MarkLogic Server

Monitoring MarkLogic Guide

MarkLogic 8
February, 2015

Last Revised: 8.0-1, February, 2015

Table of Contents

Monitoring MarkLogic Guide

1.0 Monitoring MarkLogic Server

MarkLogic Server provides a rich set of monitoring features that include a pre-configured monitoring dashboard, a plugin that allows you to monitor MarkLogic Server with Nagios, and a Management API that allows you to integrate MarkLogic Server with existing monitoring applications or create your own custom monitoring applications.

This chapter includes the following sections:

1.1 Overview

In general, you will use a monitoring tool for the following:

- To keep track of the day-to-day operations of your MarkLogic Server environment.
- For initial capacity planning and fine-tuning your MarkLogic Server environment. For details on how to configure your MarkLogic Server cluster, see the *Scalability, Availability, and Failover Guide*.
- To troubleshoot application performance problems. For details on how to troubleshoot and resolve performance issues, see the *Query Performance and Tuning Guide*.
- To troubleshoot application errors and failures.

The monitoring metrics and thresholds of interest will vary depending on your specific hardware/software environment and configuration of your MarkLogic Server cluster. This chapter lists some of the metrics of interest when configuring and troubleshooting MarkLogic Server. However, MarkLogic Server is just one part of your overall environment. The health of your cluster depends on the health of the underlying infrastructure, such as network bandwidth, disk I/O, memory, and CPU.

1.2 Selecting a Monitoring Tool

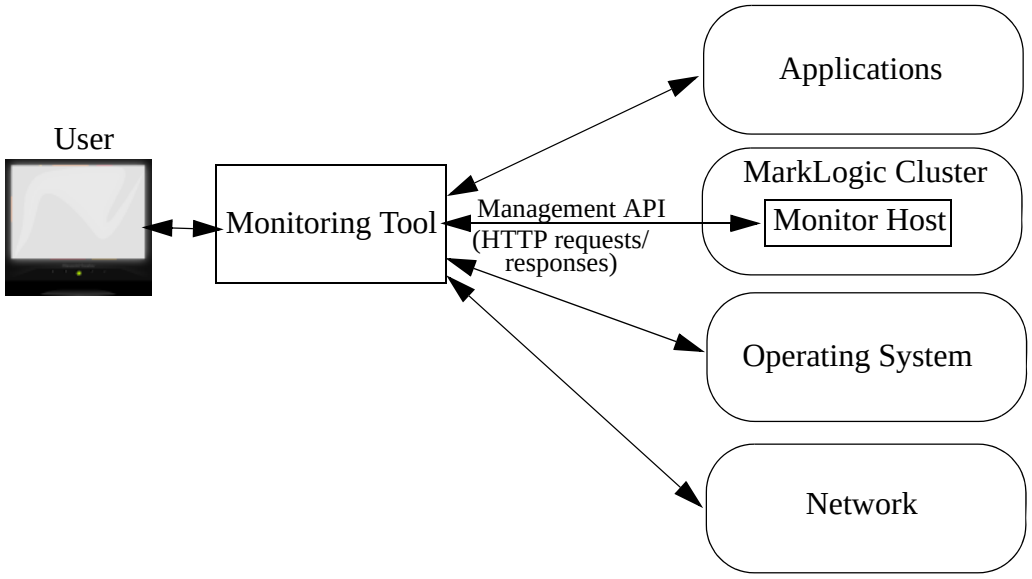
Though this guide focuses on the tools available from MarkLogic that enable you to monitor MarkLogic Server, it is strongly recommended that you select an enterprise-class monitoring tool that monitors your entire computing environment to gather application, operating system, and network metrics alongside MarkLogic Server metrics.

There are many monitoring tools on the market that have key features such as alerting, trending, and log analysis to help you monitor your entire environment. MarkLogic Server includes the following monitoring tools:

- A Monitoring dashboard that monitors MarkLogic Server. This dashboard is pre-configured to monitor specific MarkLogic Server metrics. For details, see “Using the MarkLogic Server Monitoring Dashboard” on page 16.
- A Monitoring History dashboard to capture and make use of historical performance data for a MarkLogic cluster. For details, see “MarkLogic Server Monitoring History” on page 31.
- A plugin that enables Nagios to monitor MarkLogic Server. Nagios can be configured to monitor any and all objects in MarkLogic Server and allows you to set alerts on object thresholds. Nagios is freeware and has a large support community and so it is a good default choice if you do not already have a preferred monitoring tool. For details, see “Configuring Nagios to Monitor MarkLogic Server” on page 63.
- A RESTful Management API that you can use to integrate MarkLogic Server with existing monitoring application or create your own custom monitoring applications. For details, see “Using the Management API” on page 89.

1.3 Monitoring Architecture, a High-level View

All monitoring tools use a RESTful Management API to communicate with MarkLogic Server. The monitoring tool sends HTTP requests to a monitor host in a MarkLogic cluster. The MarkLogic monitor host gathers the requested information from the cluster and returns it in the form of an HTTP response to the monitoring tool. The Management API is described in “Using the Management API” on page 89.



1.4 Monitoring Tools and Security

To gain access to the monitoring features described in this guide, a user must be assigned the `manage-user` role. Monitoring tools should authenticate as a user with that role. The `manage-user` role is assigned the `http://marklogic.com/xdmp/privileges/manage` execute privilege and provides access to the Management API, `Manage` App Server, and the UI for the Configuration Manager and Monitoring Dashboard. The `manage-user` role also provides read-only access to all of a cluster's configuration and status information, with the exception of the security settings. For details on assigning roles to users, see Users in the *Administrator's Guide*.

If you have enabled SSL on the `Manage` App Server, your URLs must start with HTTPS, rather than HTTP. Additionally, you must have a MarkLogic certificate on your browser, as described in Accessing an SSL-Enabled Server from a Browser or WebDAV Client in the *Administrator's Guide*.

1.5 Guidelines for Configuring your Monitoring Tools

Monitoring tools, such as Nagios, enable you to set thresholds on specific metrics to alert you when a metric exceeds a pre-specified value.

The topics in this section are:

- Establish a Performance Baseline
- Balance Completeness Against Performance

1.5.1 Establish a Performance Baseline

Many metrics that can help in alerting and troubleshooting are meaningful only if you understand normal patterns of performance. For example, monitoring an App Server for slow queries will require a different threshold on an application that spawns many long-running queries to the task server than on an HTTP App Server where queries are normally in the 100 ms range. Most enterprise-class monitoring tools support data storage to support this type of trend analysis. Developing a starting baseline and tuning it if your application profile changes will yield better results for developing your monitoring strategy.

1.5.2 Balance Completeness Against Performance

The templates provided with the Nagios integrations consist of a mix of MarkLogic metrics that are useful in many situations for problem-solving, performance tuning, and capacity planning.

Collecting and storing monitoring metrics has a performance cost, so you need to balance completeness of desired performance metrics against their cost. The cost of collecting monitoring metrics can differ. In general, the more resources you monitor, the greater the cost. For example, if you have a lot of hosts, server status is going to be more expensive. If you have a lot of forests, database status is going to be more expensive. In most cases, you will use a subset of the available monitoring metrics. And there may be circumstances in which you temporarily monitor certain metrics and, once the issue have been targeted and resolved, you no longer monitor those metrics.

One balancing technique is to measure system performance on a staging environment under heavy load, then enable your monitoring tool and calculate the overhead. You can reduce overhead by reducing collection frequency, reducing the number of metrics collected, or writing a Management API plugin to produce a custom view that pinpoints the specific metrics of interest. Each response from the underlying Management API includes an elapsed time value to help you calculate the relative cost of each response. For details on the Management API, see “Using the Management API” on page 89. For details on how to write a Management API plugin, see “Extending Management API with Plugins” on page 98.

1.6 Monitoring Metrics of Interest to MarkLogic Server

Environments and workloads vary. Each environment will have a unique set of requirements based on variables including cluster configuration, hardware, operating system, patterns of queries and updates, feature sets, and other system components. For example, if replication is not configured in your environment, you can remove templates or policies that monitor that feature.

This section provides a set of guiding questions to help you understand and identify the relevant metrics. The topics in this section are:

- Does MarkLogic Have Adequate Resources?
- What is the State of the System Overall?
- What is Happening on the MarkLogic Server Cluster Now?
- Are There Signs of a Serious Problem?

1.6.1 Does MarkLogic Have Adequate Resources?

MarkLogic Server is designed to fully utilize system resources. Many settings, such as cache sizes, are auto-sized by MarkLogic Server at installation.

Some questions to ask are:

- Does MarkLogic Server have enough resources on the host machine? What processes other than MarkLogic Server are running on the host and what host resources do those processes require? When competing with other processes, MarkLogic Server cannot optimize resource utilization and consequently cannot optimize performance.
- Is there enough disk space for forest data and merges? Merges require at least one and one half times as much free disk space as used by the forest data (for details, see Memory, Disk Space, and Swap Space Requirements in the *Installation Guide*). If a merge runs out of disk space, it will fail.
- Is there enough disk space for the log files reside to log system activity? If there is no space left on the log file device, MarkLogic Server will abort. Also, if there is no disk space available to add messages to the log files, MarkLogic Server will fail to start.
- Is there enough memory for the range indexes? Range index improve performance at the cost of memory and increased load/reindex time. Running out of memory for range indexes may result in undesirable memory swapping that severely impacts performance.
- Is swap space configured correctly? At query time, MarkLogic Server makes use of both memory and swap space. If there is not enough of either, the query can fail with SVC-MEMALLOC messages. For details on configuring swap memory, see Tuning Query Performance in MarkLogic Server in the *Query Performance and Tuning Guide*.
- How many hosts are in the cluster? How are the hosts configured as evaluator and data nodes? How are the hosts organized into groups? For details on configuring MarkLogic

Server clusters, see Clustering in MarkLogic Server in the *Scalability, Availability, and Failover Guide*.

- What applications use resource-intensive features, such as CPF, replication, and point-in-time recovery? Are the hardware, software, and network resources available and configured to most efficiently support such applications?

1.6.2 What is the State of the System Overall?

Many problems that impact MarkLogic Server originate outside of MarkLogic Server. Consider the health of your overall environment.

Some questions to ask are:

- How efficiently is CPU being used? How much CPU capacity exists at different time slices? What is the execution speed of the current read and write tasks? Can I optimize queries or choose a better time to batch load?
- How efficiently is I/O being used? What amount of data is currently being read from or written to disk? Are there any I/O bottlenecks?
- Is there enough free disk space for each file system?
- Are there any errors or warnings appearing in the logs for the operating system, MarkLogic Server, and applications?
- What is the current state of the network?
- Are there any serious errors in the system log files? Your monitor tool, or an auxiliary tool such as Splunk, should monitor your system logs and report on any detected errors.

1.6.3 What is Happening on the MarkLogic Server Cluster Now?

When you suspect an error or performance problem originates from MarkLogic Server, some questions to ask are:

- Are all of the hosts in the cluster online? Are all of the App Servers enabled? In what a states are the forests?
- What are the patterns of queries and updates? Do they appear to be evenly distributed across the hosts in the cluster?
- Are there any long-running queries? Longer than usual query execution times may indicate a bottleneck, such as a slow host or problems with XDQP communication between hosts. Other possible problems include increased loads following a failover or more than the usual number of total requests.
- Is there an increase in the number of outstanding requests? A consistent increase in the total number of outstanding requests may indicate the need to add more capacity and/or load balance. Decreases in total requests may indicate some "upstream" problem that needs to be addressed.
- What is the I/O rates and loads pattern? In this context, *rates* refers to amount of data applications are currently reading from or writing to MarkLogic Server databases (throughput) and *loads* refers to the execution time of the current read and write requests, which includes the time requests spend in the wait queue when maximum throughput is achieved.

  Under normal circumstances you will see loads go up as rates go up. As the workload (number of queries and updates) increases, a steadily high rates value indicates the maximum database throughput has been achieved. When this occurs, you can expect to see increasing loads that reflects the additional time requests are spending in the wait queue. As the workload decreases, you can expect to see decreasing loads that reflects less requests in the wait queue.

  If, while the workload is steady, rates decrease and loads increase, something is probably taking away I/O bandwidth from the database. This may indicate that MarkLogic Server has started a background task, such as a merge operation or some process outside of MarkLogic Server is taking away I/O bandwidth.

- What is the journal and save write rates and loads pattern? During a merge, you should see the rates for journal and save writes decrease and the loads increase. Once the merge is done, journal and save writes rates should increase and the loads should decrease. If no merge is taking place, then a process outside of MarkLogic Server may be taking away I/O bandwidth.
- What is the XDQP rates and loads pattern? In this context, *rates* refers to amount of data hosts are currently reading from or writing to other hosts and *loads* refers to the execution time of the current read and write requests, including those in the wait queue. A decrease in rates and an increase in loads may indicate that there is network problem.

- What are the cache hit/miss rates? Lots of cache hits means not having to read fragments off disk, so there is less I/O load. An increasing cache miss rate may indicate a need to increase the cache size, write queries that take advantage of indexes to reduce the frequency of disk reads, or adjust the fragment size to better match that of the queried data.
- How many concurrent updates and reads are in progress? An increase of both updates and reads may indicate that there are queries that are doing too many updates and reads concurrently. The potential problem is lock contention between the updates and reads on the same fragments, which degrades performance.
- How many database merges are in progress? Merges require both I/O and disk resources. If too many database merges are taking place at the same time, it may be necessary to coordinate merges by creating a merge policy or establishing merge blackout periods, as described in Understanding and Controlling Database Merges in the *Administrator's Guide*.
- How many reindexes are in progress? Database reindexing is periodically done automatically in the background by MarkLogic Server and requires both CPU and disk resources. If there are too many reindexing processes going on at the same time, you may need to adjust when reindexing is done for particular databases, as described in Text Indexing in the *Administrator's Guide*.
- How many backups and/or restores are in progress? Backup and restore processes can impact the performance of applications and other background tasks in MarkLogic Server, such as merges and indexing. Backups with point-in-time recovery enabled have an even greater impact on performance. If backup and/or restore processes are impacting system performance, it may be necessary to reschedule them, as described in Backing Up and Restoring a Database in the *Administrator's Guide*.

1.6.4 Are There Signs of a Serious Problem?

If you are encountering a serious problem in which MarkLogic Server is unable to effectively service your applications, some questions to ask are:

- Did MarkLogic Server abort or fail to start? This may indicate that there not enough disk space for the log files on the log file device. If this is the cause, you will need to either add more disk space or free up enough disk space for the log files.
- Is an application unable to update data in MarkLogic Server? This may indicate that you have exceeded the 64-stand limit for a forest. This could be the result of running out of merge space or that merges are suppressed.
- Are queries failing with SVC-MEMALLOC messages? This indicates that there is not enough memory or swap space. You may need to add memory or reconfigure your swap memory, as described in Tuning Query Performance in MarkLogic Server in the *Query Performance and Tuning Guide*
- Are there any forests in the async replicating state? This state indicates that a primary forest is asynchronously catching up to its replica forest after a failover or that a new replica forest was added to a primary forest that already contains content. If a forest has failed over, see Scenarios that Cause a Forest to Fail Over in the *Scalability, Availability, and Failover Guide* for possible causes.
- Are there any serious messages in the error logs? The various log levels are described in Understanding the Log Levels in the *Administrator's Guide*. All log messages at the error level and higher should be investigated, whereas lower-level messages, such as warnings and debug messages are mostly informational. Log messages that indicate a particularly serious problem include:
    - Repeated server restart messages. Possible causes include a corrupted forest, segmentation faults, or some problem with the host's operating system.
    - XDQP disconnect. Possible causes include an XDQP timeout or a network failure.
    - Forest unmounted. Possible causes include the forest is disabled, it has run out of merge space, or the forest data is corrupted.
    - SVC-* errors. These are system-level errors that result from timeouts, socket connect issues, lack of memory, and so on.
    - XDMP-BAD errors. These indicate serious internal error conditions that shouldn't happen. Look at the error text for details and the logs for context and contact MarkLogic Support.

1.7 Default Monitor Metrics

The following table lists the metrics that are monitored by the Nagios default templates, along with their default frequencies and thresholds. Some metrics do not have a default warning or critical threshold because these thresholds are dependent on your specific deployment of MarkLogic Server.

For details on how to define the threshold values of these metrics, see "Configuring Nagios to Monitor MarkLogic Server" on page 63.

Metric	Resource	Default Frequency	Default Warning?	Default Critical?	Note
database-count	Cluster	1 hr	current-count +/-1		Signals change
server-count	Cluster	1 hr	current-count +/-1		Signals change
host-count	Cluster	1 hr	current-count +/-1		Signals change
foreign-cluster-count	Cluster	1 hr	current-count +/-1		Signals change
is-bootstrapped	Foreign Cluster	10 min		0 (false)	Database Replication: Are all bootstrap hosts bootstrapped?
long-running requests (total-requests)	Server	1 min	>0	>=10	
request-rate	Server	1 min			
expanded-tree-cache-hits	Server	1 min			
expanded-tree-cache-misses	Server	1 min			
query-count	Server	1 min			
update-count	Server	1 min			
state	Database	1 min		"unavailable"	

Metric	Resource	Default Frequency	Default Warning?	Default Critical?	Note
failed-masters	Database	10 min		>=1	Failover: Down masters
async-replicating	Database	10 min		>=1	Failover: Forests are catching up
database-replication-active	Database	10 min		0 (false)	Database Replication
foreign-forests-lag-exceeded	Database	10 min	1 (true)		Database Replication
backup-count	Database	1 min			
compressed-tree-cache-hit-rate	Database	1 min			
compressed-tree-cache-miss-rate	Database	1 min			
documents	Database	60 min			
list-cache-hit-rate	Database	1 min			list-cache-hit-rate
list-cache-miss-rate	Database	1 min			list-cache-miss-rate
load-detail/*	Database	1 min			
rate-detail/*	Database	1 min			
merge-count	Database	1 min			
disk-size	Database	10 min			
reindex-count	Database	10 min			
restore-count	Database	10 min			
state-not-open	Host	1 min		>=1	
max-stands-per-forest	Host	1 min	>=35	>=50	
min-capacity	Host	10 min	<=15	<=10	

Metric	Resource	Default Frequency	Default Warning?	Default Critical?	Note
online	Host	1 min		“false”	
compressed-tree-cache-hit-rate	Host	1 min			
compressed-tree-cache-miss-rate	Host	1 min			
list-cache-hit-rate	Host	1 min			
list-cache-miss-rate	Host	1 min			
load-detail/*	Host	1 min			
rate-detail/*	Host	1 min			

2.0 Using the MarkLogic Server Monitoring Dashboard

This chapter describes how to use the Monitoring Dashboard. The Monitoring Dashboard provides task-based views of MarkLogic Server performance metrics in real time. The Monitoring Dashboard is intended to be used alongside the status pages in the Admin Interface and other monitoring tools that monitor application and operating system performance metrics.

The topics in this chapter are:

2.1 Terms Used in this Chapter

The following terms are used in this chapter:

- A *Monitoring Session* is the timeframe since the dashboard page was last refreshed. For example, if you navigate from the Query Execution page to the Rates and Loads page, you have ended the Query Execution session and started the Rates and Loads session.
- A *Monitoring Sample* is a bit of information captured during a refresh interval on a graph. For example, one of the candlesticks captured in the Query Execution graph is a single sample.

2.2 Displaying the Monitoring Dashboard

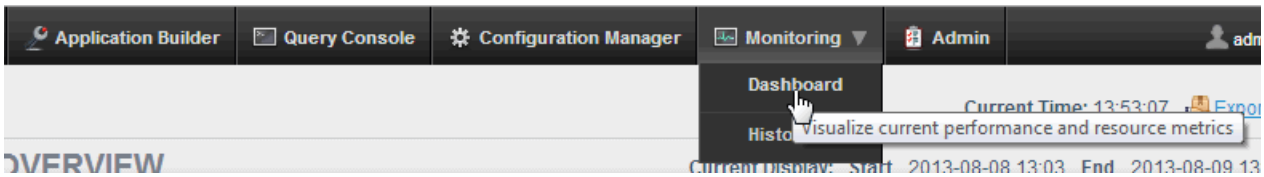
You can display the Monitoring Dashboard by doing the following:

1. Open a browser and enter the URL:

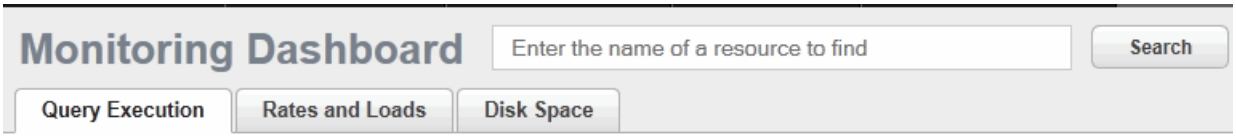
   ```
   http://monitor-host:8002/
   ```

   where *monitor-host* is a host in the cluster you want to monitor

2. At the top of the page, click on Monitoring and click on Dashboard in the pull-down menu:

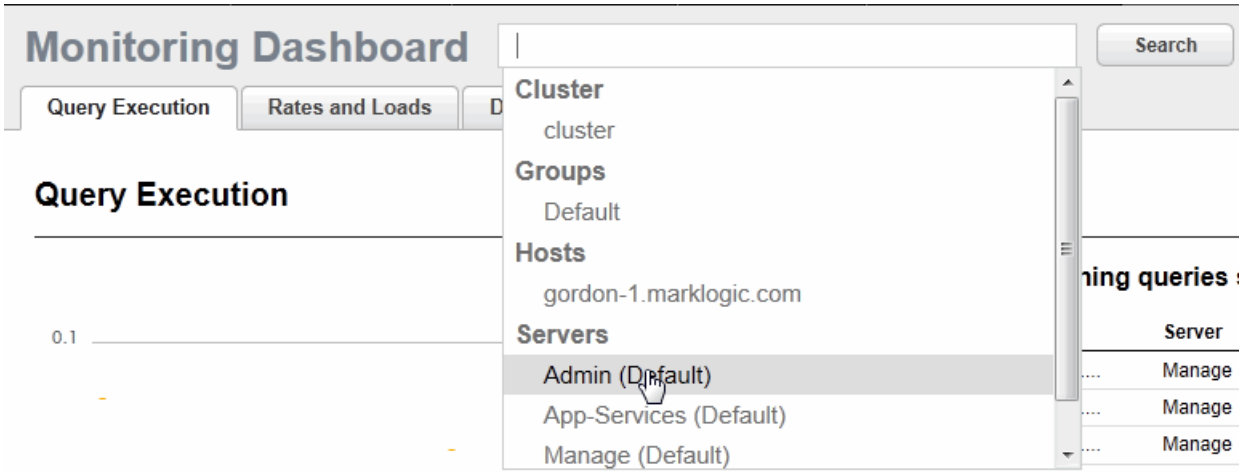


3. The Monitoring Dashboard page appears. From the default Monitoring Dashboard page, you can navigate to any of the pages described in this chapter.



2.3 Monitoring Specific Resources

By default the Monitoring Dashboard monitors the entire cluster. You can use the Search box to select a specific resource to monitor. Clicking on the search field produces a pull-down menu in which you can locate the resource. Alternatively, you can directly locate a resource by entering the name of the resource in the search field.



2.4 Monitoring Dashboard Sessions

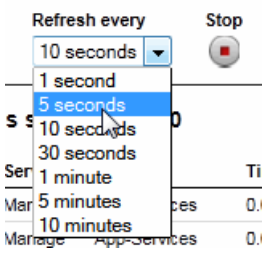
Each time you navigate to a new Dashboard page, you end the current monitoring session and begin a new one. The monitoring data from the previous session is lost from that point on. If you want to maintain multiple Dashboard sessions, you can open each page in a separate browser tab or window.

You can freeze the monitoring data for a Dashboard page by clicking on the Stop button in the upper right-hand portion of the page and restart the data by pressing Start. When you stop a page, you will lose any monitoring data between the time the page is stopped and the time it is restarted. If you have multiple Dashboard pages open, the sessions continue on the other pages; so stopping the monitoring data on one page will not stop the data on the other pages. When you start the stopped page, its session will resume at the current timestamp.

2.5 Setting the Sample Interval

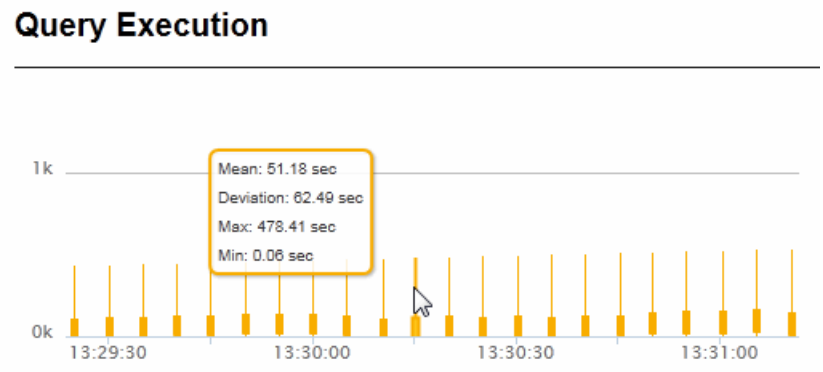
The sample interval specifies the frequency in which the selected resource is monitored. By default, the sample interval is every 10 seconds. Use the Refresh pull-down menu to set the sample interval from anything between once every 1 second to every 10 minutes.

If you have multiple Dashboard pages open in separate tabs or windows, changing the sample interval on one page will not change the interval on the other pages. However, if you switch between pages in the same browser tab or window, the interval will be the same for all pages.



2.6 Viewing Monitoring Sample Details

You can hover your mouse on any monitoring sample to view the details of the sample. For example, to view the details of a query execution sample, hover on the bar graphic as shown below.

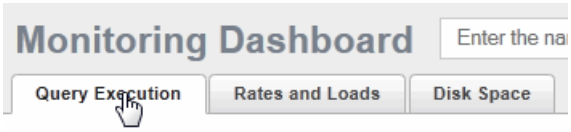


2.7 Monitoring Query Execution

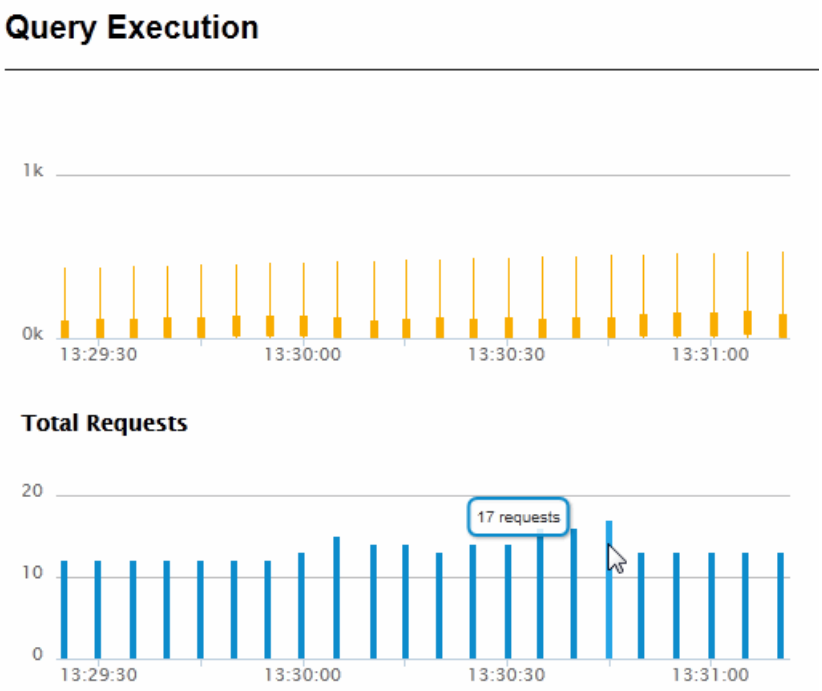
Query execution data gives you insight into the number of queries currently taking place and the execution time of these queries. Two important query execution metrics to monitor are:

- Query Execution Time — Longer than usual query execution times may indicate a bottleneck, such as a slow host or problems with XDQP communication between hosts. Other possible problems include increased loads following a failover or more than the usual number of total requests.
- Total Requests — A consistent increase in the total number of outstanding requests may indicate the need to add more capacity and/or load balance. Decreases in total requests may indicate some “upstream” problem that needs to be addressed.

To display monitoring data related to query execution, select the Query Execution tab in the top left-hand portion of the Monitoring Dashboard.



The left side of the Query Execution page displays the maximum execution time (in seconds) of the current queries and the number of requests captured at each sample interval. You can hover a query execution sample to view the mean, maximum, and minimum execution times and the standard deviation from the mean.



The right side of the Query Execution page displays the five longest running queries since the beginning of the session and the longest running queries at the current time.

5 longest running queries since 13:29:24

Host	Server	Module	Time
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	533.47s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	523.46s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	518.46s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	513.45s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	503.46s

Longest running queries at 13:31:07

Host	Server	Module	Time
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	533.47s
gordon-3.marklogic....	Documents...	/apply.xqy	124.65s
gordon-3.marklogic....	TaskServer	...ansaction-manager.xqy	72.71s
gordon-1.marklogic....	TaskServer	...push-local-forest.xqy	65.87s
gordon-1.marklogic....	TaskServer	...push-local-forest.xqy	64.61s

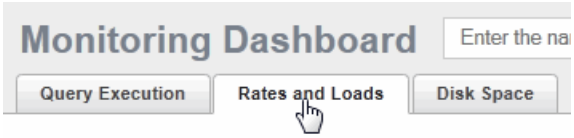
2.8 Monitoring Rates and Loads

In general, rates and loads measure how efficiently data is exchanged between applications and MarkLogic Server. Rates and loads are defined as follows:

- Rates — The amount of data (MB per second) currently being read from or written to MarkLogic Server.
- Loads — The execution time (in seconds) of current read and write requests, which includes the time requests spend in the wait queue when maximum throughput is achieved.

For details on how to interpret rates and loads, see “What is Happening on the MarkLogic Server Cluster Now?” on page 10.

To display monitoring data related to rates and loads, select the Rates and Loads tab in the top left-hand portion of the Monitoring Dashboard.



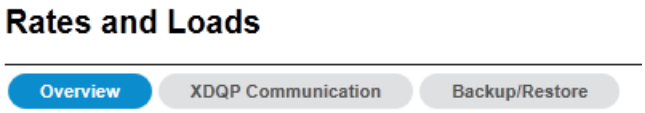
There are three types of rates and loads monitoring data. Select the type of rates and loads data by clicking on one of the three buttons displayed under Rates and Loads:

The monitoring data displayed by each of these buttons is described in the following sections:

- Overview
- XDQP Communication
- Backup/Restore

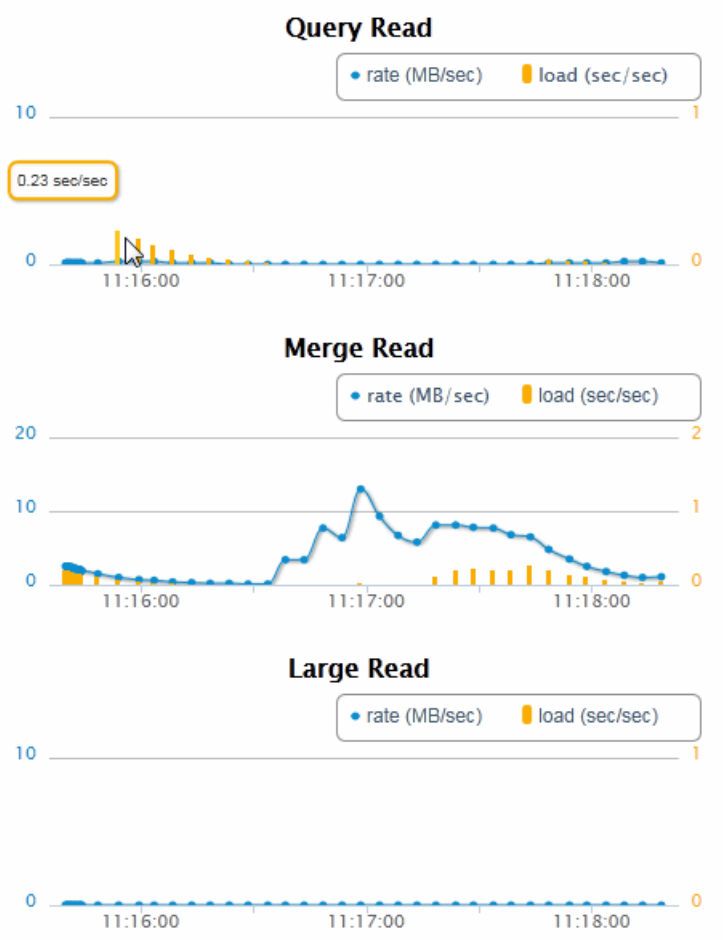
2.8.1 Overview

To obtain rates and loads data for queries, merges, and large data, click on the Overview button:

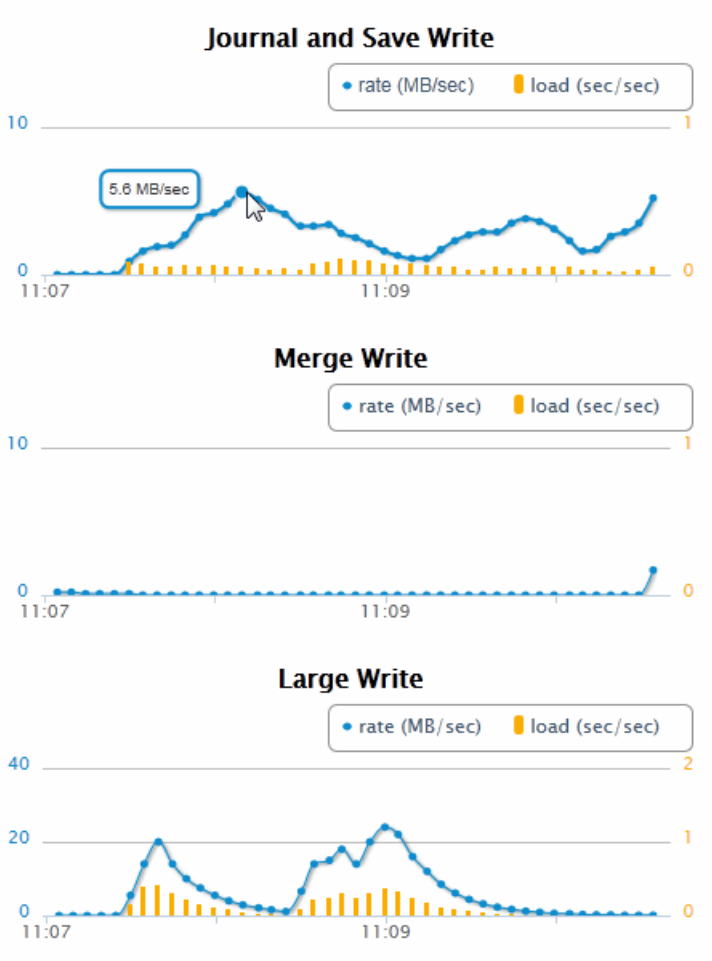


The left-hand side of the Rates and Loads Overview page displays the monitoring data related to query, merge, and large data reads.

Note: For details on Large Data, see Working With Binary Documents in the *Application Developer's Guide*.



The right-hand side of the Rates and Loads Overview page displays the monitoring data related to journal and save, merge, and large data writes.

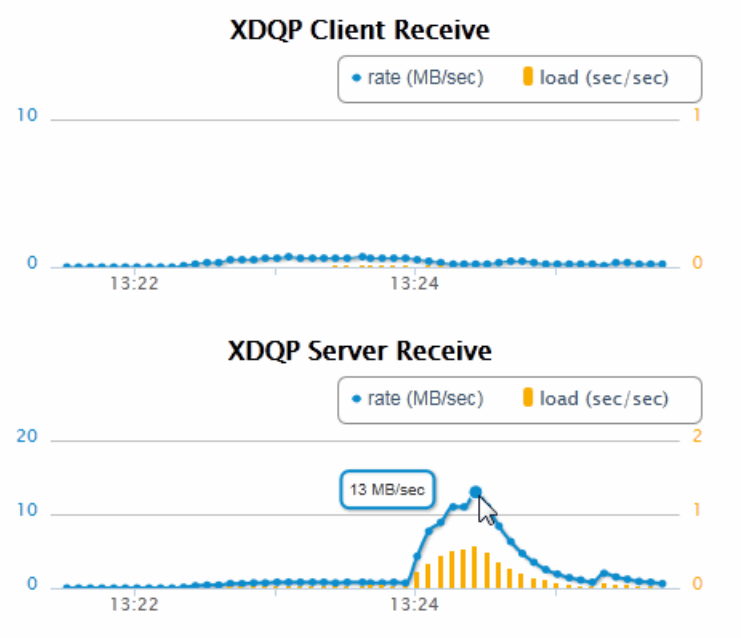


2.8.2 XDQP Communication

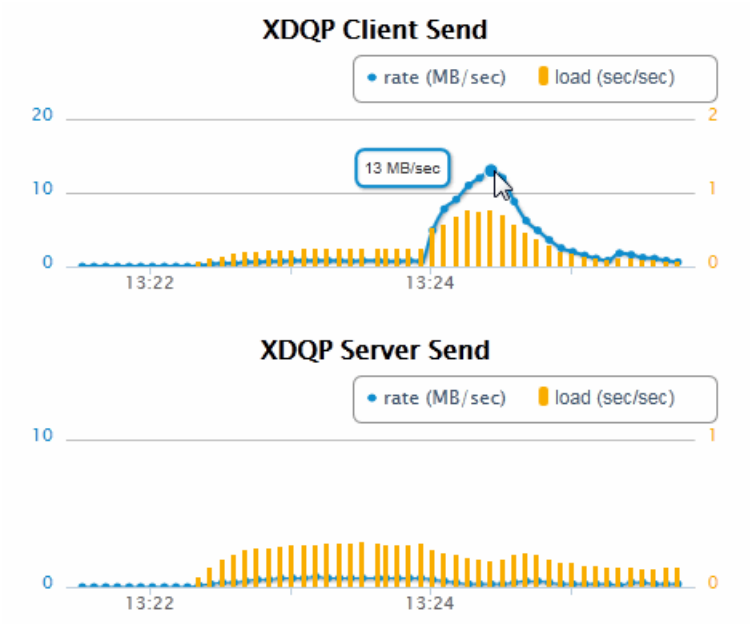
Communication between MarkLogic Server hosts within a cluster and between hosts in different clusters is done using the XDQP protocol. Both the rate and load are displayed for each sample interval. Unusually high XDQP loads may indicate a network connection problem.

To monitor the rates and loads related to XDQP communication, click on the XDQP Communication button:

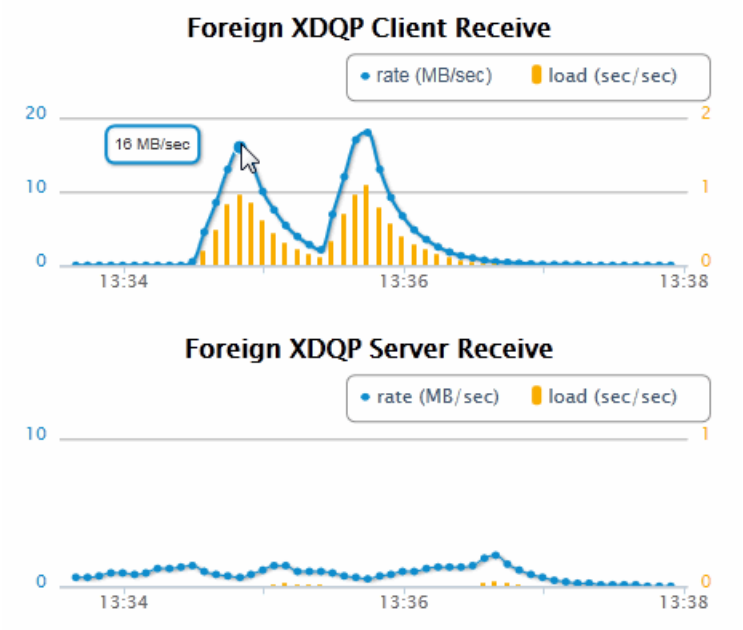
The upper left-hand side of the XDQP Communication page displays the monitoring data related to XDQP data received by the client and server.



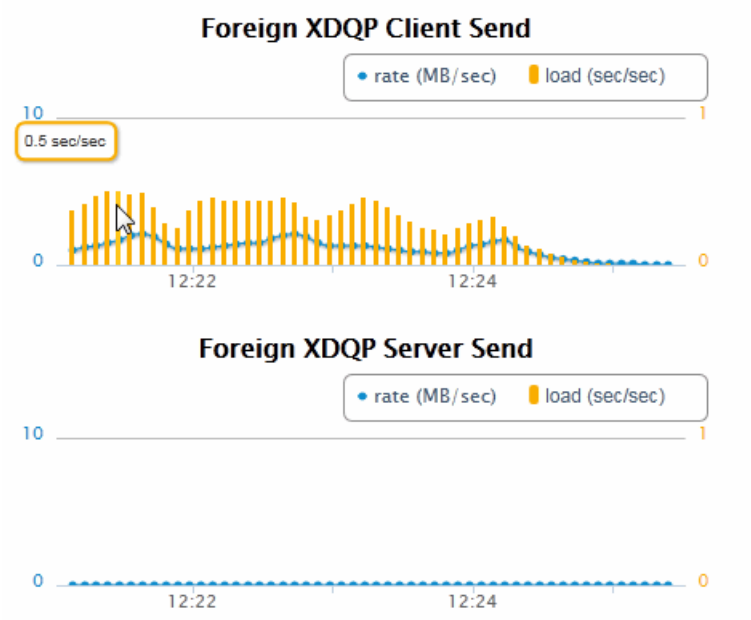
The upper right-hand side of the XDQP Communication page displays the monitoring data related to XDQP data sent by the client and server.



The lower left-hand side of the XDQP Communication page displays the monitoring data related to XDQP data received by the client and server from a foreign cluster.



The lower right-hand side of the XDQP Communication page displays the monitoring data related to XDQP data sent by the client and server to a foreign cluster.

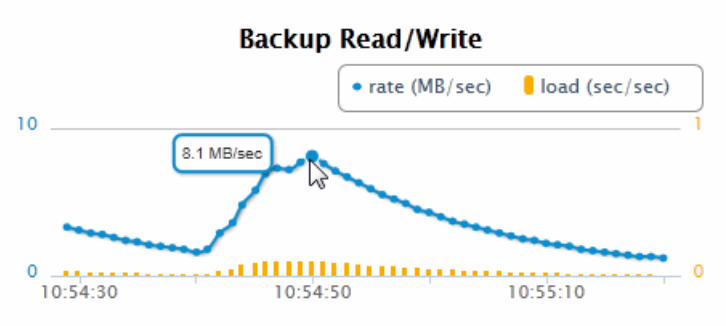


2.8.3 Backup/Restore

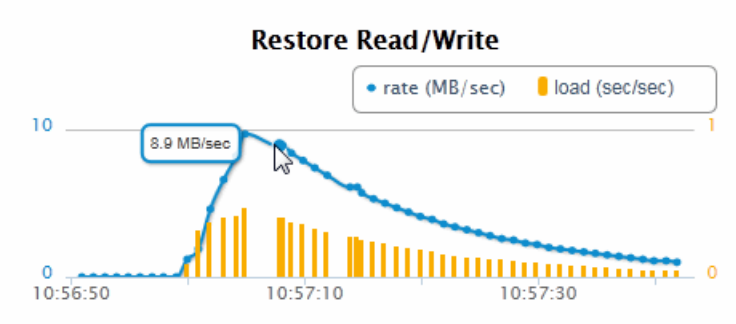
Backup and restore processes can impact the performance of applications and other background tasks in MarkLogic Server, such as merges and indexing.

To monitor the rates and loads related to backup and restore operations, click on the Backup/Restore button:

The left-hand side of the Backup/Restore page displays the monitoring data related to Backup reads and writes.



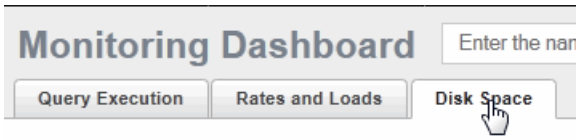
The right-hand side of the Backup/Restore page displays the monitoring data related to Restore reads and writes.



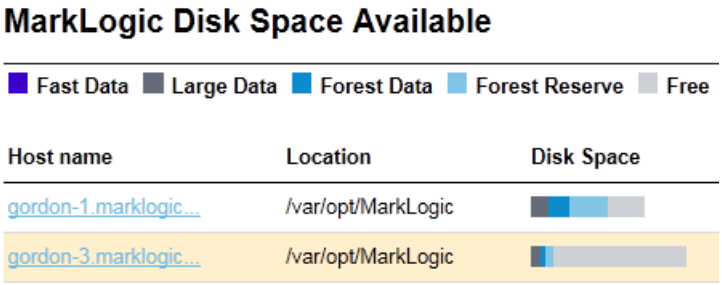
2.9 Monitoring Disk Space

Disk space usage is a key monitoring metric. In general, forest merges require twice as much disk space than that of the data stored in the forests. If a merge runs out of disk space, it will fail. In addition to the need for merge space on the disk, there must be sufficient disk space on the file system in which the log files reside to log any activity on the system. If there is no space left on the log file device, MarkLogic Server will abort. Also, if there is no disk space available to add messages to the log files, MarkLogic Server will fail to start.

To display monitoring data related to disk space, select the Disk Space tab in the top left-hand portion of the Monitoring Dashboard.



The data displayed on the Disk Space is for a specific host. You can select the host in the upper-left-hand section of the Disk Space page. The hosts in this list are sorted by those with the least available disk space at the top.

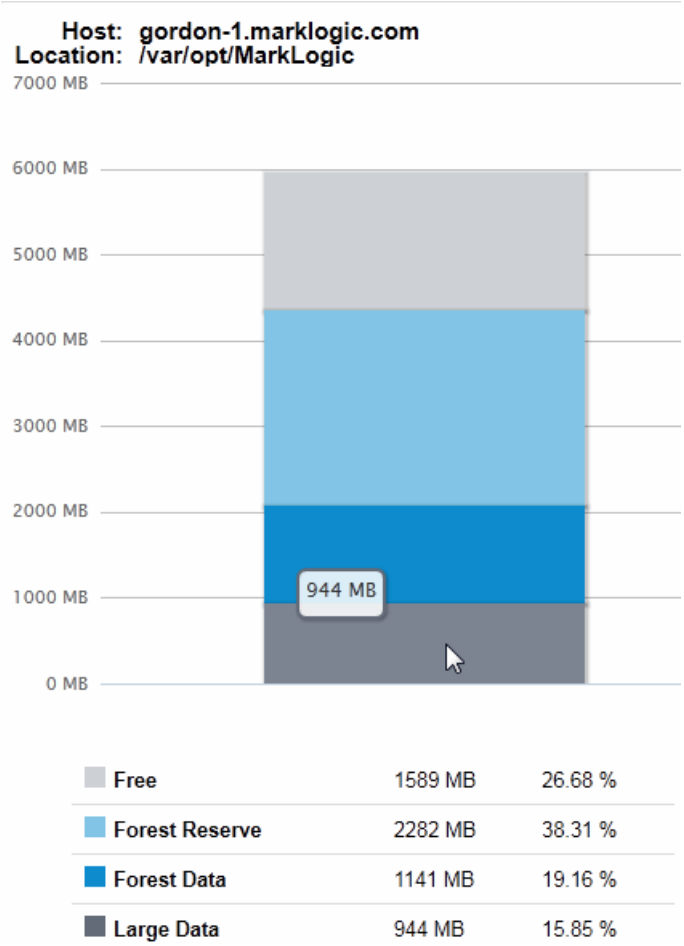


The disk space monitoring metrics are:

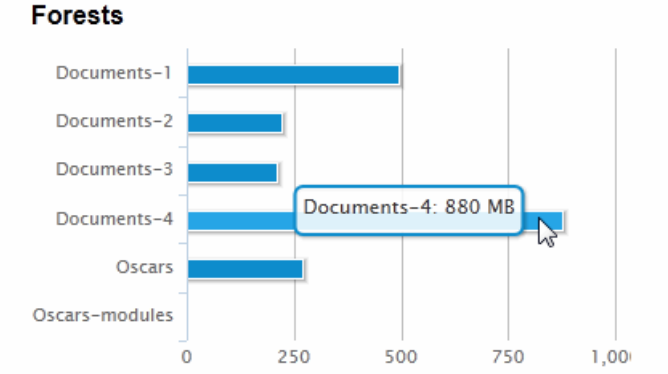
- Fast Data — The amount of disk space used by the forests' Fast Data Directory. The Fast Data Directory is typically mounted on a specialized storage device, such as a solid state disk. Fast data consists of transaction journals and as many stands that will fit on the fast storage device. For more information on Fast Data, see Fast Data Directory on Forests in the *Query Performance and Tuning Guide*.
- Large Data — The amount of disk space used by the forests' Large Data Directory. The Large Data Directory contains binary files that exceed the 'large size threshold' property set for the database. Large Data is not subjected to merges so, unlike Forest Data, Large Data does not require any additional Forest Reserve disk space. For more information on Large Data, see Working With Binary Documents in the *Application Developer's Guide*.
- Forest Data — The amount of disk space used by the data in the forest stands. This data is subject to periodic merges.

- Forest Reserve — The amount of free disk space that should be held in reserve to enable MarkLogic Server to merge the Forest Data.
- Free — The amount of free space on the disk that remains after accounting for the Forest Reserved space.

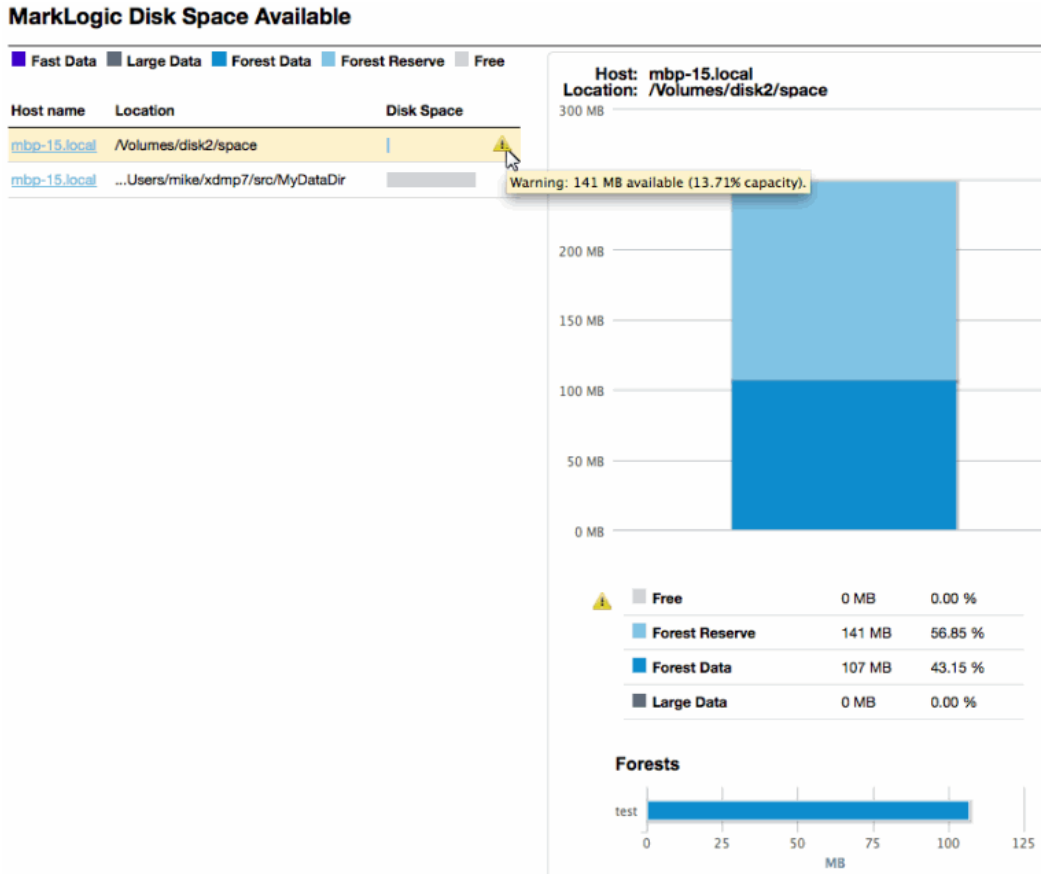
The upper right-hand section of the Disk Space page displays the amount of free space on the disk, along with how much reserve space is reserved for forest merges and the actual amount of space currently used by the forests and large data.



The lower right-hand section of the Disk Space page displays the amount of space on the disk used by the individual forests.

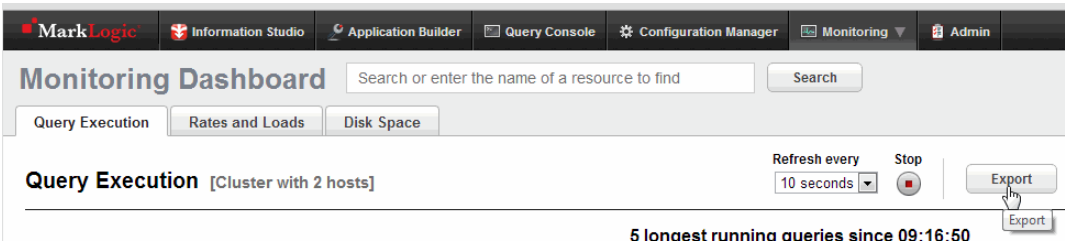


If your disk has less than 15% capacity a warning message is generated, as shown below. If the capacity falls to less than 10%, a critical message is generated.



2.10 Exporting Monitoring Data

Each of the three tabbed Monitoring Dashboard pages (Disk Space, Query Execution, Rates and Loads) has an Export button in its upper right corner, on the same line as the current tab's name. When clicked, it exports the page's data to a local XML file, formatted to be openable in Excel.



The exported files have tab-specific names incorporating a timestamp of when the file was exported. For example:

```
disk-space-20120210-160945.xml
```

indicates that it contains a page of data from the Disk Space tab, exported on February 10th, 2012 (2012 02 10) at 4:09:45 p.m. (16 09 45) (spaces added in this paragraph for clarity).

The exported data is from a JavaScript cache that automatically accumulates data as the page is drawn and refreshed. Two of the tabbed pages, Query Execution and Rates and Loads, accumulate data over time. A maximum1000 latest data points are cached for each of these pages, no matter how long the monitor page runs.

By default, data is cached every 10 seconds. This rate depends on the polling interval, which is set on the Dashboard page within the Refresh drop-down menu. See "Setting the Sample Interval" on page 18.

When using the Export button, remember these caveats:

- The cache is not in a persistent file, so manually refreshing the browser clears it of all accumulated data. Immediately after a manual browser refresh, there is no data to export.
- Clicking Export returns only the data from the current tab's page. For example, if you are on the Query Execution tab, clicking Export only writes out data from Query Execution and does not write out data from the Rates and Loads or Disk Space tabs. To get the values from all three tabs, you have to go to each tab and click its Export button, resulting in three separate files.
- However, when clicking Rates and Loads' Export button, the file does contain the data from all three of Rates and Loads' sub-tabs (Overview, XDQP Communication, and Backup/Restore).

Previously, you had to turn on caching this data with a `debug=true` parameter in the browser URL. Now, data is cached by default.

3.0 MarkLogic Server Monitoring History

This chapter describes how to use the Admin Interface and Monitoring History dashboard to capture and make use of historical performance data for a MarkLogic cluster. These same Monitoring History operations can also be done using the XQuery and REST APIs, as described in *XQuery and XSLT Reference Guide* and the *MarkLogic REST API Reference*.

> **Note:** All MB and GB metrics described in this chapter are base-2.

The main topics in the chapter are:

3.1 Overview

The Monitoring History feature allows you to capture and view critical performance data from your cluster. Once the performance data has been collected, you can view the data in the Monitoring History page. The top-level Monitoring History page provides an overview of the performance metrics for all of the key resources in your cluster. For each resource, you can drill down for more detail. You can also adjust the time span of the viewed data and apply filters to view the data for select resources to compare and spot exceptions.

By default, the performance data is stored in the Meters database. Monitoring History capture is enabled at the group level. Typically you have one group per cluster. You can also configure a consolidated Meters database that captures performance metrics from multiple groups. The group configuration defines which database is used to store performance metrics for that group (defaulting to a shared Meters database per cluster), as well as all configuration parameters for performance metrics, such as the frequency of data capture and how long to retain the performance data. The Meters database can participate in all normal database replication, security, and failover operations.

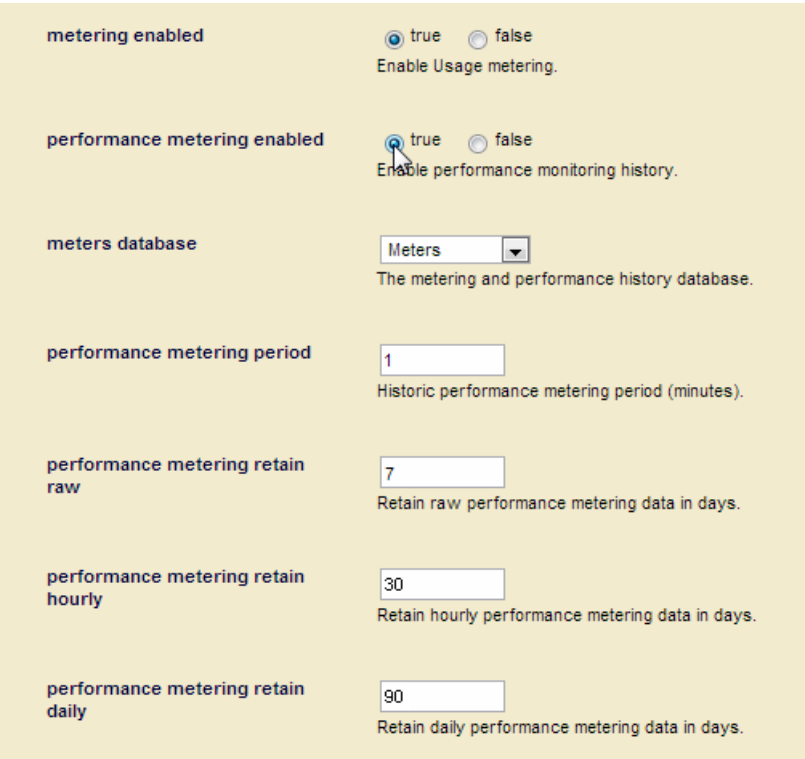
3.2 Enabling Monitoring History on a Group

In order to collect monitoring history data for your cluster, you must enable performance metering for your group.

1. Log into the Admin Interface.

2. Click the Groups icon on the left tree menu.

3. Locate the Performance Metering Enabled field toward the bottom of the Group Configuration page and click on `true`.



You can configure the parameters for collecting monitoring history, as described in the table below.

Parameter	Description
meters database	The database in which performance monitoring history and usage metrics documents are stored. By default, historical performance and usage metrics are stored in the Meters database.
performance metering period	The performance metering period, in minutes. Performance data is collected at each period. The period can be any value of 1 minute or more. **Note:** If you are collecting monitoring history for multiple groups, you should either set the same period for each group or configure your filter to view the history data for one group at a time.
performance metering retain raw	The number of days raw performance monitoring history data is retained. See “Setting the Monitoring History Data Retention Policy” on page 34 for details.
performance metering retain hourly	The number of days hourly performance monitoring history data is retained. See “Setting the Monitoring History Data Retention Policy” on page 34 for details.
performance metering retain daily	The number of days daily performance monitoring history data is retained. See “Setting the Monitoring History Data Retention Policy” on page 34 for details.

3.3 Setting the Monitoring History Data Retention Policy

The retention policy (for raw, hourly, daily) is a value set in days. If performance metering is enabled, then all data that is older than that many days for the specified period (raw, hour, day) is deleted. The retention policy is set at a group level, so different groups can have different retention policies. For example, GroupA may have raw set to 1 day and GroupB may have raw set to 10 days. The cleanup code follows this retention value on a per-group basis.

There are cases where metering data may become orphaned, so it may no longer belong to an existing group. Some examples of when this could occur are:

- Deleting a group
- Importing metering data from another cluster

Any metering data that no longer belongs to any active group in the current cluster is deleted. To avoid this, turn off metering or avoid deleting groups and instead move hosts out of the group but keep the group in the cluster configuration.

> **Note:** Loading older Monitoring History data (for example, by restoring a backup of the Meters database) will be immediately affected by data retention policy. So, you should turn off perfomance metering prior to restoring any data that is older than the time specified by your retention policy.

Deletion of data older then the retention policy occurs no sooner than the retention policy, but may, for various reasons, still be maintained for an unspecified amount of time.

> **Note:** Changing the retention policy from smaller to larger values does not restore data that has already been deleted.

The default data retention policy settings are as shown in the table below. To maximize efficiency, it is a best practice to retain raw data for the least number of days and the daily data for the most number of days.

Period	Retention Period
Raw	7 Days
Hourly	30 Days
Daily	90 Days

3.4 Viewing Monitoring History

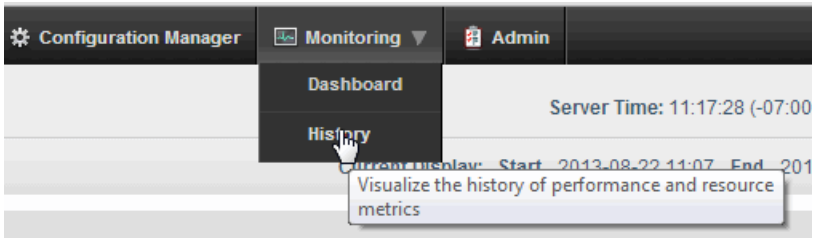
You can display the Monitoring History by doing the following:

1. Open a browser and enter the URL:

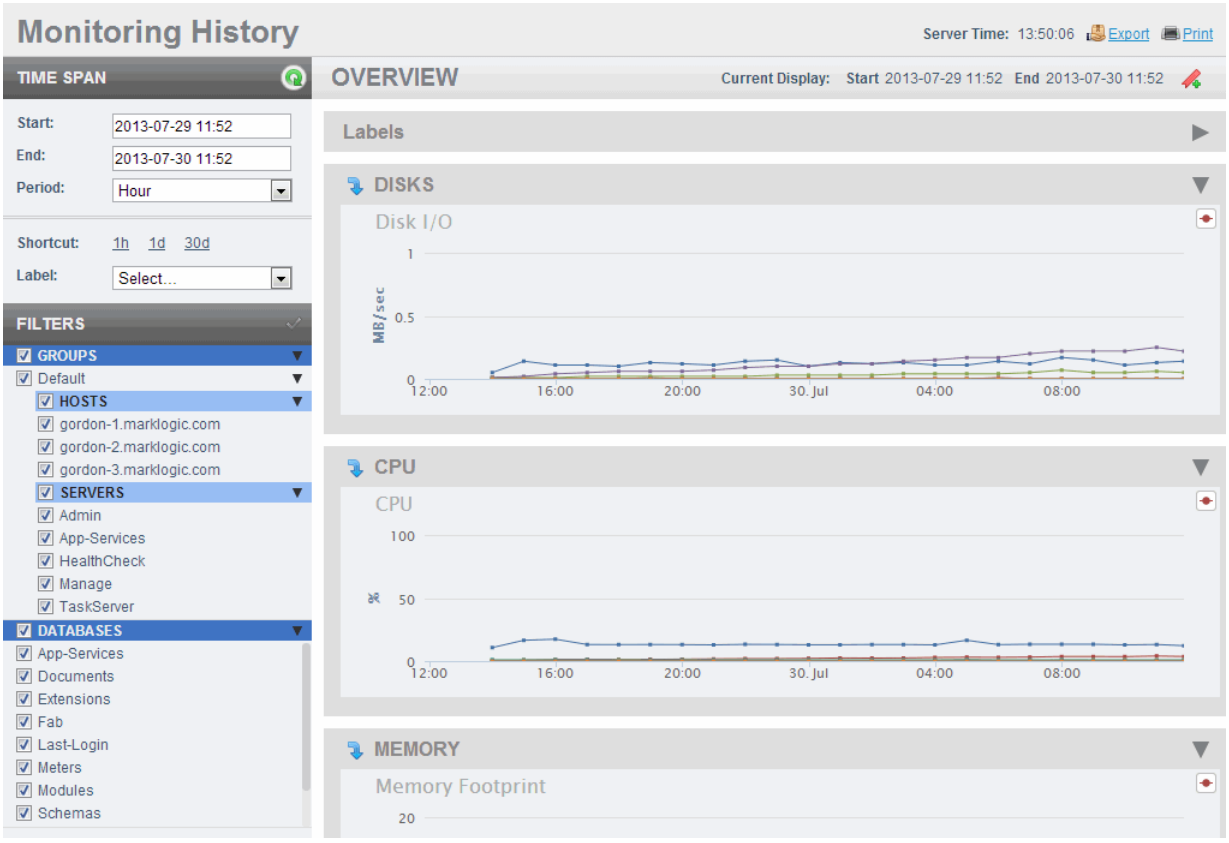
   ```
   http://monitor-host:8002/
   ```

   where *monitor-host* is a host in the cluster you want to monitor

2. At the top of the page, click on Monitoring and click on History in the pull-down menu:

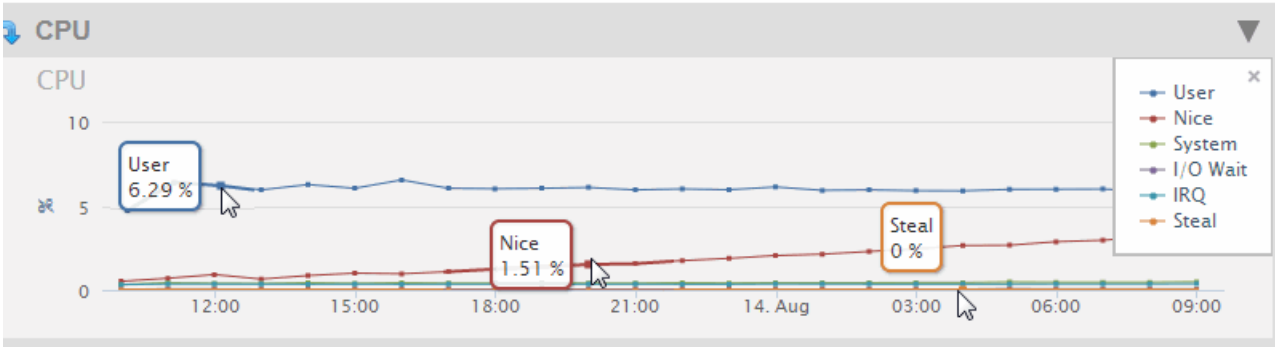


3. The Monitoring History page appears. From the Monitoring History Overview page, you can navigate to any of the pages described in this chapter.

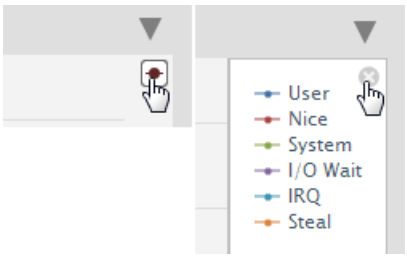


Each line in a chart represents a metric for the resource. In the Overview page, the lines represent an aggregate of the metrics for all of the cluster resources. In each Details page, the lines represent the metric for each specific resource.

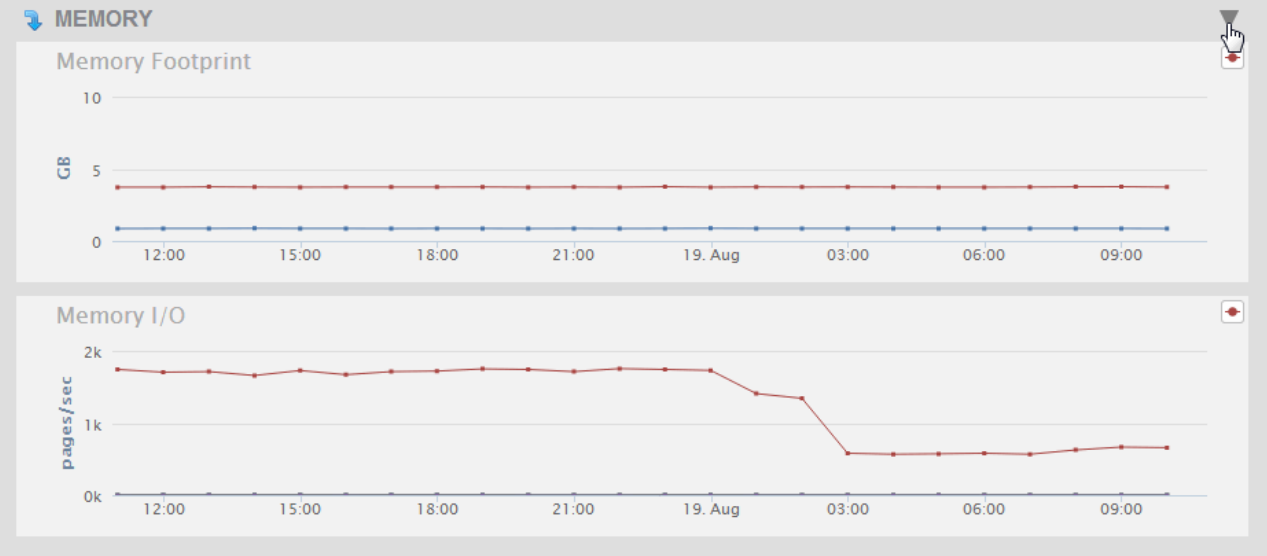
Each point on a line represents a period in which the performance data was captured. Hovering over a chart point displays the name of the resource metric, along with the performance value for the metric at that point in time.



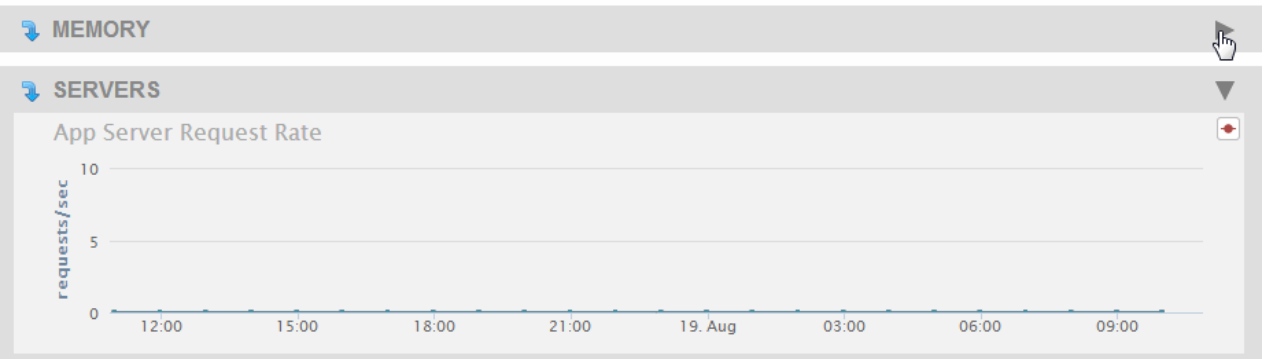
The displayed metrics (in MegaBytes per second) are color coded. You can display a legend that indicates which colors represent which metrics by clicking on the red dot in the upper right-hand section of the graph. To close the legend, click on the ‘x’ in the upper right-hand portion of the legend window.



To simplify the view of charts on a page, you can collapse a chart or a group of charts for a resource by clicking on the triangle in the upper right-hard portion of the chart or chart group.



To expand a collapsed chart view, click on the triangle in the upper right-hard portion of the collapsed chart.

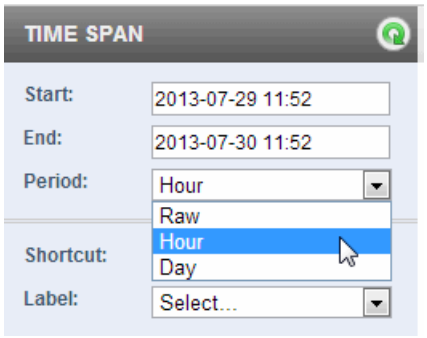


3.5 Viewing Monitoring History by Time Span and Frequency

As described in “Enabling Monitoring History on a Group” on page 32, the frequency in which performance metrics are captured is configurable, in minute intervals. The snapshots of performance metrics for each host are rolled up into a summary document that contains aggregate calculations on the values for that host.

You can configure your view of the captured performance data by time span and frequency.

The Time Span settings are located in the upper left-hand corner of the Monitoring History page.

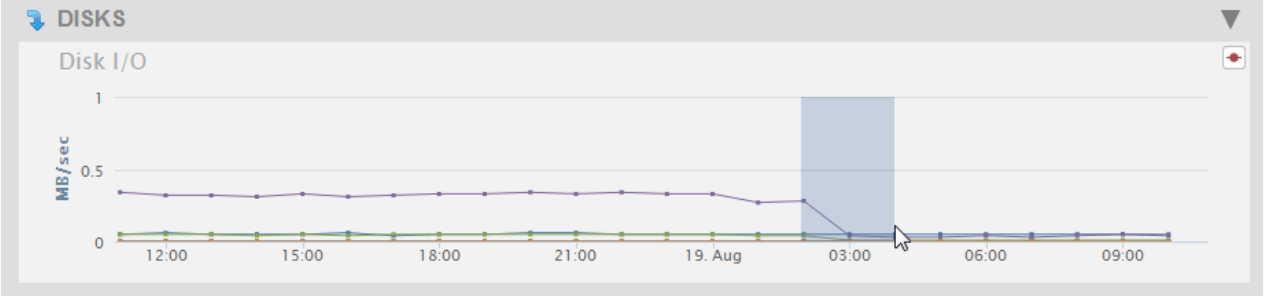


There are three basic settings you can adjust to control how the data is displayed:

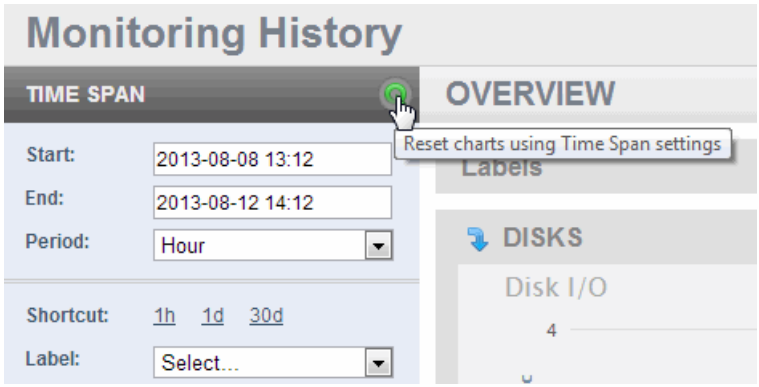
- A date/time range, down to the granularity of a minute, that determines the time span of the displayed data. (By default, this is the last 24 hours.)
- A period interval that determines the frequency of the displayed data. The possible intervals are shown in the following table.

Period	Description
Raw	Display the performance data just as it was captured with the set frequency.
Hour	Display the performance data, in aggregate form, per hour. (This is the default.)
Day	Display the performance data, in aggregate form, per day.

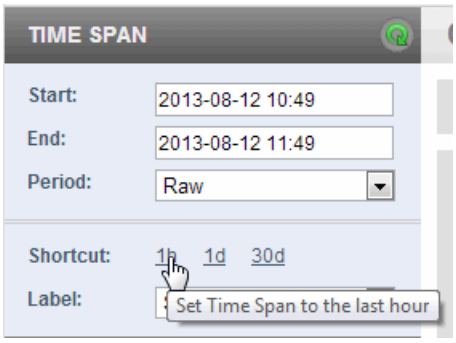
You can “zoom in” to display part of the timespan by selecting the begin time of your “zoom” on any chart and click and hold your left mouse button and drag it to the end “zoom” time. The selected timeframe is highlighted and the zoomed-in time is displayed for all of the charts in the page. Navigating to another Monitoring History page resets all of the charts to the timespan selected in the TIME SPAN panel.



After changing either the time span and/or the period, click on refresh to display the updated charts. Clicking refresh will also update any changes you've made to the Filters settings. For details about filters, see “Filtering Monitoring History by Resources” on page 43. If you have zoomed into a portion of a timespan, refresh will redisplay the charts using the timespan selected in the TIME SPAN panel.



You can use the Shortcut links to display either the last hour, day or 30 days of performance data. Selecting a Shortcut link will automatically refresh the displayed charts.



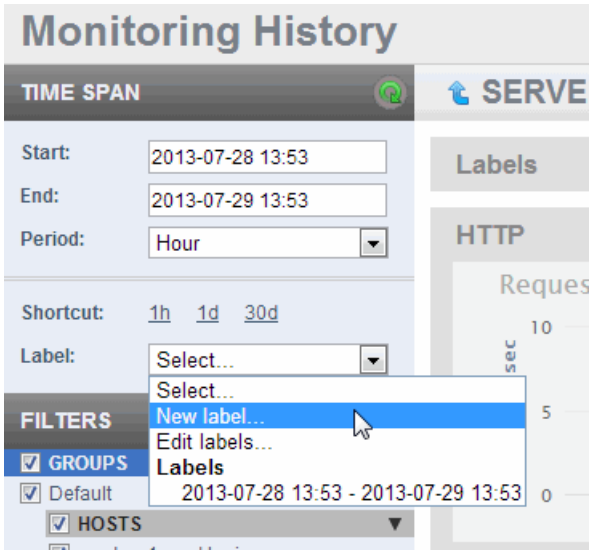
Each Shortcut also sets the Period value, as shown in the table below.

Shortcut	Period
1h	Raw
1d	Hour
30d	Day

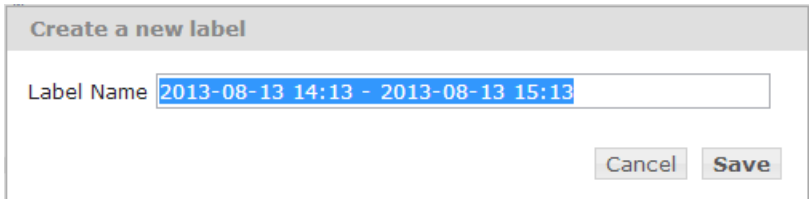
3.6 Labeling Monitoring History Time Spans

You can use the Label feature to capture and tag metrics for the set time span. You can store any number of labels. These labels can be used to identify events, instances, and periods of time. Labels can be added, updated or deleted at any time. Labels themselves are not stored with the raw metric data. They are only used for reporting purposes.

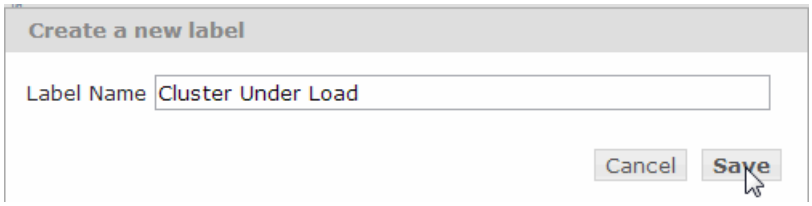
1. To create a label for your current view of the Monitoring History data, select New Label from the Label pull-down menu.



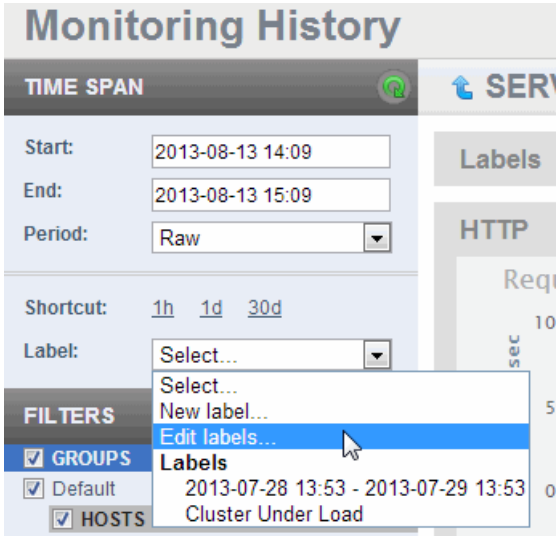
2. In the Create a New Label popup window, the name of the label is the time span of the currently displayed charts, by default.



3. You can keep the default name for the label, or change it to be more descriptive. Click Save.



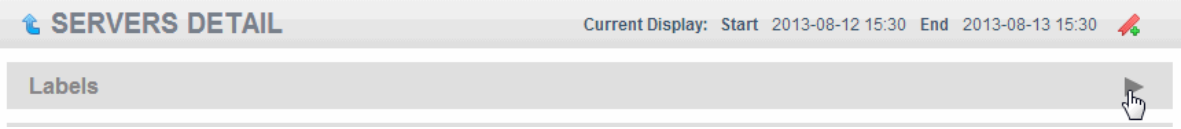
4. You can edit your label names or delete labels by selecting Edit Labels from the Labels pull-down menu.



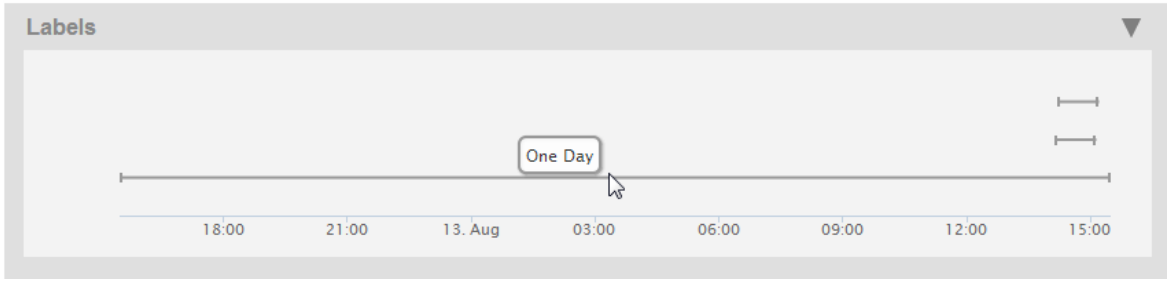
5. In the Edit Labels popup window, you can either edit the label name or delete the label. To delete a label, hover over the label and a click on the garbage can icon to the right. When finished editing, click Close.

Note: If you edit a label and, before closing the Edit Labels window, decide not to save your edits, press the Esc key to terminate the edits and keep the original labels.

6. You can view all of the labels that have data within the currently selected timespan by clicking on the triangle to the right of the Labels section at the top of the Monitoring History page to expand the Labels chart.

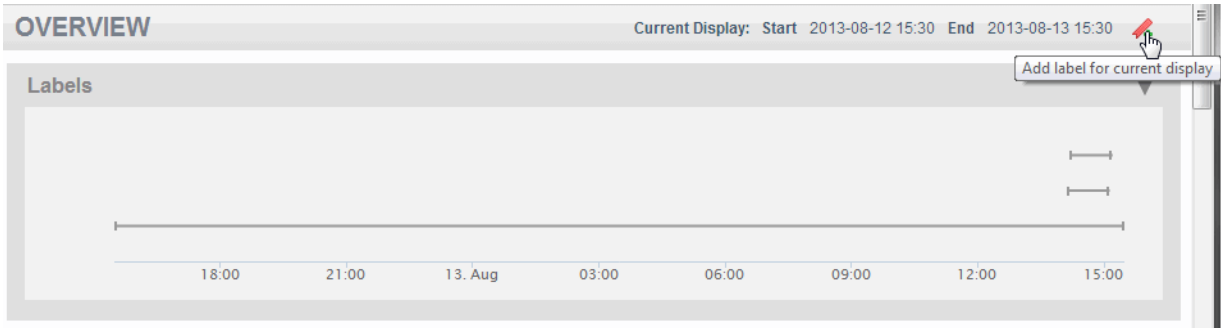


7. Each label appears as a timeline. Hover over a timeline to display the label name. Click on a timeline to update the view to the time span associated with the label. Selecting a timeline is functionally equivalent to selecting a label from the Label menu in that it updates the view with the start and end times in the TIME SPAN panel.

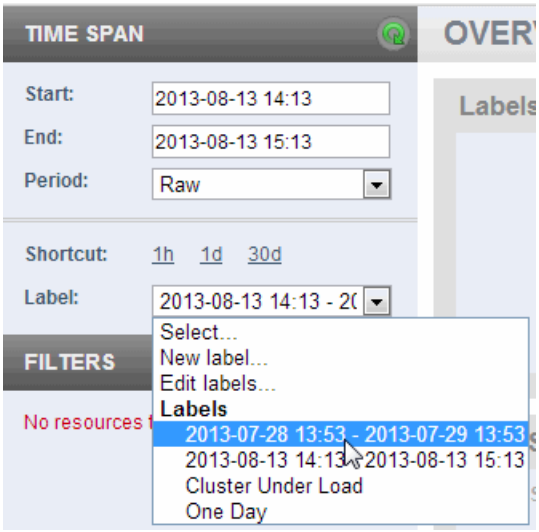


Note: If your labeled data has been purged from the Meters database, as the result of the retention policy or some other reason, the label will remain but there will be no data associated with that label.

8. You can click on the label icon at the top right-hand portion of the page to create a label for the currently displayed time span. Follow the same procedure as described in steps 2 and 3 to finish creating the label.



If the data for a label does not fall within the currently displayed timespan, the label will not be displayed in the Labels chart. To display the charts for such labels, select the label from the Label pull-down menu.

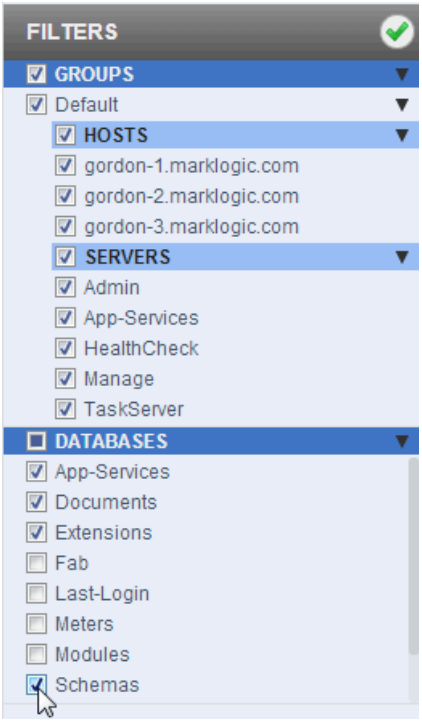


3.7 Filtering Monitoring History by Resources

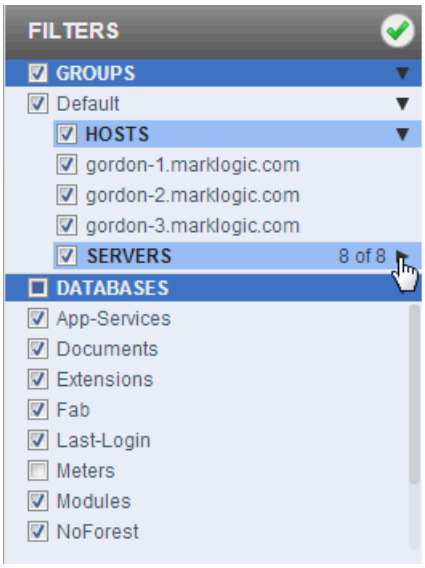
You can set filters for select resources to display only the stored performance metrics for those resources. You can filter by groups and databases. And in each group, by hosts and servers. By default, the metrics for all of the resources in the cluster are displayed.

Filter types that are active for the current view have headings highlighted in blue. For example, on the Overview page, all filters are active while on the Databases Detail view, only database resources are active.

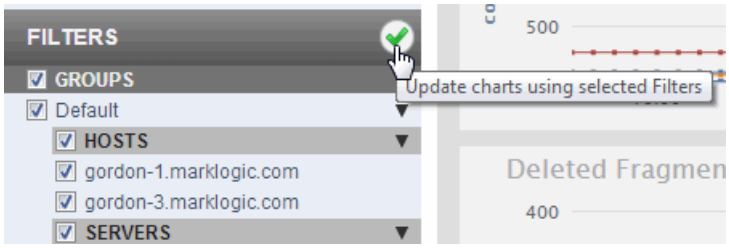
In the filters panel, you can check or uncheck a resource to display or not display the performance metrics for that resource.



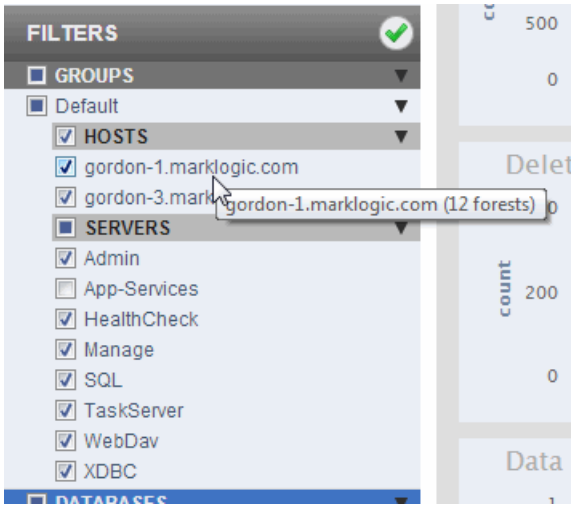
In order to focus on the resources of interest, you can collapse a category by clicking on the triangle in the right-hand section of the panel. The number of resources for the collapsed category are displayed.



Clicking the checkmark updates the charts with the current filter settings. It does not apply any changes that may have been made to the above TIME SPAN settings.



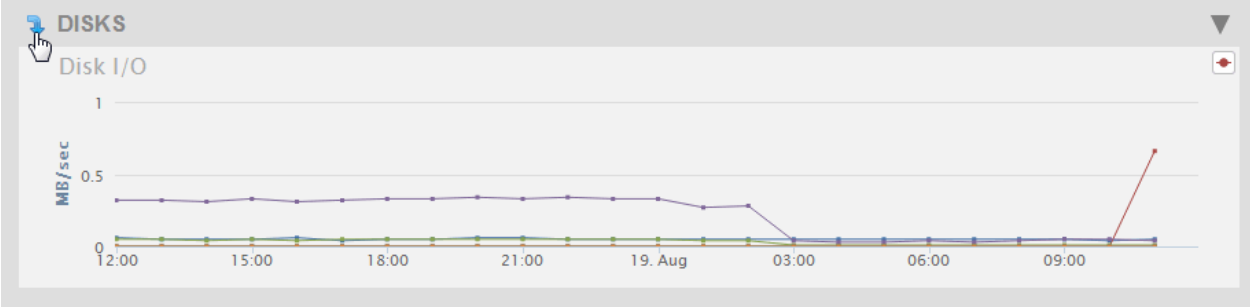
You can mouse over the resource names in the filter list to get extra information about the resources. For example, mousing over a host name shows the number of forests associated with the host and mousing over a server name shows the server type.



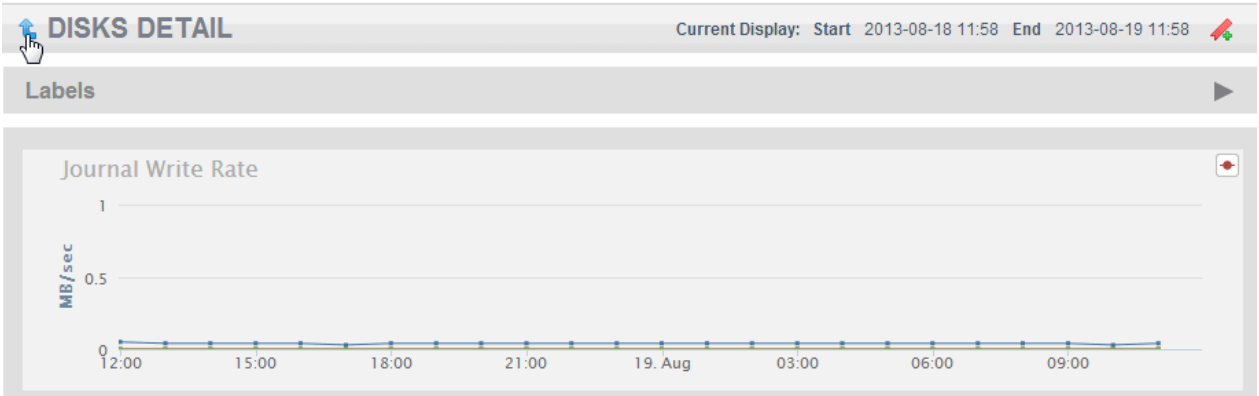
3.8 Historical Performance Charts by Resource

From the Monitoring History dashboard, you can view Overview and Detailed performance metrics in graph form for each resource in the cluster. In the Overview page, the lines on a graph represent an aggregate of the metrics for all of the cluster resources of that type. In each Details page, the lines represent the metric for each specific resource in the cluster.

To view the Detail page for a resource, click on the down arrow at the upper left-hand section of the resource graph on the Overview page.



To return to the Overview page from a Detail page, click on the up arrow at the upper left-hand section of the resource graph on the Detail page.

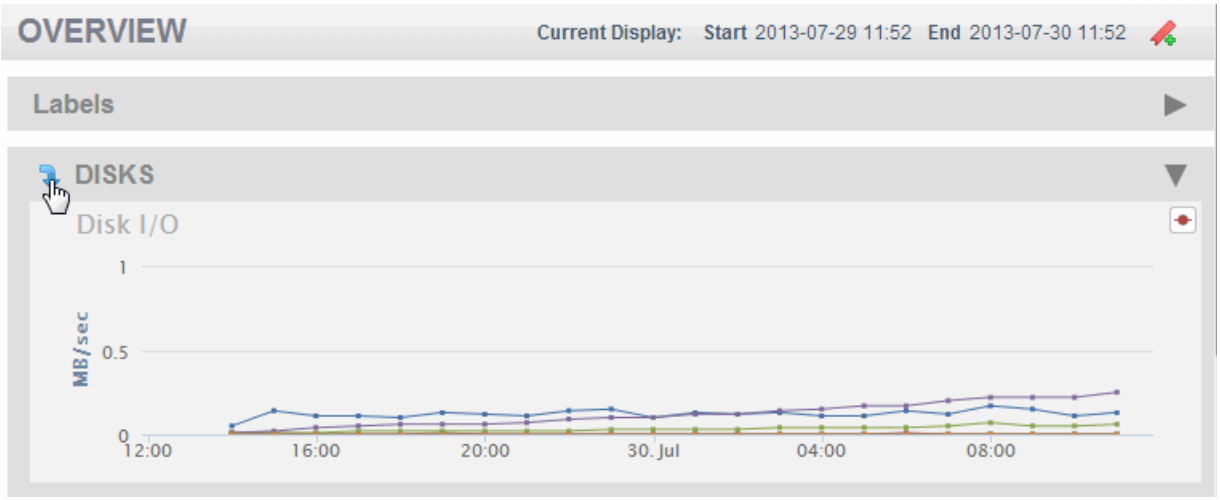


This section describes the Overview and Detail pages for the following resources:

- Disk Performance Data
- CPU Performance Data
- Memory Performance Data
- Server Performance Data
- Network Performance Data
- Database Performance Data

3.8.1 Disk Performance Data

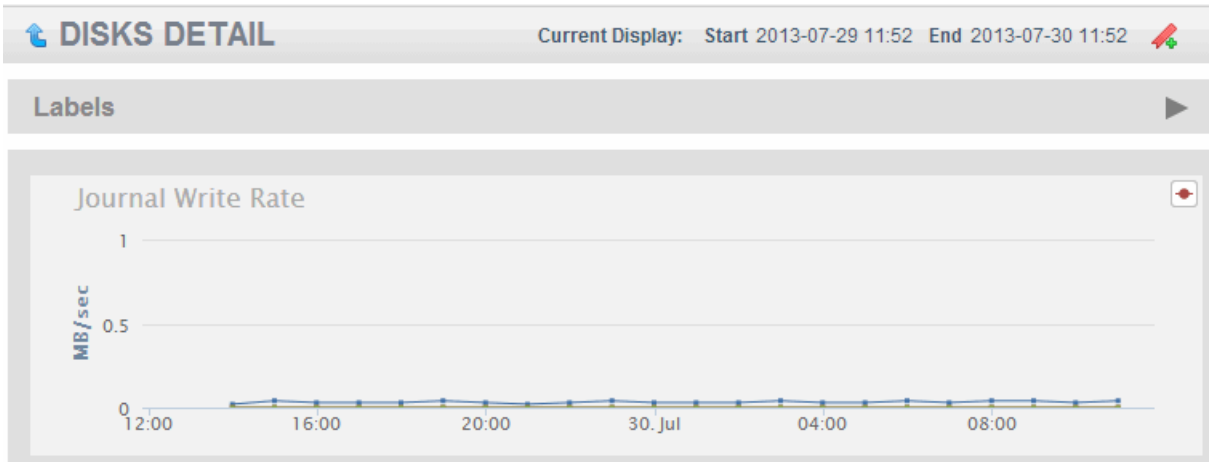
The Overview page displays a graph of the aggregate I/O performance data for the disks used by the hosts selected in the filter.



As described in “Viewing Monitoring History” on page 35, you can hover on a period point to view what disk operation was taking place at that point in time. Each performance metric is described in the table below.

Metric	Description
Writes	The disk I/O performance during journal and save write operations. This is the sum of journal-write-rate, save-write-rate, and large-write-rate.
Query Traffic	The disk I/O performance during a query or queries. This is is the sum of query-read-rate and large-read-rate.
Merge Reads	The disk I/O performance during a merge read operation.
Merge Writes	The disk I/O performance during a merge write operation.
Backup	The disk I/O performance during a backup operation. This is the sum of backup-write-rate and backup-read-rate.
Restore	The disk I/O performance during a restore operation. This is the sum of restore-read-rate and restore-write-rate.

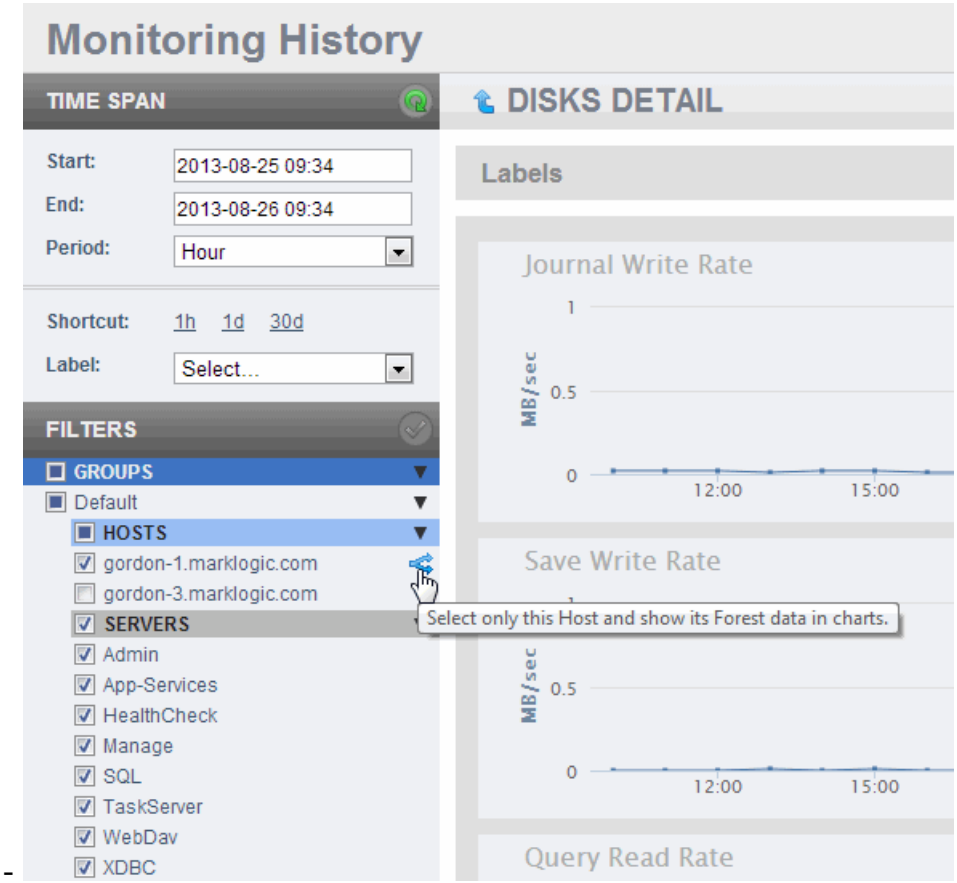
Click on the  arrow in the upper left-hand section of the DISKS graph in the Overview page to view charts that present more detailed disk performance metrics.



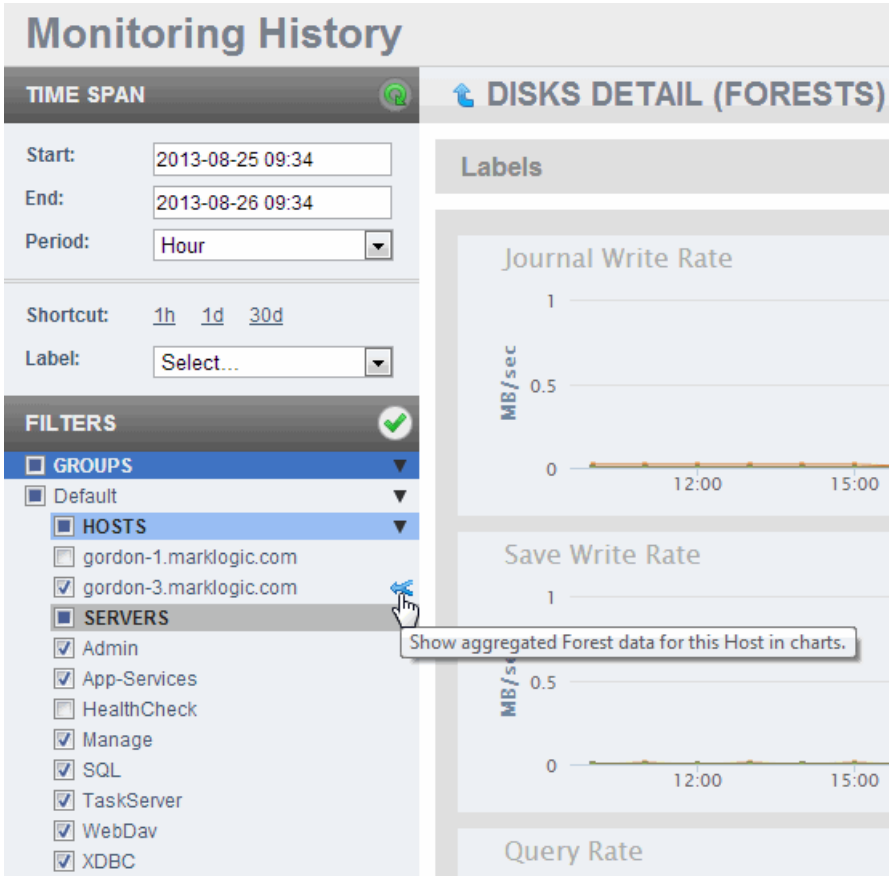
The metrics displayed by the charts on the DISKS DETAIL page are described in the table below.

Chart	Definition of Displayed Metric
Journal Write Rate	The moving average of data writes to the journal.
Save Write Rate	The moving average of data writes to in-memory stands.
Query Read Rate	The moving average of reading query data from disk
Merge Read Rate	The moving average of reading merge data from disk
Merge Write Rate	The moving average of writing data for merges
Backup Rate	The moving average of reading and writing backup data to disk. This is the sum of backup-write-rate and backup-read-rate.
Restore Rate	The moving average of reading and writing restore data from disk. This is the sum of restore-read-rate and restore-write-rate.
Large Binary Read Rate	The moving average of reading large documents from disk.
Large Binary Write Rate	The moving average of writing data for large documents to disk.

By default, Host data is viewed in aggregated form and must be viewed that way if multiple hosts are selected. When in the DISK DETAIL page, you can rollover any Host filter to reveal the Select and Expand button. This will deselect all of the other Hosts across all Groups, and apply all pending filter changes. The expanded charts display the data for each forest in that host as separate line in each chart.

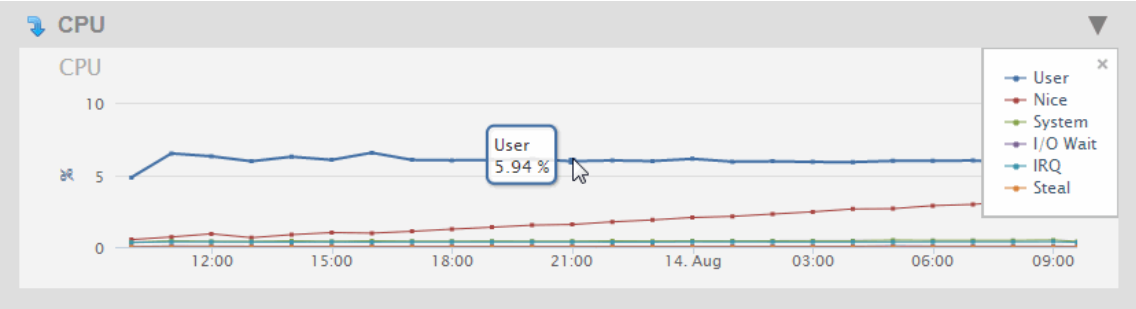


To return to the aggregate view, click on Aggregate button on an expanded Host. Doing so will also apply all pending filter changes to the displayed charts.



3.8.2 CPU Performance Data

The Overview page displays a graph of the aggregate I/O performance data for the CPUs used by the hosts selected in the filter.



As described in “Viewing Monitoring History” on page 35, you can hover on a period point to view what CPU operation was taking place at that point in time. Each performance metric in the CPU Overview chart is described in the table below.

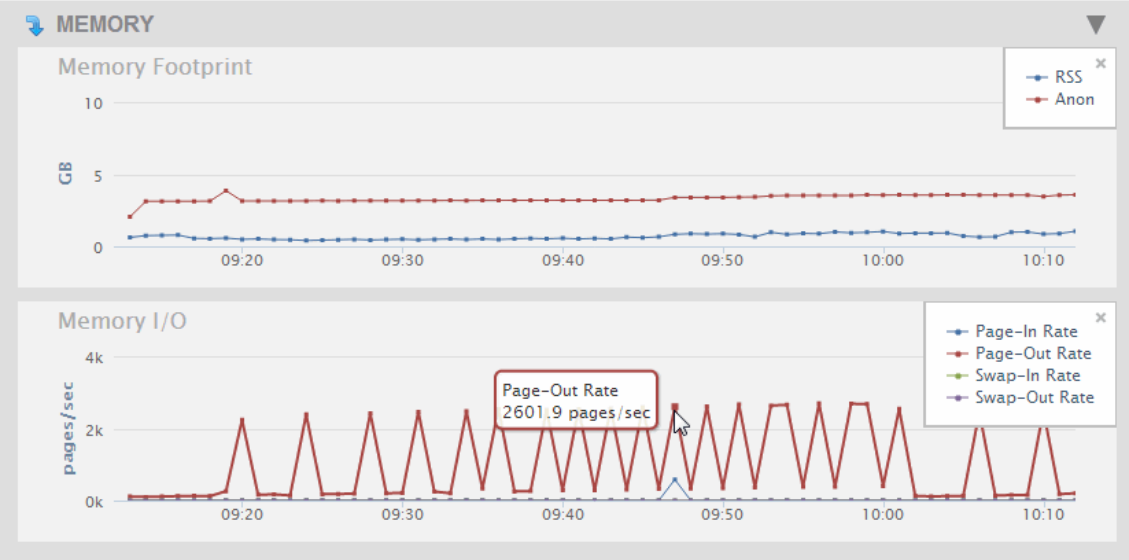
Metric	Description
User	Total percentage of CPU used running user processes that are not niced.
Nice	Total percentage of CPU used running user processes that are niced.
System	Total percentage of CPU used running the operating system kernel and its processes.
I/O Wait	Total percentage of CPU time spent waiting for I/O operations to complete.
IRQ	Total percentage of CPU utilization for servicing soft interrupts.
Steal	Total percentage of CPU ‘stolen’ from this virtual machine by the hypervisor for other tasks (such as running another virtual machine).

Click on the  arrow in the upper left-hand section of the CPU graph in the Overview page to view graphs that present more detailed CPU performance metrics. The charts on the CPU DETAIL page are described in the table below.

Chart	Description
I/O Wait	The percentage of CPU used waiting for I/O operations to complete for each host.
User	The percentage of CPU used running user processes that are not niced for each host.
System	The percentage of CPU used running the operating system kernel and its processes for each host.
Nice	The percentage of CPU used running user processes that are niced for each host.
Steal	The percentage of CPU ‘stolen’ from this virtual machine by the hypervisor for other tasks (such as running another virtual machine) for each host.
Idle	The percentage of CPU that is not doing any work for each host.
IRQ	The percentage of CPU servicing soft interrupts for each host.

3.8.3 Memory Performance Data

The Overview page displays a graph of the aggregate performance data for the Memory used by the hosts selected in the filter.



As described in “Viewing Monitoring History” on page 35, you can hover on a period point to view what CPU operation was taking place at that point in time. Each chart and associated performance metrics are described in the table below.

Chart	Description
Memory Footprint	The total amount (in GB) of memory consumed by all of the hosts in the cluster. The displayed metrics are: • RSS: The total amount of GB of Process Resident Size (RSS) consumed by the cluster. • Anon: The total amount of GB of Process Anonymous Memory consumed by the cluster.

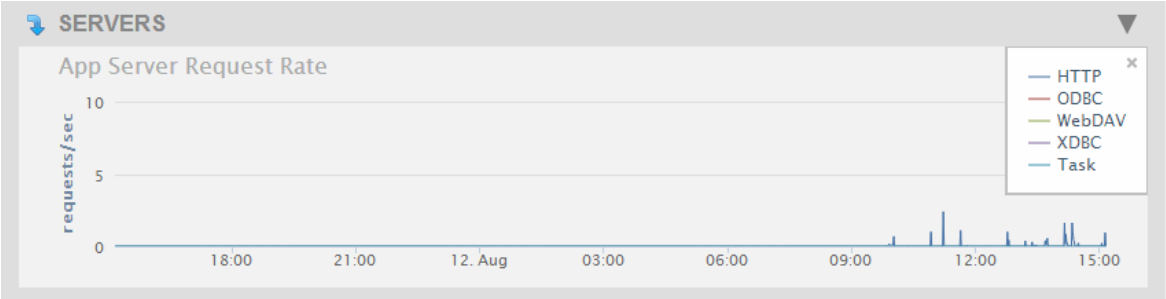
Chart	Description
Memory I/O	The number of pages per second moved between memory and disk. The displayed metrics are: • Page-In Rate: The page-in rate (from Linux /proc/vmstat) for the cluster in pages/sec. • Page-Out Rate: The page-out rate (from Linux /proc/vmstat) for the cluster in pages/sec. • Swap-In Rate: The swap-in rate (from Linux /proc/vmstat) for the cluster in pages/sec. • Swap-Out Rate: The swap-out rate (from Linux /proc/vmstat) for the cluster in pages/sec.

Click on the  arrow in the upper left-hand section of the MEMORY graph in the Overview page to view graphs that present more detailed MEMORY performance metrics. The charts on the MEMORY DETAIL page are described in the table below. The displayed metrics are drawn from `/proc/vmstat`.

Chart	Description
RSS	The amount of GB of Process Resident Size (RSS) for each host in the cluster.
Anon	The amount of GB of Process Anonymous Memory for each host in the cluster.
Page-In Rate	The page-in rate (in pages/sec) for each host in the cluster.
Page-Out Rate	The page-out rate (in pages/sec) for each host in the cluster.
Swap-In Rate	The swap-in rate (in pages/sec) for each host in the cluster.
Swap-Out Rate	The swap-out rate (in pages/sec) for each host in the cluster.

3.8.4 Server Performance Data

The Overview page displays graphs of the aggregate performance data for the App Servers selected in the filter.



The Overview page displays the charts described in the table below.

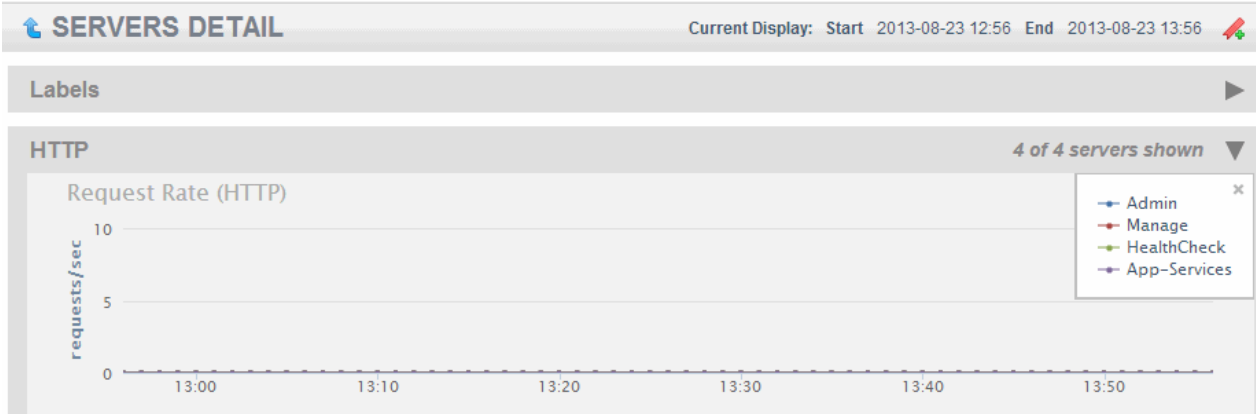
Chart	Description
App Server Request Rate	The total number of queries being processed per second, across all of the App Servers.
App Server Latency	The average time (in seconds) it takes to process queries, across all of the App Servers.
Task Server Queue Size	The number of tasks in the Task Server queue.
Expanded Tree Cache Hits/Misses	The number of times per second that queries could use (Hits) and could not use (Misses) the expanded tree cache.

With the exception of the Task Server Queue Size chart, which only displays the queue size for the one task server, the color-coded metrics for the server charts are as shown in the table below.

Metric	Description
HTTP	The metrics for the HTTP servers.
ODBC	The metrics for the ODBC servers.
WebDAV	The metrics for the WebDAV servers.
XDBC	The metrics for the XDBC servers.
Task	The metrics for the Task server.

Click on the  arrow in the upper left-hand section of the SERVERS graph in the Overview page to view graphs that present more detailed performance metrics for each App Server. The charts displayed on the SERVERS DETAIL page are described in the table below.

Note: If there are multiple groups defined, server names have the group that they are associated with in square brackets in the legend and rollovers.



The number of servers displayed out of the number of servers of each type in the cluster (for example, HTTP) is shown in the upper right-hand section of each server type group.

The following detailed charts are displayed for each type of App Server:

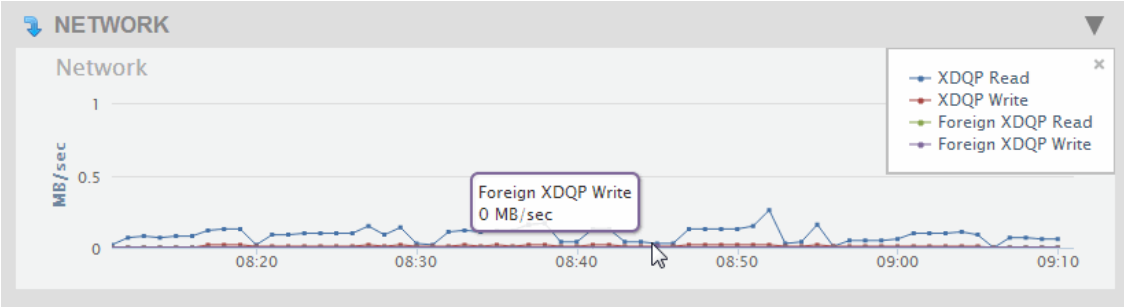
Chart	Description
Request Rate	The number of queries being processed per second by each App Server.
Latency	The average time it takes each App Server to process queries.
Expanded Tree Cache Rate Hits	The number of times queries could use the expanded tree cache on each App Server.
Expanded Tree Cache Rate Misses	The number of times queries could not use the expanded tree cache on each App Server.
Queue Size (Task Server Only)	The number of tasks in the Task Server queue on each host.

3.8.5 Network Performance Data

The network performance data graphs display performance in terms of XDQP reads and writes. XDQP is the protocol MarkLogic uses for internal host-to-host communication on port 7999.

The Overview page displays various XDQP performance as the sum of XDQP activity across the cluster. High XDQP rates are usually not an issue unless they are so high as to saturate your internal network. Higher usage occurs during data load and query execution. Merges do not involve XDQP.

Note: If XDQP is excessively high during loads, running the MarkLogic Content Pump (`mlcp`) with fast forest placement will minimize XDQP communication needs. For details on the MarkLogic Content Pump, see Loading Content Using MarkLogic Content Pump in the *Loading Content Into MarkLogic Server Guide*.



The Overview page displays a chart with the metrics described in the table below.

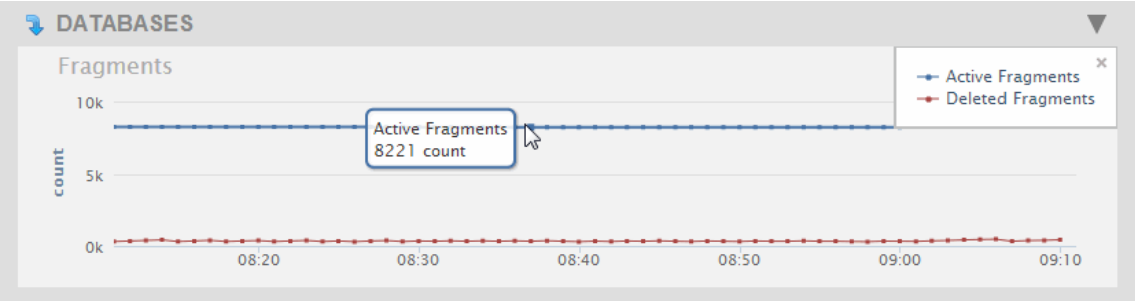
Metric	Description
XDQP Read	The total volume of all XDQP reads between hosts in the cluster. This is the sum of xdqp-client-receive-rate and xdqp-server-receive-rate.
XDQP Write	The total volume of all XDQP writes between hosts in the cluster. This is the sum of xdqp-client-send-rate and xdqp-server-send-rate.
Foreign XDQP Read	The total volume of all XDQP reads by the hosts in the cluster from a foreign cluster. This is the sum of foreign-xdqp-client-receive-rate and foreign-xdqp-server-receive-rate.
Foreign XDQP Write	The total volume of all XDQP writes by the hosts in the cluster to a foreign cluster. This is the sum of foreign-xdqp-client-send-rate and foreign-xdqp-server-send-rate.

Click on the  arrow in the upper left-hand section of the NETWORK graph in the Overview page to view graphs that present more detailed performance metrics for each host in the cluster. The charts displayed on the NETWORK DETAIL page are described in the table below.

Chart	Description
XDQP Read Rate	The amount of data (in MB/sec) read over XDQP by each host in the cluster. This is the sum of foreign-xdqp-client-receive-rate and foreign-xdqp-server-receive-rate.
XDQP Write Rate	The amount of data (in MB/sec) written over XDQP by each host in the cluster. This is the sum of foreign-xdqp-client-send-rate and foreign-xdqp-server-send-rate.
XDQP Read Load	The execution time (in seconds) of read requests by each host in the cluster. This is the sum of xdqp-client-receive-load and xdqp-server-receive-load.
XDQP Write Load	The execution time (in seconds) of write requests by each host in the cluster. This is the sum of xdqp-client-send-load and xdqp-server-send-load.
Foreign XDQP Read Rate	The amount of data (in MB/sec) read over XDQP by each host in the cluster from a foreign cluster. This is the sum of foreign-xdqp-client-receive-rate and foreign-xdqp-server-receive-rate.
Foreign XDQP Write Rate	The amount of data (in MB/sec) written over XDQP by each host in the cluster to a foreign cluster. This is the sum of foreign-xdqp-client-send-rate and foreign-xdqp-server-send-rate.
Foreign XDQP Read Load	The execution time (in seconds) of read requests by each host in the cluster from a foreign cluster. This is the sum of foreign-xdqp-client-receive-load and foreign-xdqp-server-receive-load.
Foreign XDQP Write Load	The execution time (in seconds) of write requests by each host in the cluster to a foreign cluster. This is the sum of foreign-xdqp-client-send-load and foreign-xdqp-server-send-load.

3.8.6 Database Performance Data

The Overview page displays graphs of the aggregate performance data for all of the databases in the cluster.



The table below describes the charts displayed in the Databases section of the Overview page.

Chart	Description
Fragments	Displays the aggregate number of fragments in all of the databases in the cluster. The displayed lines are: • Active Fragments: The fragments available to queries. • Deleted Fragments: The fragments to be deleted during the next merge operation.
Storage FootPrint	The total disk capacity (in GBs) used by all of the databases in the cluster. The displayed lines are: • Data Size: The amount of data in the forest data directories. • Fast Data Size: The amount of data in the forest fast data directories. • Large Data Size: The amount of data in the forest large data directories.

Chart	Description
Lock Rate	The number of locks set per second across all of the databases in the cluster. The displayed lines are: • Read: The number of read locks set per second. • Write: The number of write locks set per second. • Deadlock: The number of deadlocks per second.
Lock Wait Load	The aggregate time (in seconds) transactions wait for locks; The displayed lines are: • Read: The time transactions wait for read locks. • Write: The time transactions wait for write locks.
Lock Hold Load	The aggregate time (in seconds) locks are held. The displayed lines are: • Read: The time read locks are held. • Write: The time write locks are held.
Deadlock Wait Load	The aggregate time (in seconds) deadlocks remain unresolved.
Database Replication	The amount of data (in MB per second) sent by and received from this cluster and foreign clusters. The displayed lines are: • Database Replication Send: The amount of data sent to foreign clusters. • Database Replication Receive: The amount of data received from foreign clusters.

Click on the  arrow in the upper left-hand section of the DATABASES graph in the Overview page to view graphs that present more detailed performance metrics for each database. The charts displayed on the DATABASES DETAIL page are described in the table below. The metrics for each database in the cluster are displayed as a separate line.

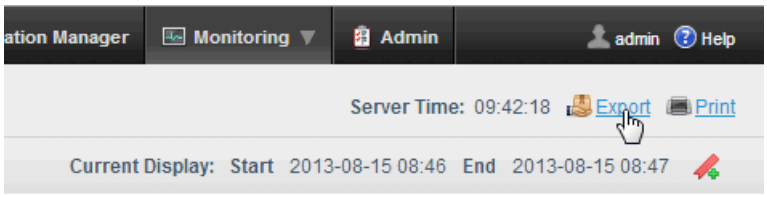
Chart	Description
Active Fragments	The number of active fragments (the fragments available to queries) in each database.
Deleted Fragments	The number of deleted fragments (the fragments to be removed by the next merge operation) in each database.
Data Size	The amount of data in the data directories of the forests attached to each database.
Fast Data Size	The amount of data in the fast data directories of the forests attached to each database.
Large Data Size	The amount of data in the large data directories of the forests attached to each database.
Read Lock Rate	The number of read locks set per second on each database.
Write Lock Rate	The number of write locks set per second on each database.
Deadlock Rate	The number of deadlocks per second on each database.
Read Lock Wait Load	The time (in seconds) transactions wait for read locks on each database.
Write Lock Wait Load	The time (in seconds) transactions wait for write locks on each database.
Deadlock Wait Load	The aggregate time (in seconds) deadlocks remain unresolved on each database.
Read Lock Hold Load	The time (in seconds) read locks are held on each database.
Write Lock Hold Load	The time (in seconds) write locks are held on each database.
Database Replication Send Rate	The amount of replication data (in MB per second) sent by each database to foreign clusters.
Database Replication Receive Rate	The amount of replication data (in MB per second) received by each database from foreign clusters.

Chart	Description
Database Replication Send Load	The time (in seconds) it takes each database to send replication data to foreign clusters.
Database Replication Receive Load	The time (in seconds) it takes each database to receive replication data from foreign clusters.

3.9 Exporting and Printing Monitoring History

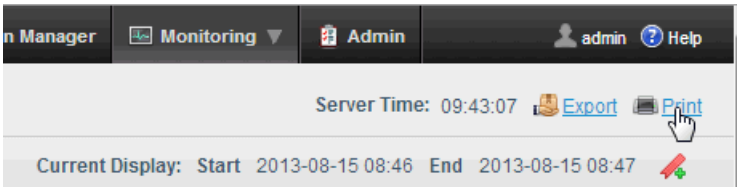
You can export and print your Monitoring History data.

To export the Monitoring History data to an Excel Spreadsheet file, click the Export at the upper-right portion of the Monitoring History page.



The metrics are displayed in separate tabs at the bottom of the spreadsheet.

To print out the charts displayed on the current page, click Print. This will open the printer dialog page from which you can print the charts.



4.0 Configuring Nagios to Monitor MarkLogic Server

Nagios is a popular open source application for monitoring computer systems and networks. MarkLogic provides a Nagios plugin that makes it easy to use Nagios to monitor your MarkLogic Server cluster. This chapter describes how to configure and use Nagios to monitor your MarkLogic Server cluster.

The main topics in this chapter are:

4.1 Terms Used in this Chapter

The following terms are used in this chapter:

- The *Nagios plugin* is a generic Perl script that plugs into your Nagios environment to manage the requests and responses between Nagios and MarkLogic Server. Nagios uses the results returned from the plugin to display the current status of objects in a MarkLogic cluster.
- An *Nagios object* is a particular resource in MarkLogic Server, such as a cluster, host, App Server, or database.
- The *Nagios host* is the computer on which you have installed Nagios and the Nagios plugin.
- The *Monitor Host* is the host in the MarkLogic Server cluster that communicates with the Nagios plugin and returns monitoring information for the objects in the cluster.
- A *resource path* is a URL sent to MarkLogic Server to return monitoring information for an object. The resource paths are described in “Using the Management API” on page 89.
- A *service* describes what to monitor and how to monitor one or more objects in a MarkLogic cluster. Services can define warning and critical thresholds for alerting and can monitor one or more objects in MarkLogic Server.
- A *service group* is a group of one or more services.

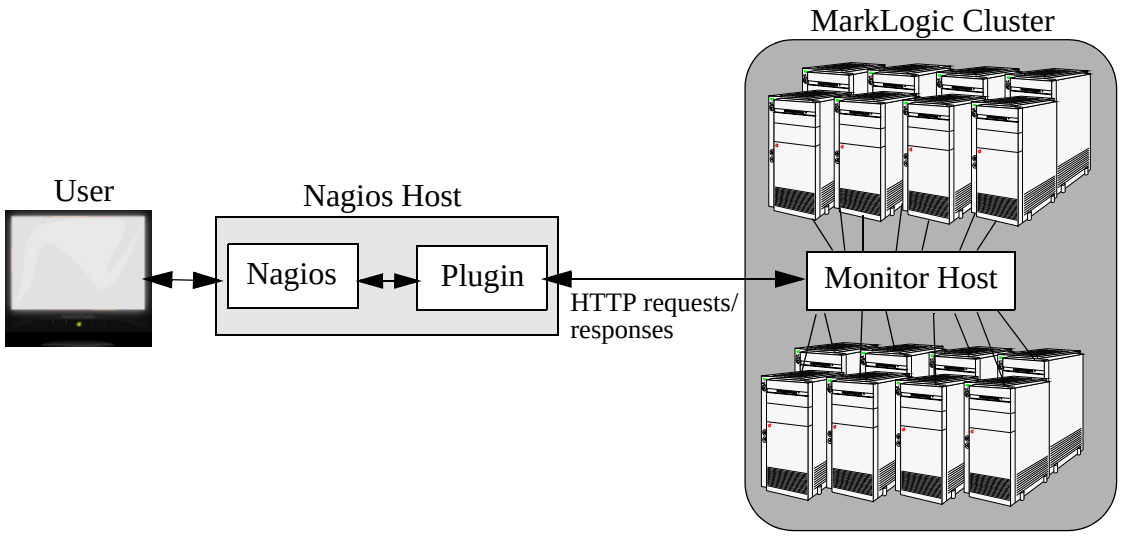
- A *host* describes a MarkLogic Server object, such as a host, database, App Server, or cluster.
- A *host group* is a group of one or more MarkLogic Server objects.

4.2 Overview of the Nagios Plugin Package

This section describes the contents of the Nagios Plugin Package for MarkLogic Server, which you can download from developer.marklogic.com.

The recommended configuration is to install Nagios and the MarkLogic Nagios Plugin on a host outside the MarkLogic cluster. The Nagios plugin communicates via HTTP with a single MarkLogic host that is designated to monitor the entire cluster. This MarkLogic host is referred to as the *monitor host* in this chapter. The Nagios plugin translates a Nagios query into an HTTP request that contains a resource path to be executed by the Management API on the monitor host. The monitor host responds to the request with an XML node that is formatted by the plugin into status and performance data for display by Nagios.

Note: Nagios and the MarkLogic Nagios Plugin must be installed on the same host, unless you use the Nagios Remote Plugin Executor (NRPE) add-on to enable Nagios to communicate remotely with the MarkLogic Nagios Plugin.



For more information on the use of the MarkLogic resource paths, see “Using the Management API” on page 89.

The table below shows the contents of the Nagios Plugin for MarkLogic Server package.

File	Description
`ml_generic.cfg`	Contains the default settings for hosts and services that are inherited by all of the host and service definitions, as well as defines the command to execute the MarkLogic Nagios plugin.
`ml_v7_installation.cfg` | `ml_v5+6_installation.cfg`	An object definition file that contains all of the default settings for monitoring an out-of-the-box MarkLogic Server installation. Use `ml_v7_installation.cfg` with MarkLogic 7 or 8 and `ml_v5+6_installation.cfg` with MarkLogic 5 or 6.
`check_marklogic.pl`	The MarkLogic Nagios Plugin.
`generate_marklogic_config.pl`	The script used to automatically generate the object definition file for your cluster. This is used in the configuration procedure described in “Configuring Nagios for use with MarkLogic Server” on page 68.
`ml_v7_template.xml` | `ml_v5+6_template.xml`	An XML file that defines the services to be monitored. This is used in the configuration procedure described in “Configuring Nagios for use with MarkLogic Server” on page 68. Use `ml_v7_template.xml` to generate configurations compatible with MarkLogic 7 or 8 and `ml_v5+6_template.xml` to generate configurations compatible with MarkLogic 5 or 6.

4.3 Nagios Plugin Requirements

The computer on which you install Nagios and the plugin is referred to in this chapter as the *Nagios host*. This section describes the following requirements for the Nagios Host:

- Nagios Host Supported Platforms
- Nagios Host Library Requirements

4.3.1 Nagios Host Supported Platforms

The Nagios host must be one of the following platforms:

- Red Hat Enterprise Linux 5 (x64)
- SUSE Linux Enterprise Server 11 (x64)
- CentOS 5 (x64)
- Sun Solaris 10 (x64)

The Nagios plugin is not supported on Windows and Mac OS platforms.

The Nagios host can be used to monitor a MarkLogic cluster built on any supported platform.

4.3.2 Nagios Host Library Requirements

Before you can set up Nagios for monitoring your MarkLogic cluster, you must have the following libraries installed on your Nagios host:

- Download and install the cURL executable on your host computer and make sure the executable directory is in your PATH environment variable.

  http://curl.haxx.se/download.html

- Download and install Nagios Core and the Official Nagios Plugins on your Nagios host computer:

  http://www.nagios.org/download

- To execute the Plugin and the configuration generation script, you will need Perl, version 5 and the following modules, which are available from CPAN:
  - Nagios::Plugin::Threshold, version 0.35: http://search.cpan.org/CPAN/authors/id/T/TO/TONVOON/Nagios-Plugin-0.35.tar.gz
  - Getopt::Long, version 2.38: http://search.cpan.org/CPAN/authors/id/J/JV/JV/Getopt-Long-2.38.tar.gz
  - List::Util, version 1.23: http://search.cpan.org/CPAN/authors/id/G/GB/GBARR/Scalar-List-Utils-1.23.tar.gz
  - Params::Validate, version 1.08: http://search.cpan.org/CPAN/authors/id/D/DR/DROLSKY/Params-Validate-1.08.tar.gz
  - XML::LibXML, version 1.88: http://search.cpan.org/CPAN/authors/id/S/SH/SHLOMIF/XML-LibXML-1.88.tar.gz
  - List::MoreUtils, version 0.33: http://search.cpan.org/CPAN/authors/id/A/AD/ADAMK/List-MoreUtils-0.33.tar.gz

4.4 Installing the Nagios Plugin

Below is the procedure to install the Nagios plugin for MarkLogic Server. The `ml_v7_installation.cfg` and `ml_v5+6_installation.cfg` files described in this procedure only recognizes the out-of-the-box objects in your initial installation of MarkLogic Server and should only be used to confirm your installation. In order to monitor any objects created after your initial installation, follow the procedure described in either "Configuring Nagios for use with MarkLogic Server" on page 68.

Note: The following procedure assumes you have installed Nagios in the default location.

1. Unzip the Nagios Plugin package you downloaded from developer.marklogic.com.

2. Move the `check_marklogic.pl` plugin to the directory:

```
/usr/local/nagios/libexec
```

3. Confirm that you have executable permission on the `check_marklogic.pl` plugin.

4. Move the `ml_{version}_installation.cfg` and `ml_generic.cfg` files to the directory:

```
/usr/local/nagios/etc/objects
```

5. From the `/usr/local/nagios/libexec` directory, execute the `check_marklogic.pl` script as shown below to verify that you have all of the dependencies installed and can connect to the Management API on the Monitor host. The `check_marklogic.pl` command uses the following parameters, where *`user:pwd`* are the credentials for a user with the `manage-user` role and *`hostname`* is the name of the monitor host in the MarkLogic cluster:

```
perl check_marklogic.pl -a user:pwd -H hostname -p 8002 --path
/manage/v2/databases
```

Note: All text is case-sensitive.

It should return `OK`; if it does not, use the `--verbose` parameter, as follows, to troubleshoot the problem:

Verbose Level	Description
`--verbose 1`	Adds the user input in the status message. Good for debugging the Nagios object definition files. This parameter can also be used in the `check_command` of a host or service definition in order to debug any problems. For details, see "The check_command Parameter" on page 74.

Verbose Level	Description
`--verbose 2`	Adds limited debug messages. Best for command line debugging.
`--verbose 3`	Adds all debug messages. Best for command line debugging and detecting problems with the plugin.

6. Modify the `/usr/local/nagios/etc/nagios.cfg` file to point to your object definition and generic files:

```
cfg_file=/usr/local/nagios/etc/objects/ml_{version}_installation.cfg
cfg_file=/usr/local/nagios/etc/objects/ml_generic.cfg
```

   Where `{version}` is either:

   - `v7` — for MarkLogic 7 and 8
   - `v5+6` — for MarkLogic 5 and 6

7. Edit the `ml_{version}_installation.cfg` file as follows:

   - Replace all instances of `myhost.marklogic.com` with the name or IP address of your monitor host.
   - Replace `user:pw` with the username and password of a user with the `manage-user` role.

8. Test that there are no errors in the object definition file:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

9. Restart Nagios

```
sudo /etc/rc.d/init.d/nagios restart
```

10. Confirm that you can access Nagios, as described in “Using Nagios” on page 84

4.5 Configuring Nagios for use with MarkLogic Server

This section describes how to automatically generate a Nagios object definition file for your cluster.

This section includes the following topics:

4.5.1 The generate_marklogic_config.pl Script

You can use the `generate_marklogic_config.pl` script to automatically generate an object definition file for your MarkLogic cluster. The script reads the `services.xml` file described in "The Monitoring Services File" on page 72, which defines the services to be monitored, the threshold information for each service, and any objects you want to exclude from monitoring. When you run the `generate_marklogic_config.pl` script, it reads all of the objects (hosts, databases, and App Servers) from your cluster and creates an object definition file that can be used by the Nagios plugin to monitor all of the objects using the services specified in the XML services file.

To run the script, make sure you have the required Perl modules, as described in "Nagios Plugin Requirements" on page 65, open a shell window, and enter a command with the following syntax:

```
perl generate_marklogic_config.pl -a user:pwd -H monitorHost -verbose 1
-f services.xml -u prefix -c clusterName -p port >
MyObjectDefinition.cfg
```

Warning The username and password you enter will be stored in the `MyObjectDefinition.cfg` file.

In the above example, the output from the script is directed to the object definition file, `MyObjectDefinition.cfg`. Reference this file from your `nagios.cfg` file and restart Nagios, as described in "Installing the Nagios Plugin" on page 67.

The parameters for `generate_marklogic_config.pl` are described in the table below.

Parameter	Description	Example	Required?
`-a (--authentication)` *`user:pwd`*	The authentication credentials for the user on the host machine selected to monitor the cluster. This user must have the `manage-user` role.	`-a myName:myPwd`	yes
`-H (--host)` *`monitorhost`*	The name or IP address of the Monitor host selected to monitor the cluster.	`-H Monitor-Host`	yes
`-p (--port)` *`portNumber`*	The port number to be used by the Monitor host.	`-p 8002`	yes
`-c (--clustername)` *`clusterName`*	The name you want to assign to the host group. For information on host groups, see the Host Group Definition in the Nagios documentation.	`-c ML-Cluster1`	yes

Parameter	Description	Example	Required?
`-u (--uniqueshortcut)` *`prefix`*	A unique string to be used as a prefix for every object in your cluster. This prefix must be unique across all clusters.	`-u Clu-1`	yes
`-f (--filename)` *`servicesFile`*`.xml`	The name of the XML file that contains the configuration settings for your monitoring services. Omitting the `-f` option will by default select either the `ml_v7_template.xml` or `ml_v5+6_template.xml` configuration file, depending on the host server version of MarkLogic. However, the best practice is to copy the `ml_v7_template.xml` or `ml_v5+6_template.xml` file and edit the copy, which is referred to in this chapter as `services.xml`, and then to use the `-f` option to reference the `services.xml` file.	`-f services.xml`	yes
`-h (--help)`	Display help information.	`-h`	no
`-t (--timeout)` *`seconds`*	Set the number of seconds to wait for each service to execute before returning an error message.	`-t 60`	no
`-V (--version)`	Display the Nagios plugin version.	`-V`	no
`-v (--verbose) {0 | 1 | 2 | 3}`	`-v 0` Returns no debugging information. `-v {1 | 2 | 3}` Validates the check_commands by executing them against the plugin and reports any problematic services. Higher verbosity levels report more debugging information. The plugin must be in the same directory as the `check_marklogic.pl` and `generate_marklogic_config.pl` scripts.	`-v 1`	no

Parameter	Description	Example	Required?
`-ssl (--ssl) {0|1}`	Determine whether to enable SSL on the monitor host. 0: Do not enable SSL (default) 1: Enable SSL. The `manage` App Server must have SSL enabled and use the same certificate as specified by the `-cert` parameter below. For details on configuring SSL, see Configuring SSL on App Servers in the *Administrator's Guide*.	`-ssl 1`	no
`-cert (--sslcertificate) mycertificate.crt`	Specifies the location of the certificate used for SSL access to MarkLogic Server. Only specify this parameter when SSL is enabled (`-ssl 1`). The path given must match that of the certificate on the Nagios host. The `manage` App Server must use the same certificate specified by this parameter. For details creating an SSL certificate, see Configuring SSL on App Servers in the *Administrator's Guide*.	`-cert cert.crt`	Only if SSL is enabled

If you are running the `generate_marklogic_config.pl` script with the `-ssl` and `-cert` options and you receive an error related to a bad connection or an invalid certificate, confirm that the specified certificate is installed on the Nagios host and the correct path to the certificate is specified by `-cert`. If that is all correct, you may have a bad certificate.

You can test the certificate as follows:

- Export the https certificate (in PEM format) from your browser.
- Run a cURL command with the following form on the certificate:

  ```
  curl --cacert /tmp/CertificateName.crt https://example.org
  ```

  The results will confirm whether cURL is connecting properly.

4.5.2 The Monitoring Services File

The `ml_v7_template.xml` or `ml_v5+6_template.xml` file describes the services used to monitor your cluster in XML format. It is recommended that you copy this file and make changes on the copy. This copy is referred to in this chapter as `services.xml`. You can edit the `services.xml` file to add or modify services, add or modify the sample intervals and thresholds, and to exclude specific resources from any of the services used to monitor your cluster.

This section describes the following topics:

4.5.2.1 Globally Excluding Resources

At the top of the `ml_v7_template.xml` and `ml_v5+6_template.xml` file is a global `excludes` element that lists the resources (by name) to be excluded by all of the monitoring services. For example, to exclude the `Schemas` database, the `Admin` App Server in the `Default` group, and the `myhost` host from being monitored by all of the services, the `excludes` element would look like the following:

```
<excludes>
    <exclude resourcetype="Databases" name="Schemas" />
    <exclude resourcetype="Servers" name="Admin?group-id=Default"/>
    <exclude resourcetype="Hosts" name="myhost.marklogic.com"/>
</excludes>
```

4.5.2.2 Service Definitions

The purpose of this section is to describe the service definitions used by the `generate_marklogic_config.pl` script to generate the object definition file described in “Understanding the Generated Object Definition File” on page 81.

For example, the following XML defines the `backup-count` service in the `services.xml` file:

```
<service-template type="Databases" refresh="default">
    <service_description>backup-count</service_description>
    <service_note>Number of backups in progress</service_note>
    <check-command>
        check_command check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v2/databases/$_HOSTMLALIAS$?view=status
-key $SERVICEDESC$
    </check-command>
    <exclude name="Security"></exclude>
</service-template>
```

Assuming you have a default installation of MarkLogic Server, the resulting `backup-count` service generated from the `generate_marklogic_config.pl` script will look like the following in your generated object definition file:

```
define service{
    use             ML-generic-service
    host_name       MyCl-App-Services, MyCl-Documents, MyCl-Fab,
MyCl-Last-Login, MyCl-Modules, MyCl-Schemas, MyCl-Triggers
    service_description backup-count
    notes           Number of backups in progress
    servicegroups   MyCl-Databases
    check_command   check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v2/databases/$_HOSTMLALIAS$?view=status
-key $SERVICEDESC$ $_HOSTMLSSL$ $_HOSTMLTIMEOUT$
}
```

Note: Nagios uses the `$SERVICEDESC$` variable to hold the value of `service_description`. Consequently, in order to use `$SERVICEDESC$` in the resource path, the `service_description` must be an element in the node returned by `-path`.

The `service-template` element contains three attributes:

- `type` — The `servicegroups` value in the resulting service definition is made up from the string specified by the `generate_marklogic_config.pl -u` parameter and the value specified by the `type` attribute. The `type` can be any object you can specify in a resource path, such as `Local-Cluster`, `Foreign-Clusters`, `Groups`, `Servers`, `Hosts`, or `Databases`. For details on resource paths in the Management API, see “Resource Addresses” on page 92.
- `refresh` — Specifies the value for `check_interval` in the resulting service definition. If set to `default`, Nagios uses the value from the `ml_generic.cfg` file.

The `check_command` is described in detail in “The check_command Parameter” on page 74.

The `exclude` element is used to exclude certain resources from this specific service. For example, to exclude the `Security` database from this service, use:

```
<exclude name="Security"></exclude>
```

For details on how to exclude resources on a global level, see “Globally Excluding Resources” on page 72.

4.5.2.3 The check_command Parameter

This section provides more detail on the `check_command` defined in a service definition in the `services.xml` file.

The `check_command` contains the information used by the plugin to construct the HTTP request sent to MarkLogic Server. For example, the `update-count` service defines the `check_command` as follows:

```
check_command   check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v2/requests?server-id=$_HOSTMLALIAS$
-key $SERVICEDESC$
```

The `check_marklogic.pl` script identifies the plugin that translates the monitoring requests and responses between MarkLogic Server and Nagios. The name of the plugin is defined in the `ml_generic.cfg` file and it must be specified in every `check_command`.

The `check_command` contains what Nagios refers to as *custom variable macros* and *standard macros*. The custom variable macros used by the `-a`, `--port`, and `--host` parameters represent the login credentials, port number, and MarkLogic Server host name (or IP address) set in the `MyCl_abstract` host definition. The `--path` specifies the resource path to be sent to MarkLogic Server. The `$_HOSTMLALIAS$` is a custom variable macro that specifies the resource to be monitored.

The `$SERVICEDESC$` macro is a standard macro used by Nagios to hold the value of the `service_description` (see the definition of the $SERVICEDESC$ macro in the Nagios documentation). The `service_description` in this example is `update-count`, so when Nagios checks the `update-count` service for the four App Servers, it will replace the macros and call the plugin four times with the following resource addresses:

```
http://localhost:8002/manage/v2/requests?server-id=Admin
http://localhost:8002/manage/v2/requests?server-id=App-Services
http://localhost:8002/manage/v2/requests?server-id=Manage
http://localhost:8002/manage/v2/requests?server-id=TaskServer
```

The Management API is described in detail in “Using the Management API” on page 89. See Understanding Macros and How They Work and Standard Macros in Nagios in the Nagios documentation for details on custom variable and standard macros.

The following table lists the possible parameters that can be used by `check_marklogic.pl` script. The parameters that are required in the `check_marklogic.pl` script are flagged by "(Required)." The parameters related to thresholding and ranges are described in more detail in "Defining and Setting Thresholds and Ranges" on page 78.

Parameters	Description
`-a (--authentication)` *`user:pwd`* (Required)	Authentication for the user on the monitor host that is set in the abstract host definition (required). This is captured in the `$_HOSTMLUSERPW$` macro.
`-H (--host)` *`hostname`* (Required)	Name or IP address of the monitor host that is set in the abstract host definition (required). This is captured in the `$_HOSTMLIP$` macro.
`-p (--port)` *`port`* (Required)	Port number used by the Management API on the monitor host that is set in the abstract host definition (required). This is captured in the `$_HOSTMLPORT$` macro.
`--path` *`path`* (Required)	The resource address path (required).
`-ssl (--ssl) {0 | 1}`	0: Do not enable SSL (default) 1: Enable SSL
`-cert (--sslcertificate)` *`mycertificate`*`.crt`	Specifies the location of the certificate used for SSL access to MarkLogic Server. Only specify this parameter when SSL is enabled (`-ssl 1`).
`-t (--timeout)` *`timeout-seconds`*	The timeout, in seconds, that defines the maximum time to wait for a response to a service. (default is 10).
`-v (--verbose) {0 | 1 | 2 | 3}`	0: Return no debugging information (default 0). 1: Returns the user input in the status message. {2 | 3}: Returns various levels of additional debugging information.

Parameters	Description
`-k (--key) element` (Required if thresholds are used)	The element value to return for the specified resource. You can specify any simple or complex element in the node returned in response to the resource path. You can optionally use the `-op` parameter below to inspect the element value and determine a threshold.
`-op (--operator) {range | {{eq | ne}=string}}`	Specifies that a threshold is to be determined by a range of values, or by the presence or absence of a specific string. The default for `-op` is `range`. For details, see “Defining and Setting Thresholds and Ranges” on page 78.
`-w (--warning) range`	The range used to inspect the threshold value to determine whether flag the object with a warning. For details, see “Defining and Setting Thresholds and Ranges” on page 78.
`-c (--critical) range`	The range used to inspect the threshold value to determine whether flag the object as critical. For details, see “Defining and Setting Thresholds and Ranges” on page 78.

You can test the results of a `check_command` by navigating to the `/usr/local/nagios/libexec` directory and executing the `check_marklogic.pl` script using the following format:

```
perl check_marklogic.pl  -a user:pwd --port 8002  --host hostName
--path /manage/v2/URI [--key resource [-op range -w value -c value]]
```

For example, to return the `database-counts` node, you can enter:

```
perl check_marklogic.pl  -a admin:admin  -port 8002
--host gordon-1 -path  /manage/v2/databases/Documents?view=counts
```

You will see a result like:

```
OK - Documents-elapsed-time=0.057927s |
Documents-elapsed-time=0.057927s; Documents-documents=590003;
Documents-directories=5; Documents-active-fragments=118001 1;
Documents-deleted-fragments=16; Documents-nascent-fragments=0;
```

To return the `documents` value in the `database-counts` node, you can enter:

```
perl check_marklogic.pl  -a admin:admin  -port 8002
--host gordon-1 -path  /manage/v2/databases/Documents?view=counts
--key documents
```

You will see a result like:

```
OK - Documents-documents=590003 | Documents-documents=590003;;
```

To test a threshold, set a threshold value higher than the document count. For example, enter:

```
perl check_marklogic.pl  -a admin:admin  -port 8002  --host gordon-1
-path  /manage/v2/databases/Documents?view=counts -key documents
-op range -w 10000: -c 3000000:
```

This should result in a critical message like the following:

```
Critical - documents=590003 [critical(3000000:)][warning(10000:)] |
Documents-documents=590003;10000:;3000000:;
```

You can use the same approach to test other thresholds.

4.5.2.4 Defining and Setting Thresholds and Ranges

The syntax for setting thresholds and ranges in Nagios is described in the Threshold and ranges section in the Nagios documentation. The purpose of this section is to describe the parameters that are specific to setting thresholds and ranges in the object definition file used by the `check_marklogic.pl` plugin.

> **Note:** Thresholds and ranges should be set in the `services.xml` file, as described in "The Monitoring Services File" on page 72

The parameters used by `check_command` to set thresholds and ranges in the `services.xml` file are shown in the following table.

Parameter	Description
`-k (--key) element`	The element value to inspect and determine a threshold. This can be any element in the node returned in response to the resource path.
`-op (--operator) {range | {{eq | ne}=string}}`	Defines whether the threshold is to be determined by a range of values, or by the presence or absence of a specific string. This must follow a `--key` parameter.

The thresholds are set in a separate `threshold` element and they are:

Threshold Element	Description
`<default-warning>`	The range used to inspect the threshold value to determine whether flag the object with a warning.
`<default-critical>`	The range used to inspect the threshold value to determine whether flag the object as critical.
`<custom-threshold name="resource" warning="value" critical="value">`	The threshold ranges to apply to a specific resource.

For example, the `documents` service defined in the `services.xml` file returns the number of documents stored in the databases. The default service definition in your `services.xml` file is:

```
<service-template type="Databases" refresh="60">
    <service_description>documents</service_description>
    <service_note>
        Document count for attached forests (excluding replicas)
    </service_note>
    <check-command>check_command   check_marklogic.pl!
-a $_HOSTMLUSERPW$  -port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v2/databases/$_HOSTMLALIAS$?view=counts
-key $SERVICEDESC$
    </check-command>
</service-template>
```

If you want to define a threshold to generate a warning if the document count on any of the databases monitored by the `documents` service exceeds 1000 documents and a critical warning if the document count exceeds 10000, then you can add a `threshold` element to your `services.xml` file as follows:

```
<service-template type="Databases" refresh="60">
    <service_description>documents</service_description>
    <service_note>
        Document count for attached forests (excluding replicas)
    </service_note>
    <check-command>check_command   check_marklogic.pl!
-a $_HOSTMLUSERPW$  -port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v2/databases/$_HOSTMLALIAS$?view=counts
-key $SERVICEDESC$
    </check-command>
    <threshold>
       <default-critical>10000</default-critical>
       <default-warning>1000</default-warning>
    </threshold>
</service-template>
```

Note: The `check_command` must be defined as a continuous line with no returns.

The `check_commend --key` parameter specifies the XML element on which to apply a threshold. In the above example, the name of this element is `documents`, which is defined in the `service_description` and stored in the `$SERVICEDESC$` macro (see the definition of the $SERVICEDESC$ macro in the Nagios documentation). The `--key` element must be an element in `database-counts` node returned by the resource path, `/manage/v2/databases/Documents/counts`.

Another type of threshold is to determine either the presence or absence of a particular string. The -`op eq=`*`string`* and `-op ne=`*`string`* parameters evaluate values as strings, calculating equality or inequality, respectively, and convert the value to either 0 (false) or 1 (true). For example, the `state` service used to detect the state of the databases makes use of the `-op ne=unavailable` parameter and the `default-critical @0:0` element defined for each database to generate a critical flag on any database that is not in the 'available' state (boolean value of 0).

```
<service-template type="Databases" refresh="default">
    <service_description>state</service_description>
    <service_note>State of the database</service_note>
    <check-command>check_command   check_marklogic.pl!
-a $_HOSTMLUSERPW$  -port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v2/databases/$_HOSTMLALIAS$?view=status
-key $SERVICEDESC$ -op ne=unavailable
    </check-command>
    <threshold>
    <default-critical>@0:0</default-critical>
    </threshold>
</service-template>
```

You can also set a custom threshold for specific objects. For example, below, the `default-critical` and `default-warning` elements set the thresholds for all of the objects, which are databases in this example. The `custom-threshold` element overrides the default thresholds and sets the given thresholds for only the `Documents` database.

```
<threshold>
    <default-critical>@10:399</default-critical>
    <default-warning>@400:1000</default-warning>
    <custom-threshold name="Documents" warning="@5000:7000"
critical="@7001:10000"></custom-threshold>
</threshold>
```

You can create a custom threshold for more than a single object. For example, the `query-count` service calls the following method to return the query count for each App Server in the cluster:

```
-path /manage/v2/requests?server-id=$_HOSTMLALIAS$ -key $SERVICEDESC$
```

You have a number of App Servers in your cluster and you want to set a different threshold for the Admin App Server `Default` group. To set a custom threshold for the Admin App Servers in the `Default` group, you can define a `custom-threshold` with the following name (note the need to escape '`&`'):

```
<custom-threshold name="Admin\\&group-id=Default" ....
```

This would result in a resource path like the one below to determine whether the query count for the Admin App Server in the `Default` group has fallen outside of the specified thresholds.

```
-path /manage/v2/requests?server-id=Admin\\&group-id=Default
-key $SERVICEDESC$
```

4.6 Understanding the Generated Object Definition File

This section describes the object definition file generated by the following call to the `generate_marklogic_config.pl` script on a default installation of MarkLogic Server on a single host.

```
perl generate_marklogic_config.pl -a admin:admin -H gordon-1
-f ml_v7_template.xml -u MyCl -c ML-Cluster1
-p 8002 > MyObjectDefinition.cfg
```

Note: Manual edits to an object definition file are not recommended, as it is easy to introduce errors and changes are hard to manage.

The Nagios process that monitors an object in MarkLogic Server is called a *service*. Services are grouped into *service groups*. The `MyObjectDefinition.cfg` object definition file generated from the above script defines the following service groups:

- MyCl-Servers
- MyCl-Databases
- MyCl-Hosts
- MyCl-Local-Cluster
- MyCl-Foreign-Cluster

The `data-size` service looks like the following:

```
define service{
    use             ML-generic-service
    host_name       MyCl-App-Services, MyCl-Documents, MyCl-Fab,
MyCl-Last-Login, MyCl-Modules, MyCl-Schemas, MyCl-Security,
MyCl-Triggers
    service_description data-size
    notes           Total size of forest data on disk (MB)
    servicegroups   MyCl-Databases
    check_command   check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v2/databases/$_HOSTMLALIAS$?view=status
-key $SERVICEDESC$   $_HOSTMLSSL$ $_HOSTMLTIMEOUT$
    check_interval 10
}
```

The second line in the service definition, `use ML-generic-service`, specifies that the service inherits default settings from the `ml_generic.cfg` file. These default settings are used unless expressly overridden for a host or service in your `MyObjectDefinition.cfg` file. For example, the value of the default `check_interval` setting is 1, but the definition of the `data-size` service above overrides the default and sets the `check_interval` value to 10.

The `host_name` portion of the service definition defines which objects are monitored by this service. The `service_description` is the name of the service that appears in the Nagios UI and `notes` is a simple comment that describes the service.

Nagios allows you to aggregate services into groups, known as *service groups*. The `servicegroups` value specifies that this service belongs to the group of database monitoring services, known as `MyCl-Databases`, for the cluster, `MyCl`. See Service Group Definition in the Nagios documentation for more detail on service groups. The `check_command` portion defines the specific monitoring method to be used on those objects. The `check_command` is described in more detail in "The check_command Parameter" on page 74.

The abstract host definition shown below defines the port number, login credentials, and name of the monitor host in the cluster to execute the monitoring methods and relay the results to the plugin for use by Nagios.

For example, the abstract host definition might look like:

```
define host{
    use         ML-generic-host
    name        MyCl_abstract
    _MLPORT     8002;
    _MLUSERPW   usr:pwd;
    _MLIP       gordon-1;
    _MLSSL
    _MLTIMEOUT
    register    0;
}
```

Like the services, the abstract host definition inherits default settings from the `ml_generic.cfg` file, which are used unless they are expressly overridden in the host definition in your `MyObjectDefinition.cfg` file.

Following the abstract host definition is a series of `define host` definitions that describe which objects to monitor on the cluster.

> **Note:** In Nagios, every object, whether it is a MarkLogic Server host, App Server, database, or the overall cluster, is referred to as a 'host', so you must distinguish between the abstract host definition for the monitor host, `MyCl_abstract`, above from the host definitions for the MarkLogic Server objects, like the one shown for the `Documents` database below.

For example, the definition that directs Nagios to monitor the `Documents` database looks like:

```
define host{
    use                     MyCl_abstract
    host_name               MyCl-Documents
    address                 MyCl-Documents
    hostgroups              ML-Cluster1
    check_command check_marklogic.pl!  -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$ $_HOSTMLSSL$ $_HOSTMLTIMEOUT$
-path /manage/v2/databases/Documents?view=status -key state -c @0:0
-op eq=available
    _MLALIAS    Documents

    _CRITICAL-STATE @0:0
    _CRITICAL-FAILED-MASTERS 0:0
    _CRITICAL-ASYNC-REPLICATING 0:0
    _CRITICAL-DATABASE-REPLICATION-ACTIVE @0:0
    _WARNING-FOREIGN-FORESTS-LAG-EXCEEDED 0:0
}
```

The `use MyCl_abstract` parameter specifies that the host definition is to inherit all of the setting from the abstract host, `gordon-1`, defined earlier. The `host_name` identifies this database as `MyCl-Documents`. The `MyCl` prefix is used to distinguish between different `Documents` databases in different clusters.

The `address` value, `MyC1-Documents`, is a dummy value that is required by Nagios but is not used in this configuration. The `hostgroups` value, `ML-Cluster1`, specifies that this database belongs to this group of hosts. You don't need to be concerned with this name, unless you are going to include more than one object definition file in your `nagios.cfg` file, in which case the names specified for your `host_names` and `hostgroups` in each object definition file must be unique.

The `check_command` is used to report whether the resource is enabled. The `_MLALIAS` variable specifies that the object to be monitored is the `Documents` database. This variable can be used as part of a macro (`_HOSTMLALIAS`) in the resource path so that multiple objects can be monitored by a single service, as described in "The check_command Parameter" on page 74.

The macros `_CRITICAL-STATE`, `_CRITICAL-FAILED-MASTERS 0:0`, `CRITICAL-ASYNC-REPLICATING 0:0`, `_CRITICAL-DATABASE-REPLICATION-ACTIVE @0:0`, and `_WARNING-FOREIGN-FORESTS-LAG-EXCEEDED 0:0` define the thresholds for this resource. Thresholding is described in "Defining and Setting Thresholds and Ranges" on page 78.

4.7 Updating a Previously Generated Object Definition File

If you have an object definition file generated for MarkLogic 6 or earlier, you will need to made some modifications before you can use it on MarkLogic 7.

Edit the object definition file and perform a find and replace, as follows:

- `/v1/` with `/v2/`
- `/status` with `?view=status`
- `/counts` with `?view=counts`
- `on-disk-size` with `data-size`
- remove `total-` from any status (key) name

4.8 Using Nagios

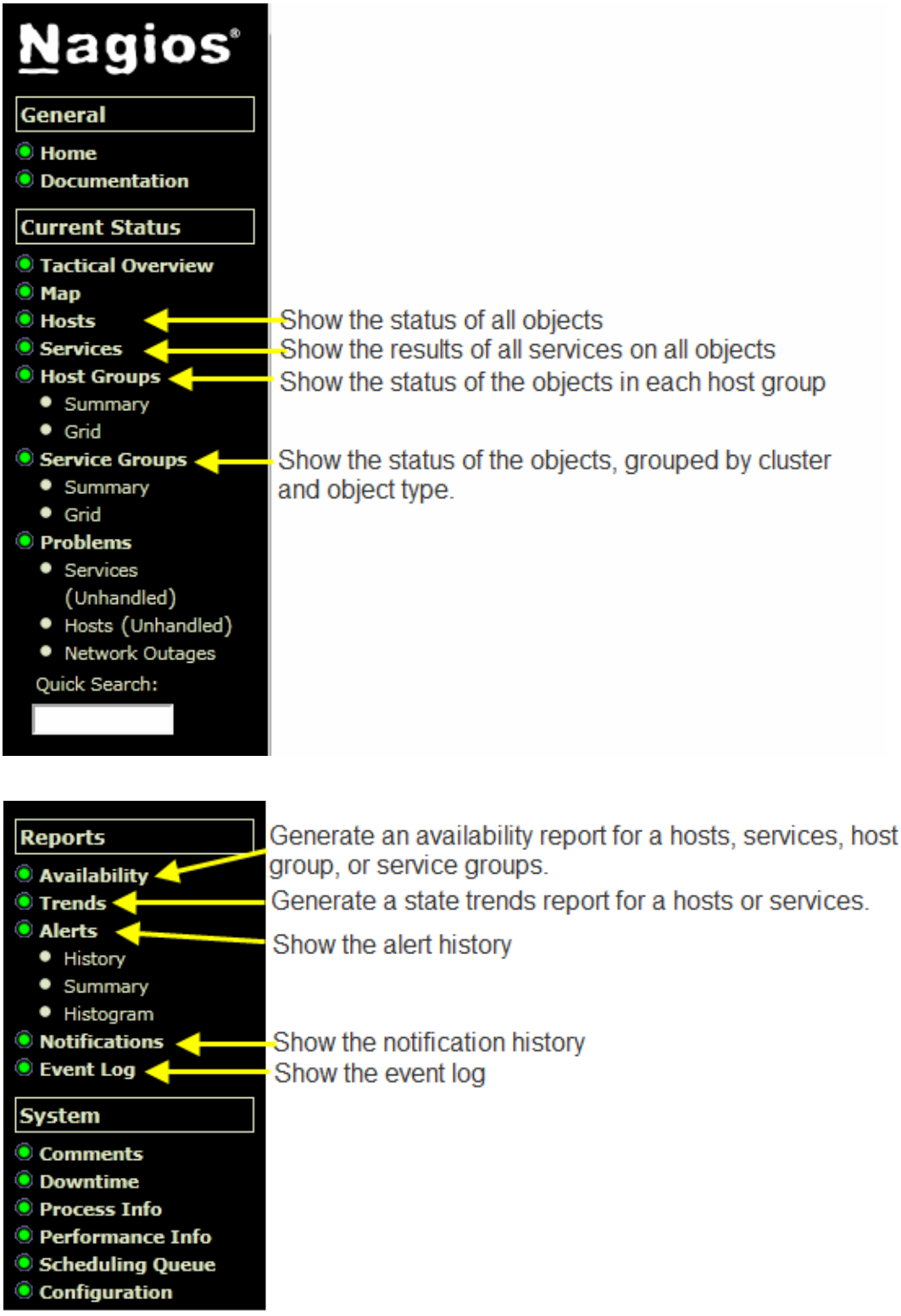
For details on how to use the Nagios User Interface, see the Nagios Core 3.x Documentation. To access Nagios, enter a URL with the following format:

```
http://hostName/nagios/
```

This section contains the following topics:

4.8.1 Nagios Navigation Panels

The navigation panel on the left side of the page looks like:



4.8.2 Host Groups

The Host Groups page displays all of the Nagios services grouped by cluster. For example, the Host Groups below displays all the services for both the `FooMe` and `FooYou` clusters.

Service Overview For All Host Groups

ML-FooM (FooMe)

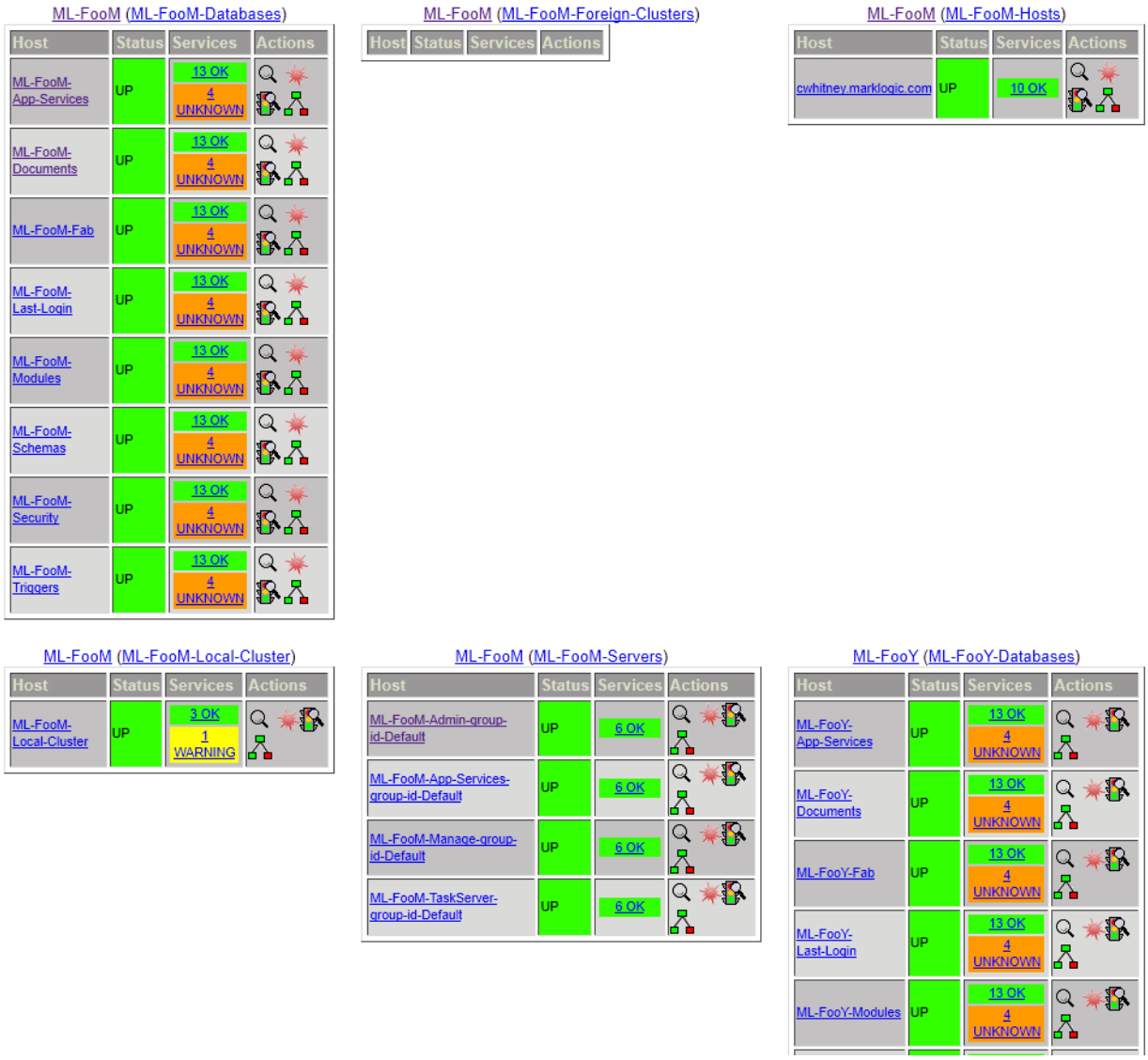
Host	Status	Services	Actions
ML-FooM-Admin-group-id-Default	UP	6 OK	
ML-FooM-App-Services	UP	13 OK 4 UNKNOWN	
ML-FooM-App-Services-group-id-Default	UP	6 OK	
ML-FooM-Documents	UP	13 OK 4 UNKNOWN	
ML-FooM-Fab	UP	13 OK 4 UNKNOWN	
ML-FooM-Last-Login	UP	13 OK 4 UNKNOWN	
ML-FooM-Local-Cluster	UP	3 OK 1 WARNING	
ML-FooM-Manage-group-id-Default	UP	6 OK	
ML-FooM-Modules	UP	13 OK 4 UNKNOWN	
ML-FooM-Schemas	UP	13 OK 4 UNKNOWN	
ML-FooM-Security	UP	13 OK 4 UNKNOWN	
ML-FooM-TaskServer-group-id-Default	UP	6 OK	
ML-FooM-Triggers	UP	13 OK 4 UNKNOWN	
cwhitney.marklogic.com	UP	10 OK	

ML-FooY (FooYou)

Host	Status	Services	Actions
ML-FooY-Admin-group-id-Default	UP	6 OK	
ML-FooY-App-Services	UP	13 OK 4 UNKNOWN	
ML-FooY-App-Services-group-id-Default	UP	6 OK	
ML-FooY-Documents	UP	13 OK 4 UNKNOWN	
ML-FooY-Fab	UP	13 OK 4 UNKNOWN	
ML-FooY-Last-Login	UP	13 OK 4 UNKNOWN	
ML-FooY-Local-Cluster	UP	3 OK 1 WARNING	
ML-FooY-Manage-group-id-Default	UP	6 OK	
ML-FooY-Modules	UP	13 OK 4 UNKNOWN	
ML-FooY-Schemas	UP	13 OK 4 UNKNOWN	
ML-FooY-Security	UP	13 OK 4 UNKNOWN	
ML-FooY-TaskServer-group-id-Default	UP	6 OK	
ML-FooY-Triggers	UP	13 OK 4 UNKNOWN	
wlan31-13-237.marklogic.com	UP	10 OK	

4.8.3 Service Groups

The Service Groups page displays the services groups by resource type on each cluster. For example, the Service Groups below displays the services for the databases, servers, hosts, and local/foreign clusters in separate groups for both the `FooMe` and `FooYou` clusters.



4.8.4 Service Status Details for a Resource

Below is a detailed view of the services for the Documents database in the `FooMe` cluster. Note that some services may report a status of UNKNOWN. This indicates that the monitoring metric is unavailable to Nagios for that resource. For example, the `database-replication-active` service shown below reports UNKNOWN because Database Replication is not configured for the database. In this example, if you do not have a license for Database Replication, you can remove the `database-replication-active` service from your `services.xml` file. If you have a license for Database Replication, but only have the feature enabled for some databases, you can exclude the databases that are not replicated from being monitored, as described in "The Monitoring Services File" on page 72.

Service Status Details For
Host 'ML-FooM-Documents'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ML-FooM-Documents	async-replicating	UNKNOWN	10-11-2011 12:08:56	0d 23h 10m 3s	10/10	The XML element (async-replicating) was not present in the data returned by the API (curl "http://cwhitney:8002/manage/v1/databases/Documents/status" --max-time 10 --user [username:pw] -i --anyauth -H "Accept: application/xml" --silent 2>&1)
	backup-count	OK	10-11-2011 12:10:36	0d 4h 51m 23s	1/10	OK - Documents-backup-count=0
	compressed-tree-cache-hit-rate	OK	10-11-2011 12:10:11	0d 4h 51m 48s	1/10	OK - Documents-compressed-tree-cache-hit-rate=0
	compressed-tree-cache-miss-rate	OK	10-11-2011 12:10:48	0d 4h 51m 11s	1/10	OK - Documents-compressed-tree-cache-miss-rate=0
	database-replication-active	UNKNOWN	10-11-2011 12:01:26	0d 23h 7m 33s	10/10	XML element (database-replication-active) was not present in the data returned by the API
	documents	OK	10-11-2011 12:04:04	0d 20h 6m 55s	1/10	OK - Documents-documents=0
	failed-masters	UNKNOWN	10-11-2011 12:02:41	0d 23h 6m 18s	10/10	The XML element (failed-masters) was not present in the data returned by the API (curl "http://cwhitney:8002/manage/v1/databases/Documents/status" --max-time 10 --user [username:pw] -i --anyauth -H "Accept: application/xml" --silent 2>&1)
	foreign-forests-lag-exceeded	UNKNOWN	10-11-2011 12:03:18	0d 23h 5m 41s	10/10	The XML element (foreign-forests-lag-exceeded) was not present in the data returned by the API (curl "http://cwhitney:8002/manage/v1/databases/Documents/status" --max-time 10 --user [username:pw] -i --anyauth -H "Accept: application/xml" --silent 2>&1)
	list-cache-hit-rate	OK	10-11-2011 12:09:56	0d 4h 51m 3s	1/10	OK - Documents-list-cache-hit-rate=0
	list-cache-miss-rate	OK	10-11-2011 12:10:36	0d 4h 51m 23s	1/10	OK - Documents-list-cache-miss-rate=0
	load-detail	OK	10-11-2011 12:10:08	0d 4h 51m 51s	1/10	OK - Documents-total-query-read-load=0
	merge-count	OK	10-11-2011 12:10:42	0d 4h 51m 17s	1/10	OK - Documents-merge-count=0
	on-disk-size	OK	10-11-2011 12:10:19	0d 18h 30m 40s	1/10	OK - Documents-on-disk-size=0MB
	rate-detail	OK	10-11-2011 12:09:57	0d 4h 51m 2s	1/10	OK - Documents-total-query-read-rate=0
	reindex-count	OK	10-11-2011 12:01:35	0d 20h 19m 24s	1/10	OK - Documents-reindex-count=0
	restore-count	OK	10-11-2011 12:02:12	0d 18h 38m 47s	1/10	OK - Documents-restore-count=0
	state	OK	10-11-2011 12:10:50	0d 4h 51m 9s	1/10	OK - state=1 [critical(@0:0)]

5.0 Using the Management API

The Management API is a REST-based API that allows you to access MarkLogic Server instrumentation with no provisioning or set-up. The API provides the ability to easily capture detailed information on MarkLogic Server objects and processes, such as hosts, databases, forests, App Servers, groups, transactions, and requests from a wide variety of tools. The Monitoring Dashboard and Nagios plugin described in this guide are implemented on top of the Management API.

This chapter describes how to use the Management API to obtain monitoring data from MarkLogic Server. This chapter includes the following sections:

5.1 Terms used in this Chapter

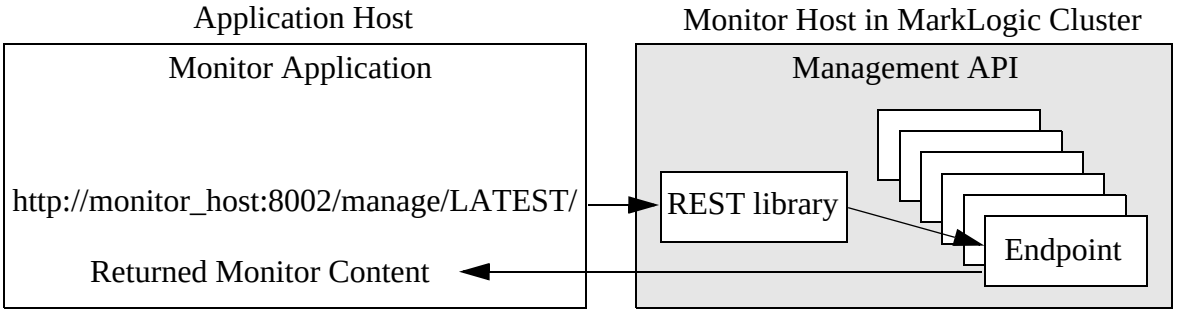
- An *Object* is a component of interest in MarkLogic Server, such as a cluster, host, App Server, or database.
- A *Process* is a request or transaction in MarkLogic Server.
- *Monitor Content* is the XML, HTML, or JSON structure that represents the data returned by the Management API.
- A *Monitor Application* can be any application that requests and makes use of monitoring data, such as a Web browser, a plugin for an existing monitoring tool, or the Monitoring Dashboad described in “Using the MarkLogic Server Monitoring Dashboard” on page 16.
- The *Monitor Host* is the host in the MarkLogic Server cluster that uses the Monitor API to respond to requests for monitoring content from the monitor application.
- The *Manage App Server* is the App Server on the Monitor Host that is configured to handle monitor requests. The Manage App Server is bound to port 8002 and is the App Server used by the Nagios plugin and the Monitoring Dashboard.

- *REST* stands for *Representational State Transfer*, which is an architecture style that, in the context of monitoring MarkLogic Server, describes the use of HTTP to make calls between a monitoring application and monitor host.
- A *Resource* is an abstraction of a MarkLogic Server object, as presented by the REST architecture.
- A *Resource Address* is a URL that identifies a MarkLogic Server resource. The resource addresses are described in "Resource Addresses" on page 92.
- A *View* is the returned monitoring information about a resource. You can have different views of the same resource. A view can be for a single resource (known as an *item view*) or a number of resources (known as a *list view*).
- A *Representation* is a view of a resource in a particular format, such as XML, HTML, or JSON.
- A *Parameter* is an addition to the end of a resource address to filter and/or format the view returned from MarkLogic Server. Parameters are expressed as query strings in the URL and are described in "Specifying Parameters in a Resource Address" on page 94.
- An *Endpoint* is an XQuery module on MarkLogic Server that is invoked by and responds to an HTTP request for monitoring information.
- A *Plugin* is an XQuery module that provides extension capabilities using the Plugin framework described in the System Plugin Framework chapter in the *Application Developer's Guide*.

5.2 Overview of the Management API

The Management API is implemented on top of the REST Library described in Creating an Interpretive XQuery Rewriter to Support REST Web Services in the *Application Developer's Guide*. Requests to monitor an object in MarkLogic Server are made by means of a resource address that returns a view containing the monitor data for the object. The view can be returned in various formats, such as XML, HTML, or JSON.

Every resource address in the Management API invokes a monitoring endpoint, which is an XQuery module on the target MarkLogic Server host. The monitoring endpoints are invoked by a resource address in an application, such as a browser. The Management API uses the REST library to validate the request, authorize the user, and rewrite the resource address to one understood by the monitoring framework before invoking the endpoint module. The endpoint module returns the monitoring data for the resource to the application.



5.3 Security

As described in "Monitoring Tools and Security" on page 6, client access to a Management API and endpoints requires that they authenticate as a user with the `manage-user` role. If custom Plugin code requires additional privileges, you can create and assign a custom role to users of the Management API to enable that functionality.

If you have enabled SSL on the `Manage` App Server, your resource address must start with HTTPS, rather than HTTP, and you must have a MarkLogic certificate authority on your browser, as described in Accessing an SSL-Enabled Server from a Browser or WebDAV Client in the *Administrator's Guide*.

5.4 Management API Requires Writing to the App-Services Database

The management API sometimes writes documents to the App-Services database for internal purposes, and it therefore assumes that the App-Services database is writable. On each cluster in which the management API runs (which might include a replica cluster), it requires a writable view of the App-Services database. The App-Services database is used to store data used by various MarkLogic applications, such as Query Console. For these reasons, MarkLogic recommends that you do not replicate the App-Services database using database replication. If the App-Services database is not available, it falls back to the schemas-database configured for the schemas-database of the App Server under which the management API is running.

5.5 Resource Addresses

This section provides an overview of the structure and capabilities of the resource addresses provided by the Management API. For details on each resource address, see the *MarkLogic REST API Reference*.

Resource addresses fall into the categories listed in the table below. The output from each type of resource address is described in "Interpreting the Output" on page 97.

Type of Resource Address	Returns
List	A list of resources. For example, as list of the forests in the cluster: `http://localhost:8002/manage/LATEST/forests`
Item	A specific resource. For example, a specific forest in the cluster: `http://localhost:8002/manage/LATEST/forests/Documents`

A resource address takes the form of a URL that includes a host name and a port number. The most basic resource address returns summary information for the entire cluster:

```
http://host:port/manage/LATEST/
```

The Management API version, `LATEST` in this release, is specified in every resource address to maintain compatibility with future revisions of the Management API.

You can optionally include the name of a resource and parameters in a resource address as follows:

```
http://host:port/manage/version/resource?param=value&param=value
```

5.6 Obtaining the Options Node for a Resource Address

As described in Creating an Interpretive XQuery Rewriter to Support REST Web Services in the *Application Developer's Guide*, the REST Library uses an `options` node to map incoming requests to endpoints. The `options` node contains information about the communication options available on the request/response chain for the resource address, such as which parameters can be specified with the resource address.

You can use the `xdmp:http-options` function to output the `options` node for any resource address. For example, you can enter the following query in Query Console to display the `options` node for the `/manage/LATEST/transactions` resource address:

```
xdmp:http-options(
  "http://localhost:8002/manage/LATEST/transactions",
  <options xmlns="xdmp:http">
    <authentication method="digest">
      <username>admin</username>
      <password>admin</password>
    </authentication>
  </options>)
```

The output will include a `request` element that defines the options associated with the GET and HEAD methods for the resource address. From this, you can determine the supported parameters and values. For example, the above resource address supports the `view`, `seconds-min`, `host-id`, `fullrefs`, and `format` parameters, as shown below.

```
<rest:http method="GET">
  <rest:param name="view" values="default" default="default"/>
  <rest:param name="seconds-min" as="string"/>
  <rest:param name="host-id" as="string"/>
  <rest:param name="fullrefs" as="boolean" required="false"/>
  <rest:param name="format" as="string" values="xml|json|html"/>
  <rest:or>
    <rest:accept>application/xml</rest:accept>
    <rest:accept>application/json</rest:accept>
    <rest:accept>text/html</rest:accept>
    <rest:accept>application/x-javascript</rest:accept>
  </rest:or>
</rest:http>
```

5.7 Specifying the Management API Version

To guarantee stable behavior of the Management API as new versions are released, each resource address in the Management API includes a version number. The examples in this chapter show the version as LATEST, which means to use the latest version of the API. However, you can also specify the version number to use a specific version of the API, using the format:

```
v#
```

Where # is the version number. For example, in the initial version of the API:

```
http://localhost:8002/manage/LATEST/databases
```

is the same as:

```
http://localhost:8002/manage/v2/databases
```

If you want to update your clients when you choose, use the explicit version number. If you want to update your clients to the most recent version of the Management API, use LATEST.

> **Note:** The version number is only updated when resource addresses and/or parameters have changed. It is not updated when resource addresses and/or parameters are added or removed.

5.8 Specifying Parameters in a Resource Address

resource addresses can take parameters to do the following:

- Specify the format of the returned view
- Return a filtered view

To specify multiple parameters, use the '?' sign before the first parameter and the '&' sign before any additional parameters:

```
http://host:port/manage/LATEST/resource?param1=value&param2=value....
```

Some resource addresses support optional parameters that are specific to that resource address. For example, to return monitoring information on the forests used by the Documents database, you can use the database-id parameter with the /forests resource as follows:

```
http://monitor_host:8002/manage/LATEST/forests?database-id=Documents
```

The remainder of this section describes the format parameter in more detail.

5.8.1 Formatting the Monitor Results

The application that issues a request to the Management API specifies the format for the returned view. For example, most Web browsers specify the default return format as HTML. If no return format is specified by the application, the view is formatted as XML. You can explicitly specify the view format by means of the `format` parameter at the end of the resource address:

```
format=value
```

Where *value* is either HTML, JSON, or XML.

The XML and JSON formats provide a rich set of data for your monitoring application. For example, to return an XML view of the entire cluster, you can enter the following in a browser:

```
http://monitor_host:8002/manage/LATEST?format=xml
```

This will return a `cluster` view in XML format. For example:

```
<cluster-default xsi:schemaLocation=
  "http://marklogic.com/manage/clusters manage-clusters.xsd">
  <meta>
    <uri>/manage/LATEST</uri>
    <current-time>2011-06-30T16:00:06.81-07:00</current-time>
    <elapsed-time units="sec">0.012</elapsed-time>
  </meta>
  <relations>
    <relation-group array="true">
      <uriref>/manage/LATEST/databases</uriref>
      <typeref>databases</typeref>
      <relation-count>13</relation-count>
    </relation-group>
    <relation-group array="true">
      <uriref>/manage/LATEST/forests</uriref>
      <typeref>forests</typeref>
      <relation-count>13</relation-count>
    </relation-group>
    <relation-group array="true">
      <uriref>/manage/LATEST/groups</uriref>
      <typeref>groups</typeref>
      <relation-count>1</relation-count>
    </relation-group>

    <relation-group array="true">
      <uriref>/manage/LATEST/hosts</uriref>
      <typeref>hosts</typeref>
      <relation-count>1</relation-count>
    </relation-group>
    <relation-group array="true">
      <uriref>/manage/LATEST/requests</uriref>
      <typeref>requests</typeref>
      <relation-count>1</relation-count>
    </relation-group>
```

```
      <relation-group array="true">
        <uriref>/manage/LATEST/servers</uriref>
        <typeref>servers</typeref>
        <relation-count>8</relation-count>
      </relation-group>
      <relation-group array="true">
        <uriref>/manage/LATEST/transactions</uriref>
        <typeref>transactions</typeref>
      </relation-group>
    </relations>
    <related-views>
      <related-view array="true">
        <view-type>item</view-type>
        <view-name>query</view-name>
        <view-uri>/manage/LATEST/query</view-uri>
      </related-view>
      <related-view array="true">
        <view-type>item</view-type>
        <view-name>status</view-name>
        <view-uri>/manage/LATEST?view=status</view-uri>
      </related-view>
    </related-views>
  </cluster-default>
```

5.9 Interpreting the Output

The reference documentation for the Management API in the *MarkLogic REST API Reference* describes each element in the XML output for each Management API resource address.

The main elements in the output from each resource address for an item or item list is shown in the table below.

Type of Resource Address	Element	Description
Item and Item View	`id`	The item id number.
Item and Item View	`name`	The item name (not available for requests and transactions).
Server Item	`server-kind`	The type of App Server (http, WebDAV, or XDBC)
Item, Item List, and View	`meta`	Metadata that describes: • The URI of the resource. • The current timestamp. • The number of seconds it took to execute the resource address.
View	*`view`*`-properties`	The properties of the item or item list view.
Item and Item List	`relations`	The items that are related to this item or item list.
Item List	`list-items`	The items in this item list
Item, Item List, and View	`related-views`	The views related to this item or item list. If an item, the item list view is also included.

6.0 Technical Support

MarkLogic provides technical support according to the terms detailed in your Software License Agreement or End User License Agreement.

We invite you to visit our support website at http://help.marklogic.com to access information on known and fixed issues, knowledge base articles, and more. For licensed customers with an active maintenance contract, see the Support Handbook for instructions on registering support contacts and on working with the MarkLogic Technical Support team.

Complete product documentation, the latest product release downloads, and other useful information is available for all developers at http://developer.marklogic.com. For general questions, join the general discussion mailing list, open to all MarkLogic developers.

7.0 Copyright

MarkLogic Server 8.0 and supporting products.

NOTICE