

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

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

## Sematext Solr Monitoring Agent

This lightweight, open-source Monitoring Agent collects Solr 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 Solr 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 Solr App in Sematext you need to install the Monitoring Agent on each host running Solr that you want to monitor. The full installation instructions can be found in the setup instructions displayed in the UI.

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

```
echo "deb http://pub-repo.sematext.com/ubuntu sematext main" | sudo tee
/etc/apt/sources.list.d/sematext.list > /dev/null
wget -O - https://pub-repo.sematext.com/ubuntu/sematext.gpg.key | sudo apt-key add -
sudo apt-get update
sudo apt-get install sematext-agent
```

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

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

The command above will set up your Solr 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 you need to provide the Monitoring token and Infra token. They are both provided in the installation instructions for your Solr 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-monitor-generic.jar=<your-monitoring-token-goes-here>
```

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=3000
```

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

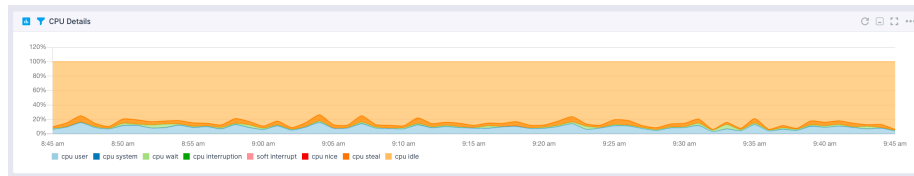
**You need to restart your Solr node after the changes above.**

## Collected Metrics

The Sematext Solr monitoring agent collects the following metrics.

### Operating System

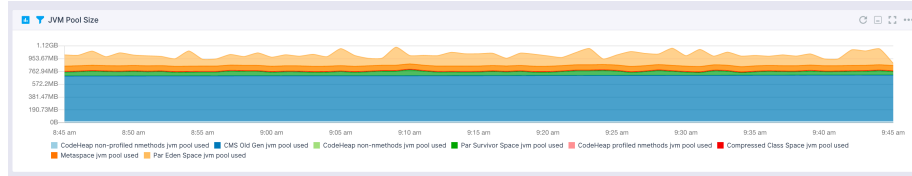
- CPU usage
- CPU load
- Memory usage
- Swap usage
- Disk space used
- I/O Reads and Writes
- Network traffic



### Java Virtual Machine

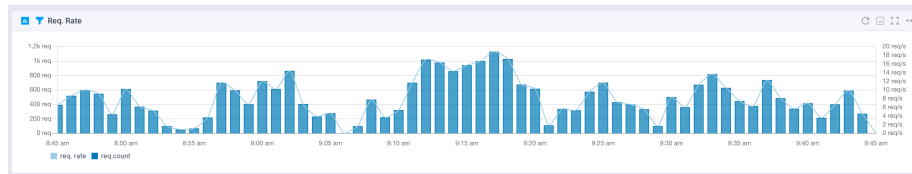
- Garbage collectors time and count
- JVM pool size and utilization
- Threads and daemon threads

- 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 lookupssolr.cache.lookups	(long counter)	lookups count
cache hitssolr.cache.hits	(long counter)	hits count

Metric Name Key <i>(Type) (Unit)</i>	Description
cache sizesolr.cache.size <i>(long gauge)</i>	cache size (count of elements)
cache evictionssolr.cache.evicted <i>(long counter)</i>	count of evictions
warmup timesolr.warmup.time <i>(long counter) (ms)</i>	warmup time
cache memory usedsolr.cache.size.bytes <i>(long gauge) (bytes)</i>	cache size in bytes
cache max sizesolr.cache.size.max <i>(long gauge)</i>	cache max size
autowarm count or %solr.cache.autowarm.count <i>(long gauge)</i>	cache autowarm count or %
request timesolr.requests.time <i>(long counter) (ms)</i>	request time
req.countsolr.requests <i>(long counter) (req)</i>	request count
avg. request latencysolr.requests.latency.avg <i>(double gauge)</i>	avg. request latency
request error countsolr.requests.error.count <i>(long counter)</i>	request error count
request timeout countsolr.requests.timeout.count <i>(long counter)</i>	request timeout count
index max docsolr.index.docs.max <i>(long gauge) (docs)</i>	max doc in the index
index num docssolr.index.docs <i>(long gauge) (docs)</i>	number of docs in the index
index segmentssolr.index.segments <i>(long gauge)</i>	index segments count
index num of filessolr.index.files <i>(long gauge)</i>	number of files in solr index
index size on the disksolr.index.files.size <i>(long gauge) (bytes)</i>	size of solr index on the disk
commitssolr.indexing.commits <i>(long counter)</i>	total count of commits
optimizessolr.indexing.optimizes <i>(long counter)</i>	count of optimizes
rollbackssolr.indexing.rollbackss <i>(long counter)</i>	count of rollbacks
expunge deletessolr.indexing.deletes.expunge <i>(long counter)</i>	count of expunge deletes
index docs addedsolr.indexing.docs.added <i>(long counter) (docs)</i>	added docs
deletes by idsolr.indexing.deletes.id <i>(long counter)</i>	deletes by id
deletes by querysolr.indexing.deletes.query <i>(long counter)</i>	deletes by query
update errorssolr.indexing.errors <i>(long counter)</i>	count of update errors
autocommitssolr.indexing.commits.auto <i>(long counter)</i>	count of auto commits
soft autocommitssolr.indexing.commits.soft <i>(long counter)</i>	count of soft auto commits

Metric Name Key <i>(Type) (Unit)</i>	Description
index docs pending <i>(long gauge) (docs)</i>	count of pending docs
autocommit max	autocommit max
timesolr.indexing.commits.auto.time.max <i>(long gauge) (ms)</i>	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 App Agent. 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 the App Agent.