Nagios Log Server – Analyzing Logs

Nagios®

The Industry Standard in IT Infrastructure Monitoring

Purpose

This document describes how to use queries and filters to drill down to see the exact information you are looking for using the Nagios Log Server search dashboard.

Target Audience

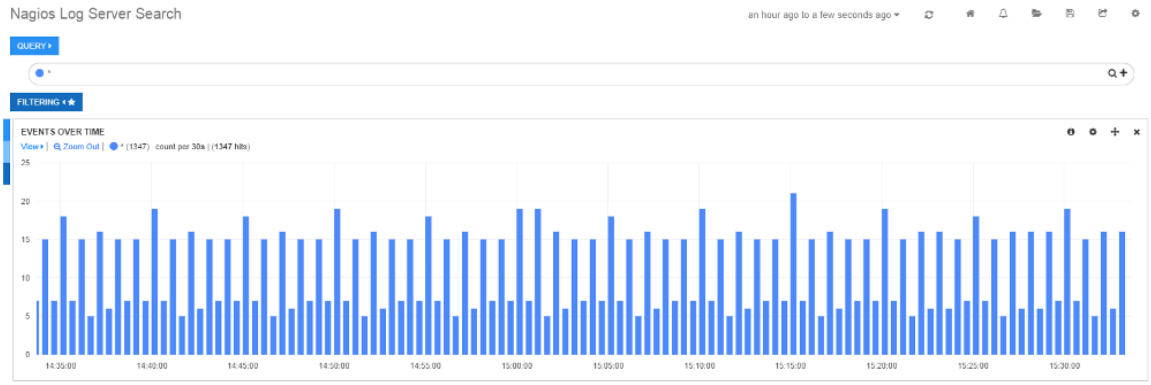
This document is intended for use by Nagios Log Server administrators and users looking for information on querying, filtering and drilling down the data in Nagios Log Server.

Analyzing Your Log Data

Queries QUERY

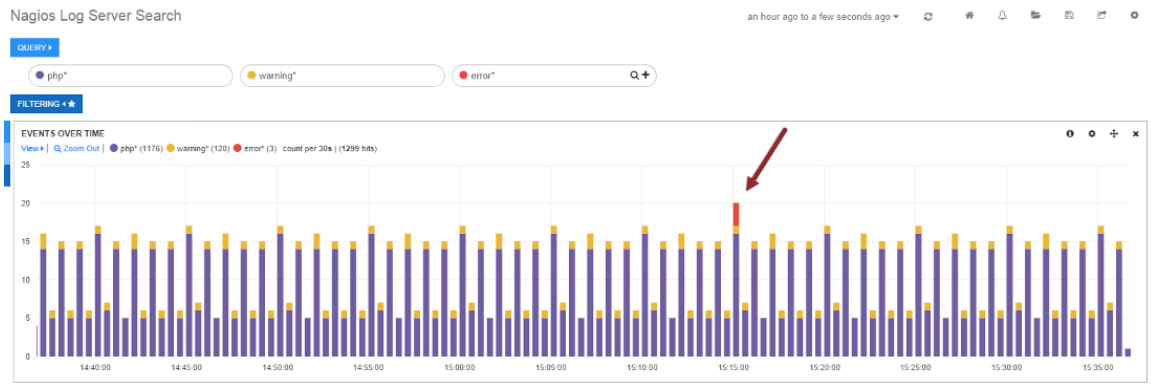
When you start collecting log data over a long period of time you will want to look at certain log types and categories. Nagios Log Server queries can show you exactly what you are looking for.

Here is what a default view of the dashboard inside Nagios Log Server:



This graph view is showing us all the log data the server is receiving since the default query is an asterisk **'*'** which will display all log data in the database. Through this view you can see the log data traffic and trends in a somewhat birds eye view for the last hour.

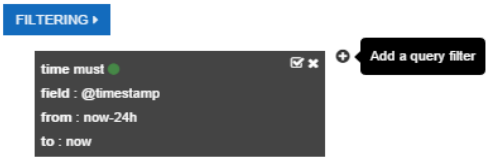
Below is a table that is specifically querying PHP log files and any critical and warning statuses that may be inside those logs. Set each search string as a different color to paint a picture of the PHP log files. When you query Nagios Log Server will check every row and column for the search string. Also queries are **not** case sensitive so in this search php is the same as PHP.



The PHP query show us all the warning files and what portion of the php files have a warning status. The point of interest in the graph above is where a log file contains a critical status **(red arrow)**. This can tell us what time and where to look for a possible source of a problem.

Filters FILTERING

Filter can be just as useful as querying log files. Setting up filters is very simple and you can make new filters by simpling clicking the 'FILTERING' button and clicking the plus '+' symbol next to the current views filter.

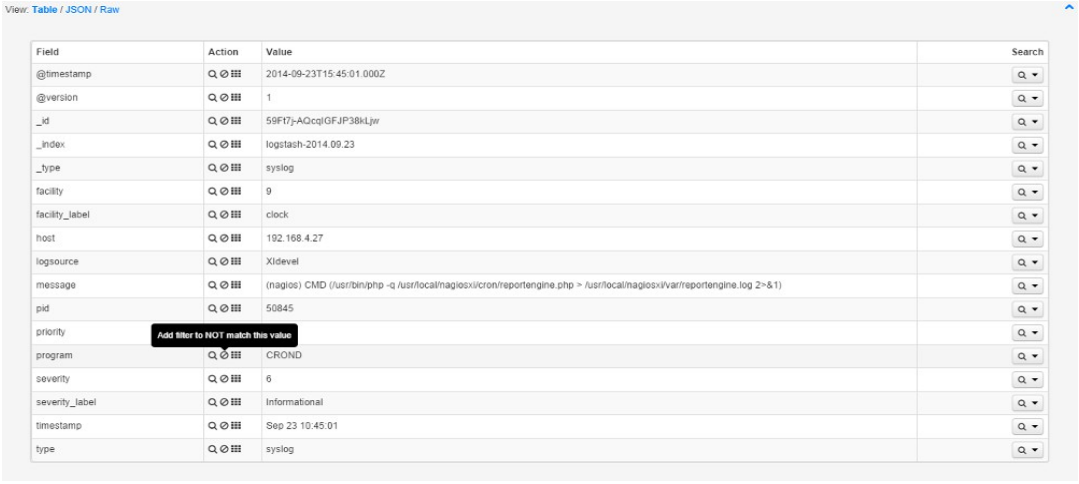


The filter you create here will be applied to the current query that is applied to the dashboard. This will help in finding specific parts of a set of log data. For example, our PHP graph above could have a filter to show only PHP notices. So we could make a filter that says **must contain notice.** Then we could see what PHP notices are present in our log files.

You can also filter based on the data you see inside the table. You can view a log inside the dashboard table and select 'Add Filter to NOT match this value' or 'Add Filter to MATCH this value.' In this example we are getting messages from a Nagios Server. They are PHP logs, but are coming from the CROND program. Since this is messing up our regular report we can make a NOT filter for the program CROND.

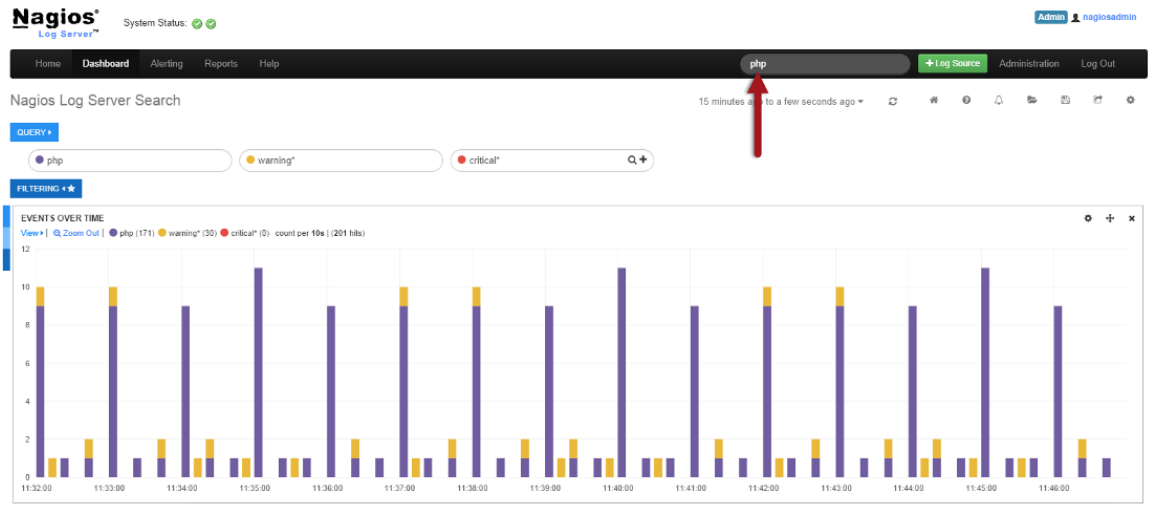
Here is a view of a logs inside the table:

- Select the spyglass to make a filter to MATCH based on the contents of a row.
- Select the circle with an X through it to create a filter to NOT match any row contents.



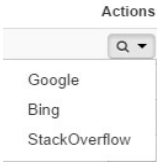
If we want CROND to not show up in our results we can click the NOT match. Then the query will update with the new filter parameters. Make sure to save the current dashboard after you drill down to the view you want. Then you can always load that view again or even set a new default dashboard to load every time you go to the dashboard page.

You can also **highlight** a search string inside the table to get a fast focus on key search terms. Here we will search for 'php' and add our warning and critical query terms again. Then we will see php highlighted in our table.



Here is the table with the highlighted search terms **php**, **warning** and **error.**

Additionally, you can take actions on specific logs by clicking the spyglass in the actions column in the log table. You can select a search engine to look for solutions to problems or just get more information about the log in question.



Dashboard Controls

The view you see when selecting the 'Dashboard' section on the navigation bar of Nagios Log Server is how you will view incoming logs from the sources you have set up. There are default dashboards, but you can also customize them based on your needs. Navigate to your dashboard, the view you see is the current default dashboard of you Log Server.

Here are the dashboard control buttons, these allow you to configure, save, load and navigate through dashboards:

The first button will take you to the saved **default dashboard**. This can be changed by selecting the Save button > Advanced > Set as Default Dashboard.

The second button is for managing **queries**. Queries are what dashboards will display. If you make a query of '*', also known as the wild card selector, it will display ALL of the logs in your database. Here you can create or import queries which can make them easy to share among users. You can also make queries global or local based on who you want to see the queries.

The third button is for creating new **alerts** which is covered in the Nagios Log Server alerting documentation.

The fourth button is the **Load** button. This is where you can load any dashboard that you have created. It will show a drop down menu with all the dashboards that you have saved and all you have to do is click the dashboard to display it in the page. You can also delete a dashboard from the load drop down by selecting the 'X' to the left of the dashboard title. Use this cautiously since you won't be able to recover a deleted dashboard after you confirm deletion.

The fifth button is the **Save** button. After you set the queries as you want them, configure the dashboard graphs and tables, configure the graph colors and highlighting you will want to save your progress. You will also use this button to name the dashboard you are in by adding a title before saving.

Here is the message you will receive after saving a dashboard:

We saved 'My Dashboard' but we want to make a change to it now. So we can edit the query as we would normally and as changes are made the save button will turn red and there will be a 'Not Saved' message at the top of the dashboard view. This will ensure that you know that the dashboard hasn't been saved and will not keep the changes you just made until you save it again.

For our My Dashboard example we want to add an additional query for error:

Once we add the query to the current queries the graph will update and there will be a notification that a previously saved dashboard has been edited and needs to be saved again:

The 'Not Saved' indicator will appear and the save button in our dashboard controls will turn from green to red. Now we know to save our dashboard if we want to retain our new error query.

Finishing Up

If you have any questions about querying or analyzing your log data on Nagios Log Server, contact our support team via our online forum at:

http://support.nagios.com/forum