title: SolrCloud Monitoring Integration description: Our monitoring and logging platform includes integration for SolrCloud. Use predefined key metrics reports combined with rich data visualization tools to monitor critical issues on your Solr machines cluster, and receive alerts on memory usage, uptime, load averages, index stats, document and filter caches, latency, rate, and more

Sematext offers simple and versatile SolrCloud monitoring agent written in Java and Golang with minimal CPU and memory overhead. It's easy to install and require no changes in the SolrCloud source code or your application's source code.

Sematext SolrCloud Monitoring Agent

This lightweight, open-source Monitoring Agent collects SolrCloud performance metrics and sends them to Sematext. It comes packaged with a Golang-based agent responsible for Operating System level metrics like network, disk I/O, and more. The SolrCloud Monitoring Agent can be installed with RPM/DEB package manager on any host running Linux or in a containerized environment using `sematext/sematext-agent`.

The Sematext SolrCloud Monitoring Agent can be run in two different modes - *in-process* and *standalone*. The *in-process* one is run as a Java agent, it is simpler to initially set up, but will require restarting your Solr node when you will want to upgrade your monitoring Agent, i.e. to get new features. The benefit of the *standalone* agent mode is that it runs as a separate process and doesn't require a Solr restart when it is installed or upgraded.

After creating a SolrCloud App in Sematext you need to install the Monitoring Agent on each host running your SolrCloud nodes to have the full visibility over the metrics from each host. The full installation instructions can be found in the setup instructions displayed in the UI.

For example, on CentOS, you need to add Sematext Linux packages and install them with the following command:

```
sudo wget https://pub-repo.sematext.com/centos/sematext.repo -O /etc/yum.repos.d/sematext.re
sudo yum clean all
sudo yum install sematext-agent
```

After that, setup the SolrCloud Monitoring Agent by running a command like this:

```
sudo bash /opt/spm/bin/setup-sematext  \
    --monitoring-token <your-monitoring-token-goes-here>   \
    --app-type solrcloud  \
    --agent-type javaagent  \
    --infra-token <your-infra-token-goes-here>
```

The command above will set up your SolrCloud Monitoring Agent in the *in-process* mode. To have it running in the *standalone* mode, run the command

below instead of the one above:

```
sudo bash /opt/spm/bin/setup-sematext  \
    --monitoring-token <your-monitoring-token-goes-here>   \
    --app-type solr  \
    --agent-type standalone  \
    --infra-token <your-infra-token-goes-here>  \
    --jmx-params '-Dspm.remote.jmx.url=localhost:3000'
```

Keep in mind that your need to provide the Monitoring token and Infra token. They are both provided in the installation instructions for your SolrCloud App.

Finally, the last thing that needs to be done is adjusting the *solr.in.sh* file and add the following section:

```
SOLR_OPTS="$SOLR_OPTS -Dcom.sun.management.jmxremote -javaagent:/opt/spm/spm-monitor/lib/spm
```

Or if you would like to run the Solr Monitoring Agent in the *standalone* mode add the following section to the `solr.in.sh` file:

```
SOLR_OPTS="$SOLR_OPTS -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=300
```

Make sure that tag `<jmx />` is enabled in your `solrconfig.xml` file.

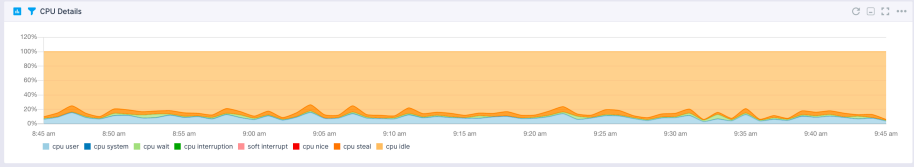
You need to restart your SolrCloud node after the changes above.

Collected Metrics

The Sematext SolrCloud monitoring agent collects the following metrics.

Operating System

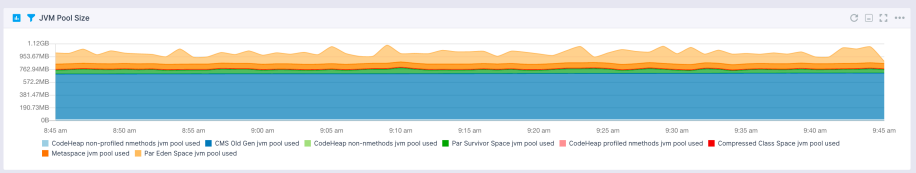
- CPU usage
- CPU load
- Memory usage
- Swap usage
- Disk space used
- I/O Reads and Writes
- Network traffic



Java Virtual Machine

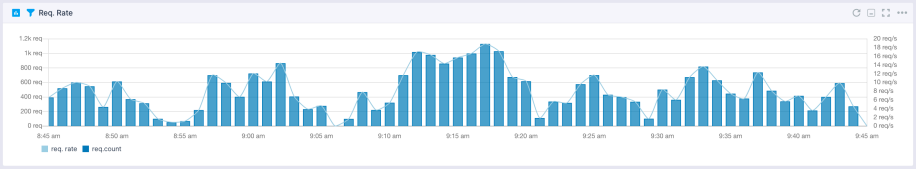
- Garbage collectors time and count
- JVM pool size and utilization

- Threads and daemon threads
- Files opened by the JVM



Solr

- Requests rate and latency
- Solr index stats and file system stats
- Added and pending documents
- Deletes by id and queries
- Filter cache statistics
- Document cache statistics
- Query result cache statistics
- Per segment filter cache statistics
- Commit events
- Warmup times



Troubleshooting

If you are having issues with Sematext Monitoring, i.e. not seeing SolrCloud metrics, see How do I create the diagnostics package.

For more troubleshooting information please look at Troubleshooting section.

Integration

- Agent: https://github.com/sematext/sematext-agent-java
- Tutorial: https://sematext.com/blog/solr-monitoring-made-easy-with-sematext/
- Instructions: https://apps.sematext.com/ui/howto/Solr/overview

Metrics

Metric Name Key *(Type) (Unit)*	Description
cache lookups**solr.cache.lookups** *(long counter)*	lookups count
cache hits**solr.cache.hits** *(long counter)*	hits count
cache size**solr.cache.size** *(long gauge)*	cache size (count of elements)
cache evictions**solr.cache.evicted** *(long counter)*	count of evictions
warmup time**solr.warmup.time** *(long counter) (ms)*	warmup time
cache memory used**solr.cache.size.bytes** *(long gauge) (bytes)*	cache size in bytes
cache max size**solr.cache.size.max** *(long gauge)*	cache max size
autowarm count or %**solr.cache.autowarm.count** *(long gauge)*	cache autowarm count or %
request time**solr.requests.time** *(long counter) (ms)*	request time
req.count**solr.requests** *(long counter) (req)*	request count
avg. request latency**solr.requests.latency.avg** *(double gauge)*	avg. request latency
request error count**solr.requests.error.count** *(long counter)*	request error count
request timeout count**solr.requests.timeout.count** *(long counter)*	request timeout count
index max doc**solr.index.docs.max** *(long gauge) (docs)*	max doc in the index
index num docs**solr.index.docs** *(long gauge) (docs)*	number of docs in the index
index segments**solr.index.segments** *(long gauge)*	index segments count
index num of files**solr.index.files** *(long gauge)*	number of files in solr index
index size on the disk**solr.index.files.size** *(long gauge) (bytes)*	size of solr index on the disk
commits**solr.indexing.commits** *(long counter)*	total count of commits
optimizes**solr.indexing.optimizes** *(long counter)*	count of optimizes
rollbacks**solr.indexing.rollbacks** *(long counter)*	count of rollbacks
expunge deletes**solr.indexing.deletes.expunge** *(long counter)*	count of expunge deletes
index docs added**solr.indexing.docs.added** *(long counter) (docs)*	added docs
deletes by id**solr.indexing.deletes.id** *(long counter)*	deletes by id
deletes by query**solr.indexing.deletes.query** *(long counter)*	deletes by query
update errors**solr.indexing.errors** *(long counter)*	count of update errors
autocommits**solr.indexing.commits.auto** *(long counter)*	count of auto commits

Metric Name Key *(Type) (Unit)*	Description
soft autocommits**solr.indexing.commits.soft** *(long counter)*	count of soft auto commits
index docs pending**solr.indexing.docs.pending** *(long gauge) (docs)*	count of pending docs
autocommit max time**solr.indexing.commits.auto.time.max** *(long gauge) (ms)*	autocommit max time

FAQ

** How do I enable JMX in Solr? **

Add or uncomment the **<jmx/>** directive in `solrconfig.xml` and restart Solr. See https://wiki.apache.org/solr/SolrJmx for more info.

** I don't see any data on Solr and JVM reports, what is the problem? **

You should probably enable JMX in your Solr. Add or uncomment the **<jmx />** directive in `solrconfig.xml` and restart Solr. See https://wiki.apache.org/solr/SolrJmx for more info.

** I don't see any data only in Solr Components or Errors reports, what should I do? **

Most likely you are using the standalone variant of Solr monitor. In that case, App Agent can't collect metrics which are available only when running in-process. If so, switch to in-process (javaagent) version of App Agent.

** Why don't I see Solr index file size metric? **

App Agent runs as 'spmmon' user. Make sure this user has appropriate access permissions on Solr's index directories. If you cannot give 'spmmon' user the read permissions you can switch to the in-process (javaagent) version of App Agent.