Altair Mistral getting started

Version 2.14.1

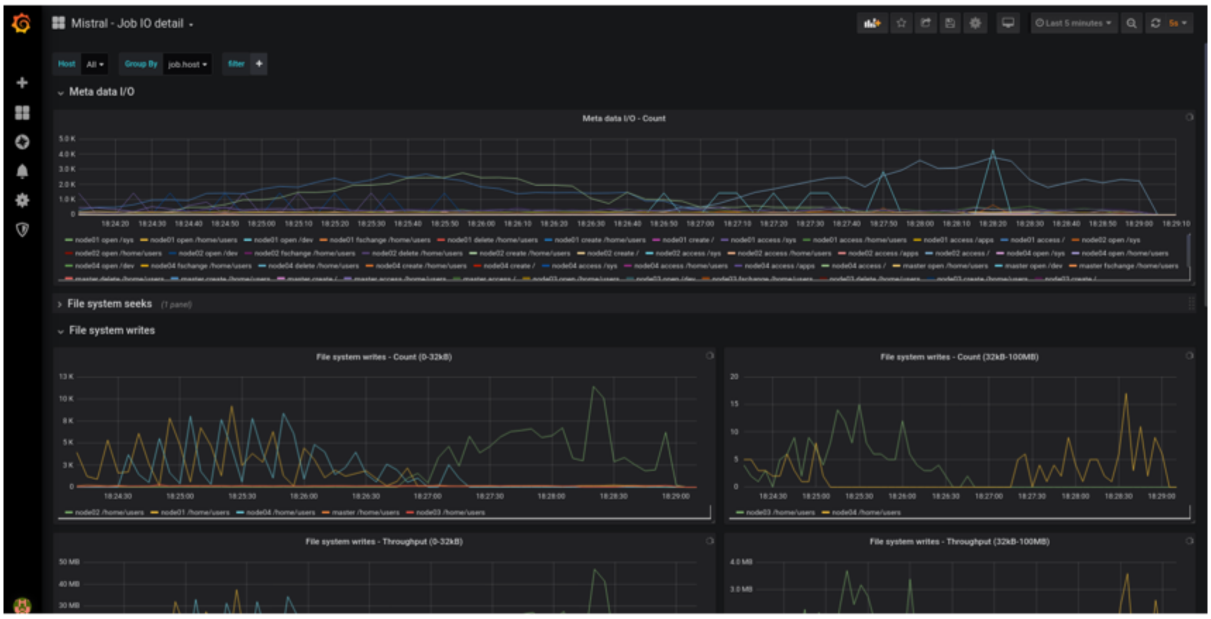
Contents

1 Introduction and product overview

This guide will show you how to get started with Altair Mistral, I/O monitoring tool. For more information about Mistral please refer to the demo videos and user manuals. Mistral was originally developed by Ellexus, now an Altair company.

1.1 Mistral: per-job I/O monitoring across HPC and distributed systems

Mistral is the leading application monitoring tool for HPC and scientific applications. Mistral is lightweight enough to run in production, but flexible enough to ensure that you get the most from on-prem HPC and that you have the information you need to manage hybrid cloud. Mistral is a storage agnostic tool that monitors I/O, CPU and memory, quickly locating rogue jobs, storage bottlenecks and keeping track of what is running on the clusters day-to-day.


Mistral - Job IO detail
Host All
Group By job.host
filter
Meta data I/O
Meta data I/O - Count
File system seeks
File system writes
File system writes - Count (0-32kB)
File system writes - Count (32kB-100MB)
File system writes - Throughput (0-32kB)
File system writes - Throughput (32kB-100MB)

1.2 Understanding how applications are using shared storage

As distributed systems and compute clusters become more complex, the need for monitoring becomes more important. To be able to run the compute efficiently and to forecast and design for the future it is important to have visibility on what is being run today.

1.3 Live analytics and historical trends

Mistral logs data in a scalable database such as Elastic Search or InfluxDB so a graphing dashboard such as Grafana or Splunk can be used to view the information. These give you live updates so you can find rogue jobs quickly or look back over historical data.

1.4 Protect shared storage

Protect shared storage from rogue jobs and noisy neighbour applications with per-job I/O monitoring. It is easy for a single user to overload shared storage with too many jobs or with applications that have bad I/O patterns. Mistral shows you which user, host or job is hitting the file system with too many requests.

1.5 Workflow qualification

Catch problems before they happen with workflow qualification at scale. New workloads can be quickly tested with Mistral to find bad I/O patterns before they become a problem.

1.6 Mistral generates a live feed of:

- Data recorded per job or application
- Data recorded for each mountpoint or filesystem

- Reads, writes and meta data
- I/O performance It’s easy to add in custom metrics to measure machine health for a full system overview.

2 Installation and getting started

This guide assumes you have access to the user manuals and documentation.

2.1 Download

Go to the Ellexus Website Downloads page and fill out the form. The site will send you an e-mail which you can use to download tar files of the products. To install, just untar the product on any filesystem which is accessible from the hosts running workloads which you wish to trace or profile.

2.2 Database and Dashboard

Mistral can be run on a single application, but to visualize data for multiple applications you will need to set up a database, to collect the data from each application, and a dashboard, to present visualisations of the data in the database. Mistral can push data to various databases such as Elasticsearch and Splunk. For customers who do not already have a log-aggregation infrastructure in place we recommend starting with Elasticsearch, and the Grafana dashboard.

2.3 Test run Mistral

We recommend that you initially run Mistral against a small program, logging the data to a local file on a disk.

```
# This can list several license servers, for failover
# reliability. Consult Altair licensing documentation.
$ export ALTAIR_LICENSE_PATH=<port>@<server>

$ export MISTRAL_CONTRACT_MONITOR_GLOBAL=<mistral path>/samples/monitoring.kitchensink.contract
$ export MISTRAL_LOG_MONITOR_GLOBAL=<path to output log file>
$ <mistral path>/mistral /bin/ls

$ cat <output log file>
```

Check that there are no errors written to the `mistral-error.log` file and that the output log file has data. This will ensure that you have the files all in the correct location, the licence is working, and there are sufficient permissions.

A contract file is a configuration file telling Mistral what data to collect. We provide two sample contract files with Mistral, in the "samples" directory: `monitoring.kitchensink.contract` and `monitoring.lightweight.contract`.

This example uses the first of those, the "kitchen sink" contract, which monitors everything at fairly high frequency: rates and bandwidths of all I/O operations on all mount points, at several different operation size ranges. This is useful for testing, but should not be used in production as it outputs a large volume of data and can have a high overhead. The "lightweight" contract is provided as an example of something more suitable for production use. We recommend that you study the results of testing with the "kitchen sink" contract, to guide the creation of a custom contract, perhaps similar to "lightweight", but tailored to your installation and profiling requirements.

2.4 Run Mistral Healthcheck (Optional)

In order to quickly see the data that Mistral can collect you can run Mistral with a more representative application now to create a Mistral log file.

```
# This can list several license servers, for failover
# reliability. Consult Altair licensing documentation.
$ export ALTAIR_LICENSE_PATH=<port>@<server>

$ export MISTRAL_CONTRACT_MONITOR_GLOBAL=<mistral path>/samples/monitoring.kitchensink.contract
$ export MISTRAL_LOG_MONITOR_GLOBAL=<path to output log file>
$ <mistral path>/mistral <myapp arg0 arg1…>
```

Once your application has completed, you can then generate a Mistral Healthcheck report using the following command:

```
$ <mistral path>/tools/mistral_report.sh -o <path to output dir> <output log file>
```

This will create a stand-alone HTML report of the I/O from your application.

2.5 Database support

Mistral has several output plug-ins for sending data to different databases. We would recommend using Elasticsearch, but we also support Splunk, InfluxDB and FluentBit (a log forwarder).

2.6 Install Elasticsearch and Grafana

Install Elasticsearch and Grafana and set-up the Mistral Elasticsearch plug-in to push data to your Elasticsearch database.

(For installation tips, see the separate section below).

Edit the `elastic_plugin.conf` file so that it has the correct plugin path, hostname/IP address, database name, username (if needed - remove these lines if authentication is not used), error log path, executable path and password. Once this is done you can now run the same test as above, but with an additional variable defined:

```
# This can list several license servers, for failover
# reliability. Consult Altair licensing documentation.
$ export ALTAIR_LICENSE_PATH=<port>@<server>

$ export MISTRAL_CONTRACT_MONITOR_GLOBAL=<mistral path>/samples/monitoring.kitchensink.contract
# This log file should be unique per host.
# If stored on shared storage this can be achieved by adding %h
# which will be replaced by the hostname
# e.g.
# export MISTRAL_LOG_MONITOR_GLOBAL=/apps/altair/logs/output-%h.log
$ export MISTRAL_LOG_MONITOR_GLOBAL=<path to output log file>
$ export MISTRAL_PLUGIN_CONFIG=<path>/elastic_plugin.conf
$ <mistral path>/mistral /bin/ls
```

You should now check if data has been added to the Elasticsearch database. This can be done either with a quick curl command or Kibana (if you have it installed). If data is being successfully passed to Elasticsearch, it will not also be written to the Mistral output log file.

2.7 Set-up the Grafana dashboard

Set-up the Grafana dashboard and check to see that you can see the data added in the previous step.

First set-up the Mistral Elasticsearch database as a data source in Grafana. If possible, name this data source `Mistral` to match the supplied dashboard. See Grafana data source for further details.

Import the `sample_grafana_dashboard.json` file as a new dashboard in Grafana, this can be done by going to http://grafana url/dashboard/import. See Importing sample grafana dashboard for further details.

Check that the time range at the top right of the dashboard covers when the previous tests were run, and check to see that there is some data visible. Once this is done, you should now be able to run more representative jobs and fine-tune the dashboard to make sure that you only have the information that you are interested in.

2.8 Job scheduler integration

Integrate Mistral with your job scheduler to automatically insert Mistral on specific queues.

See user manual for scheduler-specific integration instructions.

3 Installing Elasticsearch and Grafana

For an evaluation Elasticsearch and Grafana can easily be installed on a single VM, but when planning to go to production you should take care to estimate the likely volume of Mistral data, and provision and configure both Elasticsearch and Grafana appropriately. The open source community versions of these tools are absolutely fine for the evaluation: we don't use any enterprise-specific functionality.

3.1 Elasticsearch Install

We recommend installing both Elasticsearch (database) and Kibana (database web front-end). You can do this from the repositories:

https://www.elastic.co/guide/en/elasticsearch/reference/current/rpm.html#rpm-repo

https://www.elastic.co/guide/en/kibana/current/rpm.html#rpm-repo

3.1.1 Elasticsearch configuration

During an evaluation the following configuration changes allow a single node installation to be accessed without authentication from other machines on the network:

`/etc/elasticsearch/elasticsearch.yml`

```
# Identify this host
node.name: <hostname>
# Store log files in folder in /var/log
path.logs: /var/log/elasticsearch
# Listen on localhost and other network addresses
network.host: ["localhost", "127.0.0.1", "<IP address>", "<hostname>"]
# Mark this host as a master, no discovery required - works for single nodes
cluster.initial_master_nodes: ["<hostname>"]
# Allow larger queries from Grafana
search.max_buckets: 200000
# Allow more data per node
cluster.max_shards_per_node: 10000
# Mark this host as the only one to query for discovery
discovery.seed_hosts: ["<hostname>"]
```

Should you wish to install user security, the Mistral output plug-in supports this, you can do this using the xpack.security options.

You may need to allow ElasticSearch through the firewall. On RHEL/CentOS 7 the following commands should do this:

```
$ firewall-cmd --zone=public --permanent --add-port=9200/tcp
$ firewall-cmd --reload
```

3.1.2 Kibana configuration

The Kibana installation may require the following changes to be externally accessible:

`/etc/kibana/kibana.yml`

```
# Listen on all network connections
server.host: "0.0.0.0"
# Server name - display purposes only
server.name: "<hostname>"
```

You may need to allow Kibana through the firewall. On RHEL/CentOS 7 the following commands should do this:

```
$ firewall-cmd --zone=public --permanent --add-port=5601/tcp
$ firewall-cmd --reload
```

3.2 Elasticsearch index template

Once Elasticsearch is installed you will need to create an index template. Our plug-in creates a new Elasticsearch instance each day, of the form

```
<index name (normally mistral>-YYYY-MM-DD
```

To do this we provide a script in the `mistral_elasticsearch` plugin `schema/mistral_create_elastic_template.sh` - please run this script before trying to send data from Mistral. The `-?` argument provides details of how to run the script.

3.3 Grafana Install

Grafana (data visualization) can also be installed from the repositories:

https://grafana.com/docs/grafana/latest/installation/rpm/

You shouldn't need to make any configuration changes to the defaults provided.

You should be able to connect to Grafana web interface on port 3000

e.g. http://hostname:3000

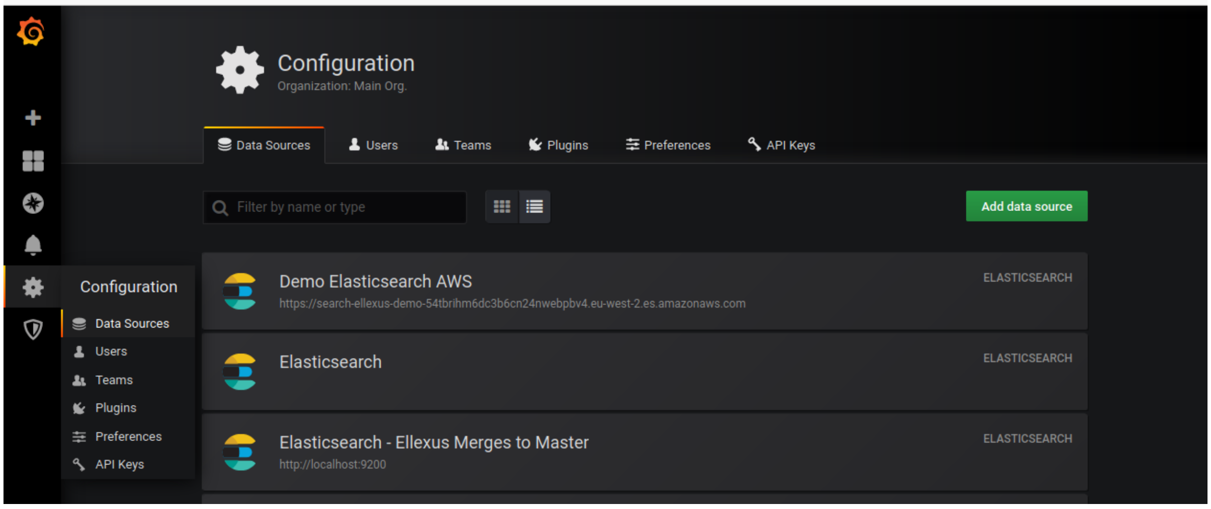
The default username is admin and the password is also admin. When first run the web U/I will ask you to change this.

You may need to allow Grafana through the firewall. On RHEL/CentOS 7 the following commands should do this:

```
$ firewall-cmd --zone=public --permanent --add-port=3000/tcp
$ firewall-cmd --reload
```

3.3.1 Grafana data source

Add a data source to Grafana by hovering over the settings cog on the left, followed by clicking on `Data Sources`. On the subsequent page you can then click `Add data source`.

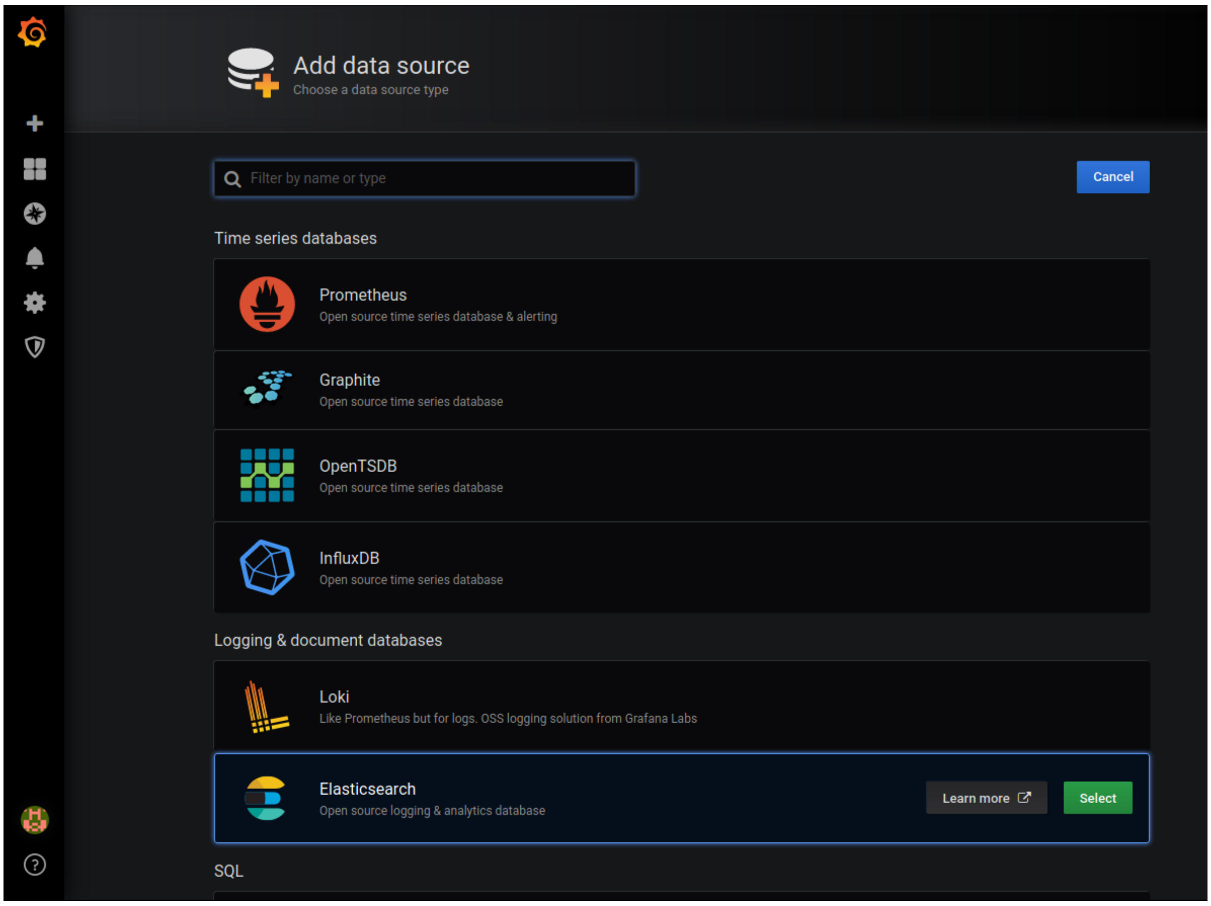


Demo Elasticsearch AWS — ELASTICSEARCH
https://search-ellexus-demo-54tbrihm6dc3b6cn24nwebpbv4.eu-west-2.es.amazonaws.com

Elasticsearch — ELASTICSEARCH

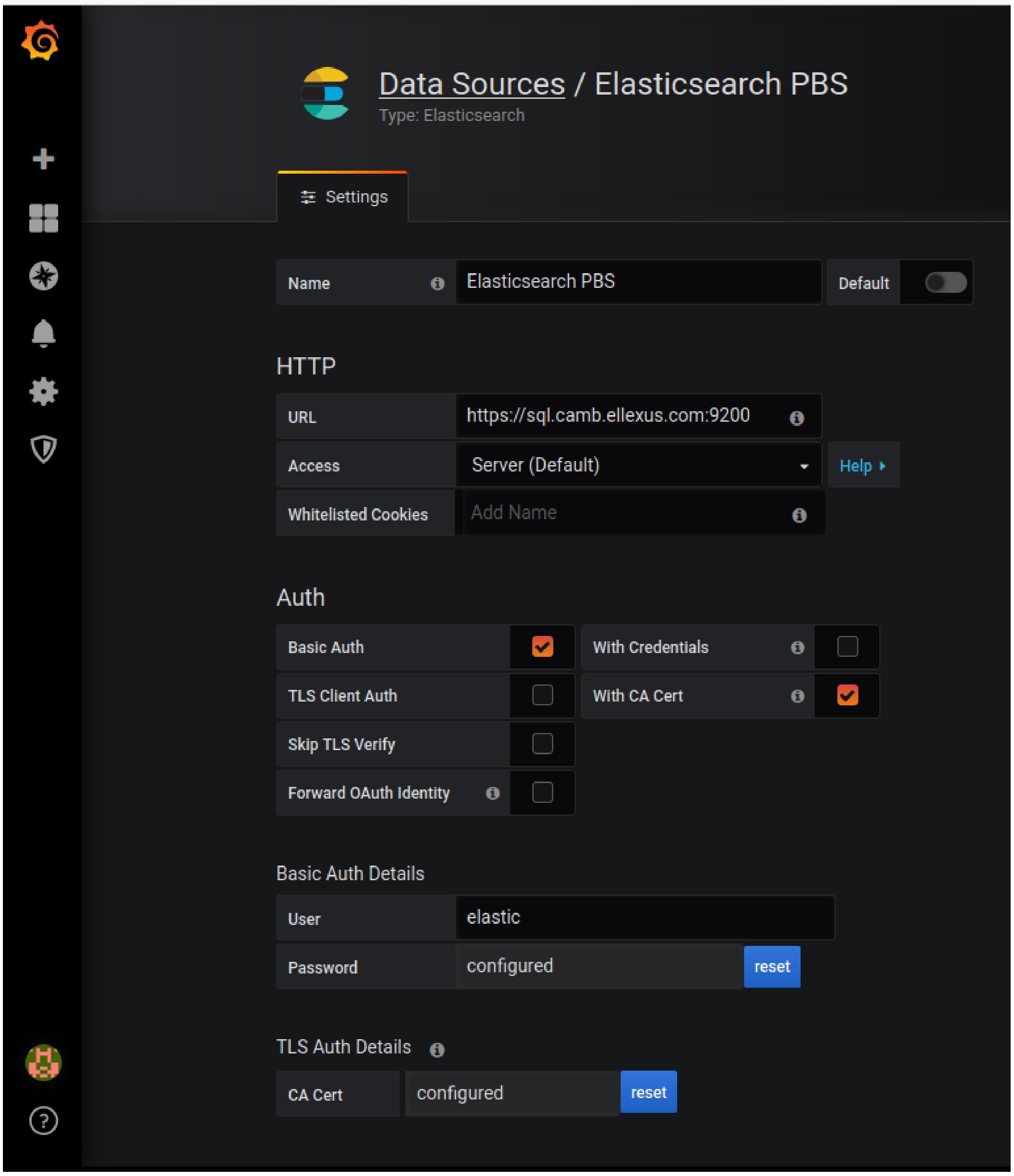
Elasticsearch - Ellexus Merges to Master — ELASTICSEARCH
http://localhost:9200

On the next page select Elasticsearch as the data source type.


Add data source
Choose a data source type
Filter by name or type
Cancel
Time series databases
Prometheus
Open source time series database & alerting
Graphite
Open source time series database
OpenTSDB
Open source time series database
InfluxDB
Open source time series database
Logging & document databases
Loki
Like Prometheus but for logs. OSS logging solution from Grafana Labs
Elasticsearch
Open source logging & analytics database
Learn more
Select
SQL

Enter the following into the new data source:

```
Name: Mistral
HTTP:
URL: http://<hostname>:9200
Access: Server (Default)
Auth:
# Leave the default settings, unless you have installed Elasticsearch authentication
Elasticsearch details:
Index name: [mistral-]YYYY-MM-DD
Pattern: Daily
Time field name: @timestamp
Version: 7.0+
Max concurrent Shard Requests: 256
```



Data Sources / Elasticsearch PBS

Type: Elasticsearch

Settings

Name	Elasticsearch PBS	Default	off

HTTP

URL	https://sql.camb.ellexus.com:9200
Access	Server (Default)
Whitelisted Cookies	Add Name

Auth

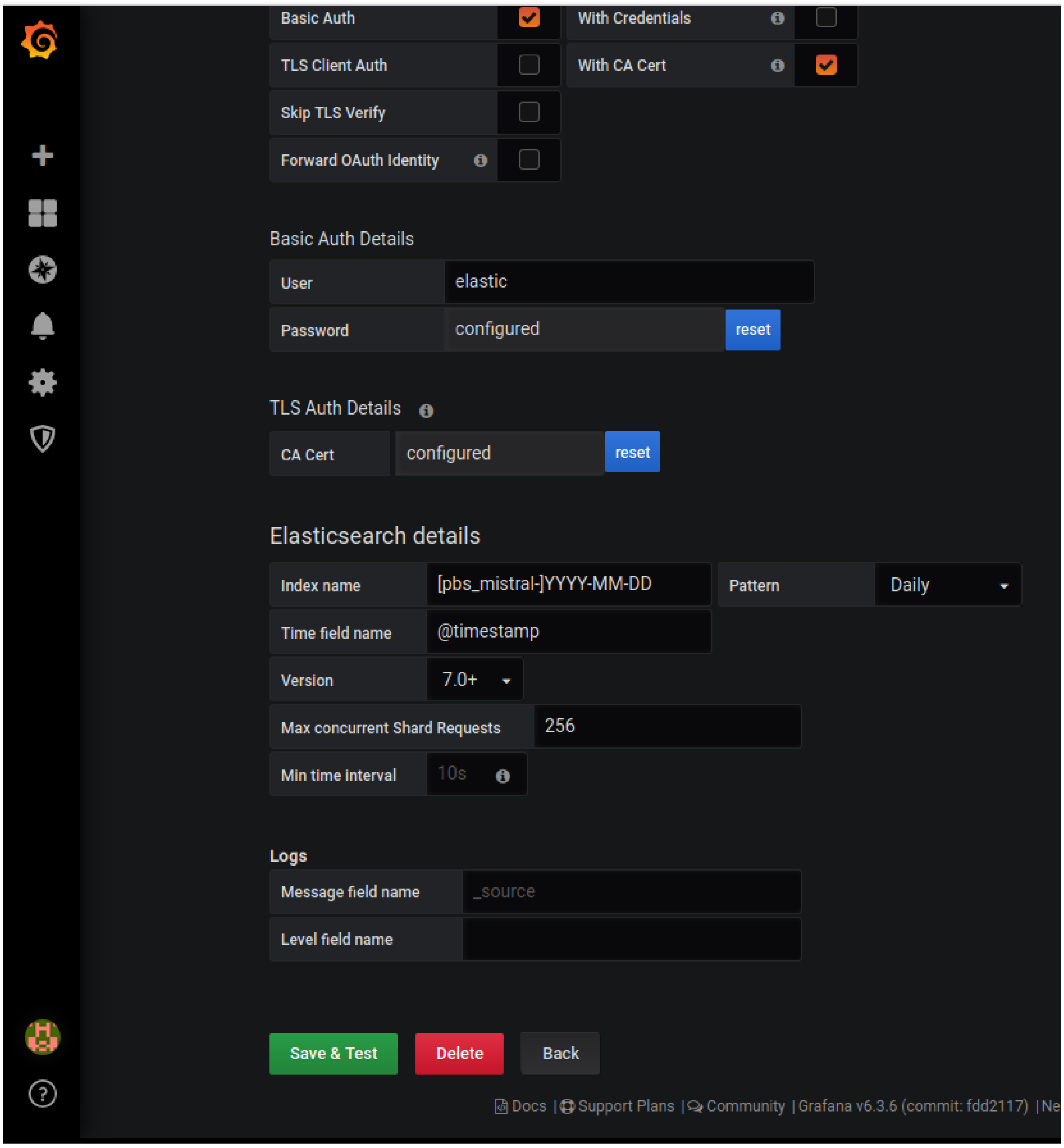
Basic Auth	✓	With Credentials	
TLS Client Auth		With CA Cert	✓
Skip TLS Verify			
Forward OAuth Identity			

Basic Auth Details

User	elastic
Password	configured

TLS Auth Details

CA Cert	configured



Basic Auth	✓	With Credentials	☐
TLS Client Auth	☐	With CA Cert	✓
Skip TLS Verify	☐		
Forward OAuth Identity	☐		

Basic Auth Details

User	elastic
Password	configured

TLS Auth Details

CA Cert	configured

Elasticsearch details

Index name	[pbs_mistral-]YYYY-MM-DD	Pattern	Daily
Time field name	@timestamp		
Version	7.0+		
Max concurrent Shard Requests	256		
Min time interval	10s		

Logs

Message field name	_source
Level field name	

reset
reset
Save & Test
Delete
Back
Docs | Support Plans | Community | Grafana v6.3.6 (commit: fdd2117) | Ne

All other settings can be left as their defaults, click `Save & Test`. If there are no errors, then you should be ready to import the sample dashboard.

3.3.2 Importing sample Grafana dashboard

Import the dashboard by hovering over the dashboards menu item (4 squares) and clicking `Manage`. On the subsequent page click `Import`.

You should then click `Upload .json file`, which will open a file browser. From there you can import the `sample_grafana_dashboard.json` from the Mistral samples directory.