_ReadStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the ReadStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_RequestResponseStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the RequestResponseStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_MutationStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the MutationStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_ReadRepairStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the ReadRepairStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_ReplicateOnWriteStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the ReplicateOnWriteStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_GossipStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the GossipStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_AntiEntropy_Pool_Pending_Tasks: Displays the amount of pending tasks for the AntiEntropyStage pool. If the amount is very high it can influence badly the performance of

Cassandra.

- Cassandra_MigrationStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the MigrationStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_MemoryMeter_Pool_Pending_Tasks: Displays the amount of pending tasks for the MemoryMeter pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_MemtablePostFlusher_Pool_Pending_Tasks: Displays the amount of pending tasks for the MemtablePostFlusher pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_FlushWriter_Pool_Pending_Tasks: Displays the amount of pending tasks for the FlushWriter pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_MiscStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the MiscStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_PendingRangeCalculator_Pool_Pending_Tasks: Displays the amount of pending tasks for the PendingRangeCalculator pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_ReplicateOnWriteStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the ReplicateOnWriteStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_Comitlog_archiver_Pool_Pending_Tasks: Displays the amount of pending tasks for the Comitlog_archiver pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_InternalResponseStage_Pool_Pending_Tasks: Displays the amount of pending tasks for the InternalResponseStage pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_HintedHandoff_Pool_Pending_Tasks: Displays the amount of pending tasks for the HintedHandoff pool. If the amount is very high it can influence badly the performance of Cassandra.
- Cassandra_Nodetool_Configuration: Detects if Nodetool is working properly. Normally Nodetool connects to the port 7199. If there isn't any connection established with that port then Nodetool isn't working correctly. A check of the java version if this module goes critical is necessary.

4. REQUISITES

The CassandraDB monitoring is done basically getting data through the execution of commands in the Operative System.

The requisites for this monitoring to work correctly are the followings:

- Cassandra db must be installed.
- A local pandora agent must be installed.
- The Cassandra configuration files must be located in the following path:

  /etc/cassandra/conf/
- The Cassandra log files must be located in the following path:

  /var/log/cassandra/

5. Specific Cassandra Configuration

This plugin requieres some specific cassandra configuration. Normally when cassandra is set up this is the normal configuration and setup that permits the correct management of the database.

First of all start by checking the java version. This plugin is based uppon nodetool to extract generic data from the system. Nodetool needs version 7 of Java. To check the Java version open a terminal and type:

java – version

Then it should show the following result:

java version "1.7.0_51"
OpenJDK Runtime Environment (rhel-2.4.4.1.el6_5-i386 u51-b02)
OpenJDK Client VM (build 24.45-b08, mixed mode, sharing)

If the version isn't the one from the example then you'd need to install the following one:

$ su -c "yum install java-1.7.0-openjdk"

The java-1.7.0-openjdk package contains just the Java Runtime Environment. If you want to develop Java programs then install the java-1.7.0-openjdk-devel package.

It is asumed that cassandra was installed from the datastax repo. From version 1.1 of cassandra cqlsh is installed alongside. There is a common error that can occure when trying to access cassandra using cqlsh. Normally happens when the version of python doesn't include cqlshlib. This error can occure even during the installation of cqlsh.
To check if the library cqlshlib is installed open a terminal and type the following:

pydoc modules | grep cqlshlib

If you don't have this library installed follow the next few steps to do so.
1. Download the setup.py file from the following url:
svn.apache.org/repos/asf/cassandra/trunk/pylib/setup.py

To do so just open a terminal and type:
wget svn.apache.org/repos/asf/cassandra/trunk/pylib/setup.py

2. Download cqlshlib from the following url:
`svn.apache.org/repos/asf/cassandra/trunk/pylib/cqlshlib/`

To do so just open a teminal and type:
wget svn.apache.org/repos/asf/cassandra/trunk/pylib/cqlshlib/

3. Then type:
python setup.py install.

Once installed cassandra should work correctly.

6. File Configuration

In order to configure correctly this plugin there are a few steps that should be followed:

- Move the cassandra.sh file from the default download directory to the /etc/pandora/plugins/ directory:

  Example of the command inside the default download directory:

  mv cassandra.sh /etc/pandora/plugins/
- Assign necessary permitions to cassandra.sh script:

  Example of the command inside the plugins directory:

  chmod +x cassandra.sh
- At the end of the pandora_agent.conf file add the following line:

  module_plugin /etc/pandora/plugins/cassandra.sh
- Restart the pandora agent process

7. Plugin Execution

Once done with the installation process of the plugin 32 modules will be automatically created in the agent where cassandra is running.

If everything is working fine you should be able to see the following image with all the modules created.

			Cassandra_Cluster_Status	Status of the cassnadra main cluster		N/A - N/A	0
			Cassandra_CPU_Usage	% of CPU used by cassandra		N/A - N/A	1
			Cassandra_Error_Log_Messages	Error messages in the cassandra log		N/A - N/A	0
			Cassandra_Memory_Use	% of memory used by the cassandra process		N/A - N/A	0.1
			Cassandra_Network_Connections	Amount of network connections established		N/A - N/A	0
			Cassandra_Warning_Log_Messages	Warning messages in the cassandra log		150/70 - 0/150	
			CPU Free	Percentage CPU free		N/A - N/A	10
			cpu_user	User CPU Usage (%)		90/70 - 100/91	29
			Cron task files	Number of cron task files		N/A - N/A	3
			http://localhost/pandora_conso...			N/A - N/A	0

This is what the module view should look like once the modules obtain data.

It is posible to see that the modules are being created but don't initialize.

In this case you'd need to check the Specific Cassandra Configuration.