Monitoring Elasticsearch

eG Innovations Product Documentation

www.eginnovations.com

eG
Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

Elasticsearch is an open-source, RESTful, distributed search and analytics engine built on Apache Lucene. Elasticsearch has quickly become the most popular search engine, and is commonly used for log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases.

Elasticsearch is a highly scalable open-source full-text search and analytics engine. It allows you to store, search, and analyze big volumes of data quickly and in near real time. It is generally used as the underlying engine/technology that powers applications that have complex search features and requirements. The speed and scalability of Elasticsearch and its ability to index many types of content mean that it can be used for various purposes:

- Application search
- Website search
- Enterprise search
- Logging and log analytics
- Infrastructure metrics and container monitoring
- Application performance monitoring
- Geospatial data analysis and visualization
- Security analytics
- Business analytics

Due to the high availability and efficient indexing features, the Elasticsearch cluster is very popular in large, mission-critical IT infrastructures, which require ready and reliable services at all times. In such environments, the non-availability of the Elasticsearch server or poor index and serach performance will ultimately slowdown the dependent end-user services. To avoid this, it is imperative to watch out for issues in the operations and availability of the server on a regular basis. This can be easily achieved using a specialized monitoring model offered by eG Enterprise.

Chapter 2: How to Monitor Elasticsearch Cluster Using eG Enterprise?

eG Enterprise monitors the Elasticsearch Single-node Cluster using an agent based or agentless approach. In case of the agentless approach, an eG agent used to monitor the cluster should be deployed on a remote Windows host in the environment. Regardless of the approach (agent-based or agentless), the eG agent makes RESTful API calls to connect to the cluster to pull metrics related to availability, health and search performance of the cluster. For this purpose, each test that the eG agent runs on the Elasticsearch cluster should be configured with the credentials of a user who has privileges to make RESTful API calls to the cluster.

By default, the eG agent will not require any authentication to collect metrics from the cluster. In some highly secured environments, to prevent unauthorized access to the Elasticsearch cluster, administrators secure the cluster using security plugins such as X-Pack, Shield, etc. In such environments, to collect metrics, you may have to configure the tests with the credentials of a user who has privileges to access the cluster via the security plugin.

2.1 Enabling JMX Support

To collect JVM related metrics from the Elasticsearch cluster, enable JMX support for the Elasticsearch. follow the steps discussed below;

- Stop the Elasticsearch cluster.
- Open the *elasticsearch.in.sh* file in the <ELASTIC-INSTALL-DIRECTORY>>/bin/ folder.
- Search for the *[JAVA_ OPTS="$JAVA_ OPTS - Xmx${ES_ MAX_ MEM}"]* line in the *elasticsearch.in.sh* file. Then, add the following lines after the above-mentioned line:

  ```
  JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.ssl=false"
  JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
  JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=<<PORT NUMBER>>"
  JAVA_OPTS="$JAVA_OPTS -Djava.rmi.server.hostname=<<IP address>>"
  ```

  Here,

  <<PORT NUMBER>> denotes Any port number which you wish to listen

  <<IP address>> is the IP address of the host on which the Elasticsearch is running

  For example:

```
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=4444"
JAVA_OPTS="$JAVA_OPTS -Djava.rmi.server.hostname=192.168.9.90"
```

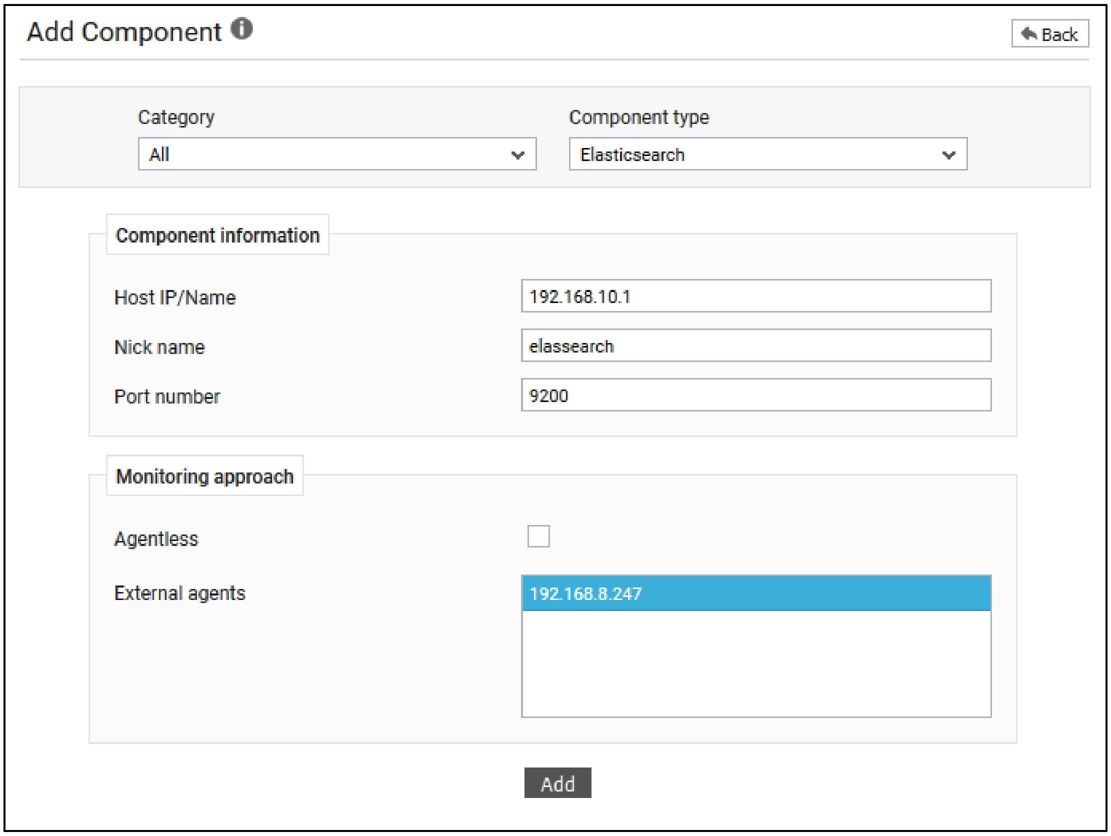
- Save the <<Elastic-Install-Directory>>/bin/elasticsearch.in.sh file
- Start the Elasticsearch server.
- Ensure that JMX is listening by executing *netstat -an | grep "LIST"* command

Once the above-said pre-requisites are set in place, manage the *Elasticsearch* component to start monitoring the target cluster. The steps for managing the *Elasticsearch* component are explained in Section **2.2**.

2.2 Managing Elasticsearch Cluster

Using eG Enterprise, you can auto-discover the Elasticsearch Cluster as well as manually add the component for monitoring. To manage an *Elasticsearch* component, do the following:

1. Log into the eG admin interface.
2. If the Elasticsearch is already discovered, then directly proceed towards managing it using the **Components – Manage/Unmanage/Delete** page.
3. However, if the target Elasticsearch cluster is yet to be discovered, then run discovery (Infrastructure -> Components -> Discovery) to get it discovered or follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu to manually add the component using the **Components** page. Remember that components manually added are managed automatically.
4. In the **Components** page that appears next, select *Elasticsearch* as the **Component type**. Then, click the **Add New Component** button. This will invoke Figure 2.1.



Category: All

Component type: Elasticsearch

Host IP/Name: 192.168.10.1

Nick name: elassearch

Port number: 9200

Agentless: □

External agents: 192.168.8.247

Figure 2.1: Adding the Elasticsearch component

5. Specify the **Host IP/Name** and the **Nick name** for the *Elasticsearch* component.

6. Next, specify the port at which the target host is listening to. By default, this is set to *9200*. If the target host is listening on a different port, then override the default setting and specify the port at which the target host is currently listening.

7. Then, choose an external agent for the target host by picking an option from the **External agents** list box.

8. Next , click the **Add** button to register the changes (see Figure 2.1).

9. Finally, signout of the eG admin interface.

Chapter 3: Monitoring Elasticsearch Server Using eG Enterprise

eG Enterprise provides a specialized Elasticsearch monitoring model that monitors the health, index health, search performance of the Elasticsearch, and promptly captures and reports abnormalities.

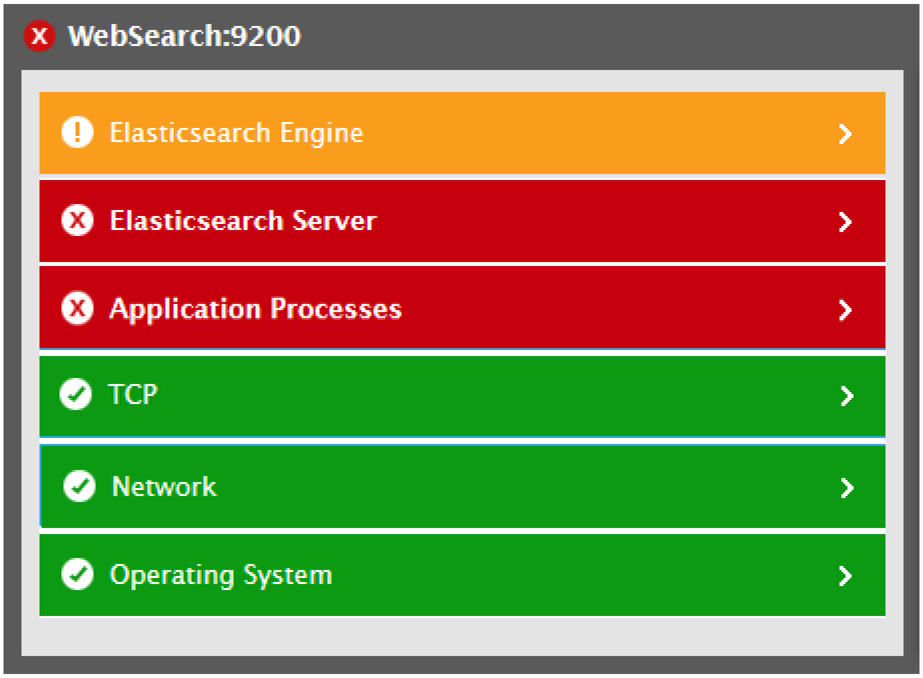


Figure 3.1: The layer model of Elasticsearch server

Every layer of the layer model above is mapped to a variety of tests to collect performance metrics of the Elasticsearch server. Analyzing the metrics reported by the tests, administrators can find out the accurate answers for the following queries:

- Is the target server connected?
- What is the current health of the Elasticsearch cluster?
- How many active primary shards are in the Elasticsearch cluster?
- How many secondary shards are created in the cluster?
- How many data nodes are in the cluster?
- What is the current health of each index on the cluster?
- What is the indexing rate of each index?

- What is the count of documents that were added to/deleted from each index?
- How many refresh/merge/flush operations are performed in each index?
- How long each index took for performing refresh/merge/flush operations?
- How many primary shards are in each index?
- How many secondary shards are created for the primary shards in each index?
- What is the count of indexes in the normal/warning/critical states?
- How many indexes are currently in relocating and initializing states?
- What is the rate at which the search queries were processed at each index?
- How many number of query evictions were performed in the query cache and fielddata cache?
- What is the size of query cache and fielddata cache in each index?
- How many threads are currently busy? Does the server appear to be handling too much load?

Since the tests mapped to the bottom 4 layers of the layer model (Figure 3.1) are elaborately dealt in the *Monitoring Unix and Windows Servers* document, the sections to come will discuss about the tests mapped to **Elasticsearch Server** and **Elasticsearch Engine** layers.

3.1 Elasticsearch Engine Layer

The tests associated with this layer help administrators to measure the availability of the Elasticsearch server and health of the Elasticsearch cluster at shards level. This way, administrators are proactively alerted when the server connectivity failures occur and health of the cluster goes down. Besides, administrators can also measure the usage of threads, count of active threads in each thread pool and the number of threads made to wait in queue.

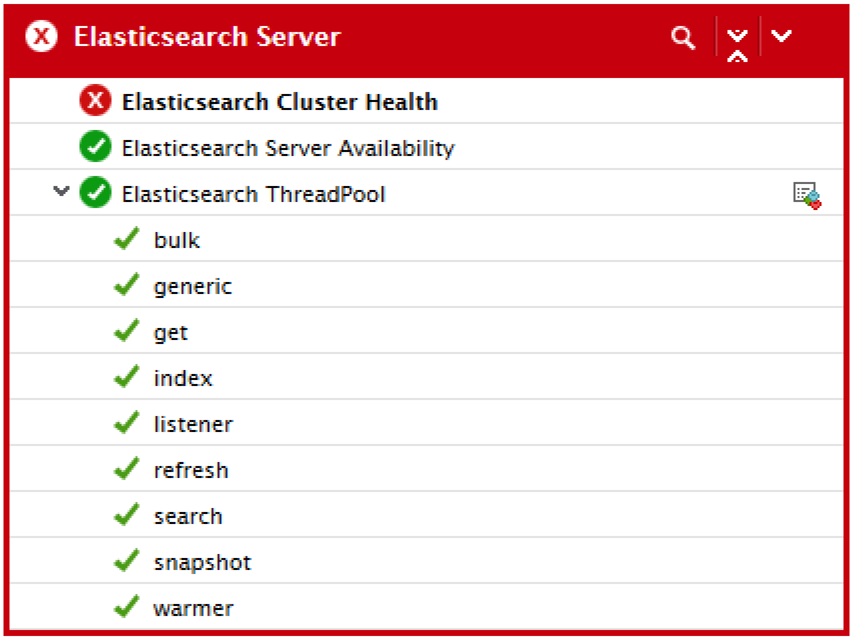


Figure 3.2: The Elasticsearch Server layer

3.1.1 Elasticsearch Cluster Health Test

An Elasticsearch cluster is a group of one or more Elasticsearch nodes that are connected together. The Elasticsearch cluster efficiently distributes the tasks, searches and indexes across all the nodes. The nodes in the Elasticsearch cluster can be assigned different jobs or responsibilities:

- **Data nodes** - stores data and executes data-related operations such as search and aggregation
- **Master nodes** - in charge of cluster management and configuration actions such as adding and removing nodes
- **Client nodes** - forwards cluster requests to the master node and data-related requests to data nodes
- **Ingest nodes** - for pre-processing documents before indexing

By default, each node is automatically assigned a unique identifier, or name, that is used for management purposes and becomes even more important in a multi- node, or clustered, environment. To add and efficiently manage a large amount of data in the cluster, Elasticsearch enables creating indexes in the cluster. An index is a collection of documents with similar characteristics, and is identified by a name. The index name is used to refer to the particular index while performing indexing, search, update, and delete operations against the documents in the cluster. The index can potentially store a large amount of data that can exceed the hardware limits of a single node. For example, a single index of a billion documents taking up 1TB of disk space may not fit on the disk of a single node or may be too slow to serve search requests from a single node alone. To solve this problem, Elasticsearch provides the ability to subdivide the index into multiple pieces called shards. When you create an index, you can simply define the number of shards you want. Each shard is a fully-functional and independent index that can be allocated to any node in the cluster. Furthermore, Elasticsearch allows you to create one or more copies of the shards called replica shards or replicas to provide high availability in case a primary shard/node goes offline or fails or becomes unavailable for any reason. Using the shards, administrators can horizontally split/scale content volume and distribute and parallelize operations across the nodes. If any of the shards is in the unassigned/relocating state for longer duration, the search queries to that particular shard will be queued or left unserviced permanently. If the issue is persisted, the incoming search queries will not be processed quickly as they have to be. This in turn will lead to processing bottleneck in the cluster which adversely impact the performance of the cluster. To avoid this, administrator should continuously monitor the health of the cluster at shards level. This can be easily achieved using the **Elasticsearch Cluster Health** test!

This test continuously monitors the cluster, and the health of the cluster at regular intervals. In addition, this test also reports the count of active shards and the number of shards in the unassigned. These revelations help administrators to track the health of the cluster continuously.

Target of the test : An Elasticsearch Cluster

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the target Elasticsearch cluster being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Port	The port number at which the target host being monitored listens. By default, this is set to 9200.
Cloud Instance URL	By default, the is set to *none*. If the target Elasticsearch cluster is hosted on the cloud environment, then you need to specify the URL of the cluster on cloud against this parameter. For example: *64bd966328067fd89e0c9b4c3bb8b042.us-east-1.aws.found.io*. When the cloud URL is specified, the eG agent will use the cloud URL to monitor the target cluster rather than using the host specified in the Host text box.
Elastic Search User	By default, the Elastic Search User and Elastic Search Password parameters are set

Parameter	Description
and Elastic Search Password	to *none* indicating that the eG agent doesn't require authentication to collect metrics from the Elasticsearch cluster. If authentication is required to access the target Elasticsearch cluster, then specify the valid credentials against these parameters.
Confirm Password	Confirm the Elastic Search Password by retyping it in the Confirm Password text box.
SSL	By default, the SSL flag is set to **No**. If the Elasticsearch cluster is SSL-enabled by default or hosted on the cluster, then set this flag to **Yes**. This indicates that the eG agent will communicate with the target cluster via HTTPS by default.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Cluster health</td><td>Indicates the health of the cluster in terms of the current state of the shards in the cluster.</td><td></td><td>This measure reveals the health of the cluster at shard level. The numeric values that correspond to the measure values mentioned above are as follows:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th><th>Description</th></tr>
<tr><td>Red</td><td>0</td><td>Indicates that the specific shard is not allocated to any node in the cluster.</td></tr>
<tr><td>Yellow</td><td>1</td><td>Indicates that the primary shard is allocated but replicas are not assigned to any node.</td></tr>
<tr><td>Green</td><td>2</td><td>Indicates that all shards in the cluster are allocated to the nodes.</td></tr>
</table>
<b>Note:</b>
This test typically reports the <b>Measure Value</b>s listed in the table above to indicate the current health of the cluster. However, the graph of this measure is</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			represented using the numeric equivalents only.
Active primary shards	Indicates the number of primary shards that are currently active on the cluster.	Number	
Active shards	Indicates the total number of shards that are currently active on the cluster.	Number	
Unassigned shards	Indicates the number of shards that are in the UNASSIGNED state.	Number	Ideally, zero is desired for this measure. A non-zero value indicates that one or many shards are yet to be allocated to the nodes, which may cause imbalance in the cluster and make the cluster unreliable when the nodes crash. To avoid this, administrators may have to allocate the unassigned shards to the various nodes on the cluster or delete the shards if the data in the shards is not needed anymore.
Initializing shards	Indicates the number of shards that are currently in the INITIALIZING state.	Number	
Relocating Shards	Indicates the number of shards that are being moved from one node to another node in the cluster.	Number	Typically, administrators move the shards from one node to another node to maintain cluster's health when a new node is added to the cluster or many shards are idle or in unassinged state.
Data nodes	Indicates the number of data nodes in the cluster.	Number	
Total nodes	Indicates the total number of nodes in the cluster.	Number	

3.1.2 Elasticsearch Server Availability Test

An Elasticsearch cluster is a group of one or more Elasticsearch nodes that are connected together. Each node is a server (either virtual or physical) that stores cluster data and participates in the indexing and searching capabilities of the cluster. This means that the node will participate in a given search query by searching the data that it stores. Therefore, the availability and active participation of the node is important for the fast search performance in the target cluster. If a node fails to connect with the cluster, then such a node cannot service search requests. This in turn adversely impacts the user experience with the cluster. In order to prevent this, administrators should monitor the availability of the node round the clock. This is where the **Elasticsearch Server Availability** test comes in handy!

This test periodically monitors the target Elasticsearch cluster and reports whether/not the server is connected with the cluster. This way, administrators can promptly detect the connection issues, initiate investigations to find out the reason behind the issues, and fix them even before the users notice and complain.

Target of the test : An Elasticsearch Cluster

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the target Elasticsearch cluster being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Port	The port number at which the target host being monitored listens. By default, this is set to 9200.
Cloud Instance URL	By default, this set to *none*. This is because, by default, the eG agent will connect to the cluster IP address provided against Host, to collect performance metrics. If the target Elasticsearch cluster is hosted on a cloud, then you need to specify the URL of the cloud instance that hosts the cluster, against this parameter. For example: *64bd966328067fd89e0c9b4c3bb8b042.us-east-1.aws.found.io*. When the cloud instance URL is specified, the eG agent will automatically connect to the specified cloud instance to pull performance metrics.
Elastic Search User	By default, the Elastic Search User and Elastic Search Password parameters are set

Parameter	Description
and Elastic Search Password	to *none*. In some high security environments, eG agent will not be able to communicate with the target server and collect metrics without valid credentials and If authentication is required to access the target Elasticsearch cluster, then specify the valid credentials against these parameters.
Confirm Password	Confirm the Elastic Search Password by retyping it in the Confirm Password text box.
SSL	By default, the SSL flag is set to **No**. If the Elasticsearch cluster is SSL-enabled, then set this flag to **Yes**.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Elasticsearch server connectivity</td><td>Indicates whether the server is connected successfully or not.</td><td></td><td>The numeric values that correspond to the measure values mentioned above are as follows:
<table>
<tr><th>Measure Value</th><th>Numeric Value</th></tr>
<tr><td>Success</td><td>1</td></tr>
<tr><td>Failure</td><td>0</td></tr>
</table>
<b>Note:</b>
This test typically reports the <b>Measure Values</b> listed in the table above to indicate the current connectivity status of the target server. However, the graph of this measure is represented using the numeric equivalents only.</td></tr>
</table>

3.1.3 Elasticsearch ThreadPool Test

Thread pools are a collection of threads which are made available to perform specific tasks in the Elasticsearch cluster. A single Elasticsearch node holds multiple thread pools for different operations such as search, indexing, bulk operations, and more. The main goal of the thread pools is to make the memory management easier by managing the life-cycles of the threads while executing large number of requests. The thread pools are associated with queues to deal with the pending requests. When no free threads are available in the thread pool, the thread pool buffers the requests to the associated queue. Once the threads become free, the pending requests in the queue will be processed. This means that if enough free threads are not available in a pool, request processing will slowdown. Moreover, if the associated queues are also full, incoming requests will be rejected and errors will be thrown. As a result, search experience of users will be adversely impacted. To proactively detect and eliminate processing bottlenecks and to assure users of a satisfactory search experience, administrators should keep an eye on thread usage within pools. This is where the **Elasticsearch ThreadPool** test helps!

This test auto-discovers the thread pools in the Elasticsearch cluster, and sheds light on the current load and usage of threads in each thread pool. In the process, administrators can figure out whether the pools are sized commensurate to the load. Improper sizing of pools is thus brought to light. The test also reports the number of threads buffered to the queues. This enables administrators to quickly capture a sudden/consistent increase in queue size, which in turn may impact request processing.

Target of the test : An Elasticsearch Cluster

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each thread pool in the target Elasticsearch cluster.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Port	The port number at which the target host being monitored listens. By default, this is set to 9200.
Cloud Instance URL	By default, the is set to *none*. If the target Elasticsearch cluster is hosted on the cloud environment, then you need to specify the URL of the cluster on cloud against this parameter. For example: *64bd966328067fd89e0c9b4c3bb8b042.us-east-1.aws.found.io*. When the cloud URL is specified, the eG agent will use the cloud URL to monitor the target cluster rather than using the host specified in the Host text box.
Elastic Search User and Elastic Search Password	By default, the Elastic Search User and Elastic Search Password parameters are set to *none* indicating that the eG agent doesn't require authentication to collect metrics from the Elasticsearch cluster. If authentication is required to access the target Elasticsearch cluster, then specify the valid credentials against these parameters.

Parameter	Description
Confirm Password	Confirm the Elastic Search Password by retyping it in the Confirm Password text box.
SSL	By default, the SSL flag is set to **No**. If the Elasticsearch cluster is SSL-enabled by default or hosted on the cluster, then set this flag to **Yes**. This indicates that the eG agent will communicate with the target cluster via HTTPS by default.

Measurements made by the test

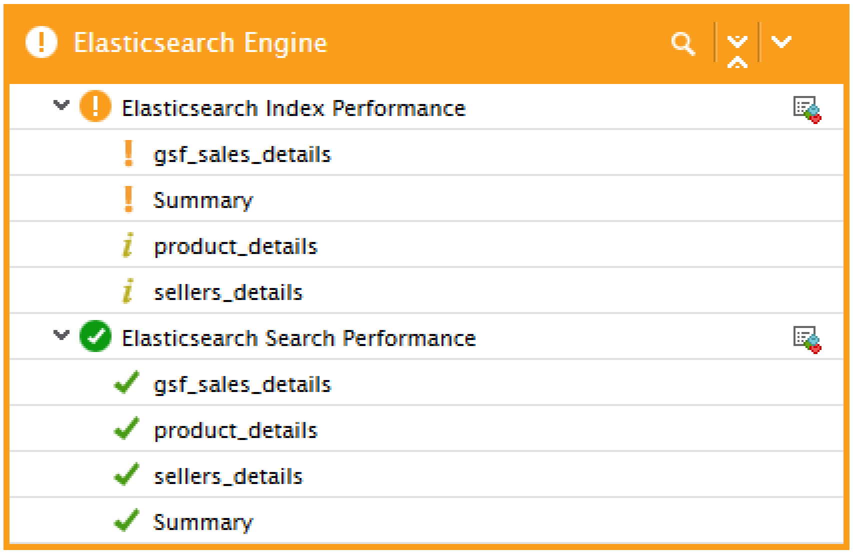
Measurement	Description	Measurement Unit	Interpretation
Thread utilization	Indicates the percentage of threads in this thread pool that are currently in use.	Percent	This measure is computed using the following formula: *Thread utilization=(Active threads / Total threads)X100* A high value is indicative of a busy pool. A value close to 100% indicates excessive utilization of threads in a pool. If the value of this measure grows closer to 100% over time, it indicates that the pool is rapidly running out of threads to service the search requests. In such a case, administrators may have to fine-tune the thread pool size.
Active threads	Indicates the number of threads in this pool that are currently active in servicing the requests.	Number	This measure serves as a good indicator of the workload of the thread pool. Comparing the value of this measure across the thread pools helps administrators identify the thread pool that is processing the maximum number of search requests.
Queued threads	Indicates the number of threads queued for processing in this pool.	Number	A low value is desired for this measure. A high value or a sudden increase in this value may indicate that the thread pool is unable to service requests as they come in. This

Measurement	Description	Measurement Unit	Interpretation
			can cause request processing delays.
Total threads	Indicates the maximum number of active threads that this pool can contain.	Number	
Rejected threads	Indicates the number of threads that were rejected from this pool.	Number	Ideally, the value of this measure should be 0. If a non-zero value is reported, it implies that one/more requests were rejected owing to the lack of free threads in the pool. Administrators may then have to increase the size of the thread pool or queue in order to avoid rejections/errors.

3.2 Elasticsearch Engine Layer

The tests mapped to this layer report the following performance metrics for each index on the Elasticsearch cluster:

- Current health;
- Number of refresh/flush/merge operations performed;
- Average time taken for performing the refresh/flush/merge operations;
- Total number of primary shards, the number of active primary shards and the number of secondary shards created for each primary shard in each index;
- Average time taken for processing the search requests;
- Number of evictions performed in query cache and fielddata cache while processing the search queries;



Elasticsearch Engine

- ! Elasticsearch Index Performance
  - ! gsf_sales_details
  - ! Summary
  - i product_details
  - i sellers_details
- ✓ Elasticsearch Search Performance
  - ✓ gsf_sales_details
  - ✓ product_details
  - ✓ sellers_details
  - ✓ Summary

Figure 3.3: The Elasticsearch Engine layer

3.2.1 Elasticsearch Index Performance Test

An index is a collection of documents that have somewhat similar characteristics. For example, you can have an index for customer data, another index for a product catalog, and yet another index for order data. An index is identified by a name (that must be all lowercase) and this name is used to refer to the index when performing indexing, search, update, and delete operations against the documents in it. In a single Elasticsearch cluster, administrators can define multiple indexes. A healthy index ensures that search queries are serviced quickly. If the index becomes unhealthy or if refresh/flush operations become infrequent/slow, search efficiency will be adversely impacted. As a result, performance of the Elasticsearch cluster will be degraded and the user experience will be affected. To avoid such anomalies, administrators should track the health of indexes and index-related operations (eg., refresh, flush, merge) at regular intervals. To achieve this, administrators can use the **Elasticsearch Index Performance** test.

This test auto-discovers the indexes in the Elasticsearch cluster, and reports the current health and change in the size of each index. In addition, this test also reveals the count of refresh/merge/flush operations performed in each index and the time taken for performing these operations. With the help of these metrics, administrators can proactively detect potential slowness in index-related operations, and promptly initiate preventive measures. Besides, this test sheds light on the number of documents added to/deleted from each index, using which the administrators can easily track changes to index size. The count of shards in different states is reported, and in the process, unassigned shards (if any) are also captured.

Target of the test : An Elasticsearch Cluster

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each index in the target Elasticsearch cluster.

This test also reports metrics for the **Summary** descriptor.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Port	The port number at which the target host being monitored listens. By default, this is set to *9200*.
Cloud Instance URL	By default, the is set to *none*. If the target Elasticsearch cluster is hosted on the cloud environment, then you need to specify the URL of the cluster on cloud against this parameter. For example: *64bd966328067fd89e0c9b4c3bb8b042.us-east-1.aws.found.io*. When the cloud URL is specified, the eG agent will use the cloud URL to monitor the target cluster rather than using the host specified in the Host text box.
Elastic Search User and Elastic Search Password	By default, the Elastic Search User and Elastic Search Password parameters are set to *none* indicating that the eG agent doesn't require authentication to collect metrics from the Elasticsearch cluster. If authentication is required to access the target Elasticsearch cluster, then specify the valid credentials against these parameters.
Confirm Password	Confirm the Elastic Search Password by retyping it in the Confirm Password text box.
SSL	By default, the SSL flag is set to **No**. If the Elasticsearch cluster is SSL-enabled by default or hosted on the cluster, then set this flag to **Yes**. This indicates that the eG agent will communicate with the target cluster via HTTPS by default.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Index Health	Indicates the current health of this index.		This measure reveals the health of the index at shard level. The numeric values that correspond to the measure values mentioned above are as follows:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure Value</th><th>Numeric Value</th><th>Description</th></tr>
<tr><td>Red</td><td>0</td><td>Indicates that the specific shard is not allocated to any node in the cluster.</td></tr>
<tr><td>Yellow</td><td>1</td><td>Indicates that the primary shard is allocated but replicas are not assigned to any node.</td></tr>
<tr><td>Green</td><td>2</td><td>Indicates that all shards in the cluster are allocated to the nodes.</td></tr>
</table>
<b>Note:</b>
This test typically reports the <b>Measure Value</b>s listed in the table above to indicate the current health of each index. However, the graph of this measure is represented using the numeric equivalents only.
This measure is not applicable for <b>Summary</b> descriptor.
</td></tr>
<tr><td>Indexing Rate</td><td>Indicates the rate at which the documents were indexed in this index.
For <b>Summary</b> descriptor, this measure indicates the rate at which the documents were indexed in the indexes of the cluster.</td><td>Documents/sec</td><td>A gradual/sudden increase in the value of this measure indicates that the index size growing as well. In such cases, administrators may need to allocate more space to the index.</td></tr>
<tr><td>Recently added documents</td><td>Indicates the number of documents that were newly added to this</td><td>Number</td><td></td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
	index during last measurement period. For **Summary** descriptor, this measure indicates the number of documents that were newly added to the indexes of the cluster.		
Total indexed documents	Indicates the total number of documents that have been added to this index since this index was created. For **Summary** descriptor, this measure indicates the number of documents that are added to the indexes of the cluster.	Number	
Recently deleted documents	Indicates the number of documents that were deleted from this index during last measurement period. For **Summary** descriptor, this measure indicates the number of documents that were deleted from the indexes.		
Total deleted documents	Indicates the total number of documents that are deleted from this index since the index was created. For **Summary**	Number	

Measurement	Description	Measurement Unit	Interpretation
	descriptor, this measure indicates the total number of documents that are deleted from the indexes.		
Index refreshes	Indicates the number of refresh operations performed for this index. For **Summary** descriptor, this measure indicates the total number of refresh operations performed for the indexes in the cluster.	Number	An Elasticsearch refresh makes the documents available for search whenever new documents are added to the index. The refresh operation is performed at regular intervals. A very low value of this measure denotes that the newly added documents may not be available for the search.
Time spent on index refreshes	Indicates the time taken for performing refresh operations for this index. For **Summary** descriptor, this measure indicates the time taken for performing refresh operations in the cluster.	Seconds	Compare the value of this measure across the indexes to identify the index that took maximum time to refresh.
Average time spent per refresh	Indicates the average time taken for performing a single refresh operation for this index. For **Summary** descriptor, this measure indicates the time taken for performing a single refresh operation in the cluster.	Seconds	A low value is desired for this measure.
Index merges	Indicates the number of merge operations performed in this index.	Number	A shard in the index is broken down into segments. Segments are internal storage elements in the index where the index

Measurement	Description	Measurement Unit	Interpretation
	For **Summary** descriptor, this measure indicates the number of merge operations performed in the cluster.		data is stored, and are immutable. Smaller segments are periodically merged into larger segments to keep the index size at bay and to expunge deletes.
Time spent on index merges	Indicates the average time taken for performing the merge operations in this index. For **Summary** descriptor, this measure indicates the average time taken for performing the merge operations in the cluster.	Seconds	Comparing the value of this measure across the indexes will reveal the index that took maximum time to perform merge operations.
Average time spent per merge	Indicates the time taken for performing a single merge operation in this index. For **Summary** descriptor, this measure indicates the time taken for performing a single merge operation in the cluster	Seconds	A low value is desired for this measure.
Index flushes	Indicates the number of flush operations performed in this index. For **Summary** descriptor, this measure indicates the number of flush operations performed in the cluster.	Number	The flush process is performed to ensure that any data that is currently only persisted in the transaction log is also permanently persisted in Lucene search library. This reduces recovery times as that data does not need to be reindexed from the transaction logs after the Lucene index is opened. By default, Elasticsearch uses heuristics in order to automatically trigger flushes as required. A very low value of this measure may indicate the increase in recovery time of

Measurement	Description	Measurement Unit	Interpretation
			the index.
Time spent on index flushes	Indicates the total time taken to perform flush operations in this index. For **Summary** descriptor, this measure indicates the time taken for performing a single flush operation in the cluster.	Seconds	
Average time spent per flush	Indicates the time taken to perform a single flush operation in this index. For **Summary** descriptor, this measure indicates the time taken for performing a single flush operation in the cluster.	Seconds	
Primary shards	Indicates the number of primary shards in this index.	Number	This measure will not be reported for the **Summary** descriptor.
Replica shards per primary shard	Indicates the number of replica shards created for each primary shard in this index.	Number	This measure will not be reported for the **Summary** descriptor.
Active primary shards	Indicates the number of primary that are currently active in this index.	Number	This measure will not be reported for the **Summary** descriptor.
Total active shards	Indicates the total number of active shards in this index.	Number	This measure will not be reported for the **Summary** descriptor.
Relocating shards	Indicates the number of relocating shards in this index.	Number	This measure will not be reported for the **Summary** descriptor.

Measurement	Description	Measurement Unit	Interpretation
Initializing shards	Indicates the number of shards that are currently in initializing state in this index.	Number	This measure will not be reported for the **Summary** descriptor.
Unassigned shards	Indicates the number of unassigned shards in this index.	Number	This measure will not be reported for the **Summary** descriptor.
Indices in normal state	Indicates the number of indexes in the normal state.	Number	This measure is only applicable for the **Summary** descriptor. The detailed diagnosis of this measure reveals the name of the indexes that are in the normal state.
Indices in warning state	Indicates the number of indexes in the warning state.	Number	This measure is only applicable for the **Summary** descriptor. The detailed diagnosis of this measure reveals the name of the indexes that are in the warning state.
Indices in critical state	Indicates the number of indexes in the critical state.	Number	This measure is only applicable for the **Summary** descriptor. The value of this measure should be very low. The detailed diagnosis of this measure reveals the name of the indexes that are in the critical state.

3.2.2 Elasticsearch Search Performance Test

This test auto-discovers the indexes in the Elasticsearch server, and for each index, reports the average time taken to process the search queries and the rate at which queries are processed. This way, administrators can easily identify the index that is currently experiencing processing bottlenecks. Additionally, the test also measures the number of evictions performed in the query cache and the field data cache and the size of these caches. These metrics alert administrators to inconsistencies in the sizing of the caches in each index.

Target of the test : An Elasticsearch Cluster

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each index in the Elasticsearch cluster being monitored.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Port	The port number at which the target host being monitored listens. By default, this is set to 9200.
Cloud Instance URL	By default, the is set to *none*. If the target Elasticsearch cluster is hosted on the cloud environment, then you need to specify the URL of the cluster on cloud against this parameter. For example: *64bd966328067fd89e0c9b4c3bb8b042.us-east-1.aws.found.io*. When the cloud URL is specified, the eG agent will use the cloud URL to monitor the target cluster rather than using the host specified in the Host text box.
Elastic Search User and Elastic Search Password	By default, the Elastic Search User and Elastic Search Password parameters are set to *none* indicating that the eG agent doesn't require authentication to collect metrics from the Elasticsearch cluster. If authentication is required to access the target Elasticsearch cluster, then specify the valid credentials against these parameters.
Confirm Password	Confirm the Elastic Search Password by retyping it in the Confirm Password text box.
SSL	By default, the SSL flag is set to **No**. If the Elasticsearch cluster is SSL-enabled by default or hosted on the cluster, then set this flag to **Yes**. This indicates that the eG agent will communicate with the target cluster via HTTPS by default.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Average query latency	Indicates the average time taken for processing the search requests in this index.	Milliseconds	This value is calculated using the following formula: *Query latency=Total time taken for processing all queries/Total number of queries received* The value of this measure is desired to be low. A high value indicates that the index is taking too long to process the

Measurement	Description	Measurement Unit	Interpretation
			queries. If this condition is allowed to persist, then it may degrade user experience with the Elasticsearch cluster.
Query rate	Indicates the rate at which the search queries were processed in this index.	Queries/sec	This measure helps administrator to know how well/badly the search requests were processed in each index. A very high value for this measure indicates that the index is imposed with many search requests.
Query cache evictions	Indicates the number of cache evictions performed while processing the search queries in this index.	Number	The query cache is responsible for caching the results of queries. When the cache is full, the less-frequently used data is evicted to make way for new data. A high value of this measure could indicate that the index is sized with inadequate query cache. You may want to increase the size of the query cache, to reduce evictions and improve overall search performance.
Query cache size	Indicates the current size of the query cache of this index.	MB	
Fielddata cache evictions	Indicates the number of evictions performed in the field data cache while processing the search queries in this index.	Number	The field data cache is used mainly when sorting on or computing aggregations on a field, and loads all the field values to memory in order to provide fast document based access to computed values. Compare the value of this measure across the indexes to identify the index in which the maximum number of evictions were performed in the field data cache.
Fielddata cache size	Indicates the current size of the filed data cache of this index.	MB	

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.