

Altair Mistral InfluxDB Quick Start Guide

Version 2025.2.0



Contents

1	Introduction	3
1.1	Database and Dashboard	3
1.2	Install InfluxDB and Grafana	3
1.3	Set-up the Grafana dashboard	4
2	Installing InfluxDB and Grafana	5
2.1	Installing with Docker Compose	5
2.2	Installing with Podman Compose	6
2.2.1	Possible error with subnet allocation	6
2.2.2	Possible error with resource limits	6
2.2.3	Possible error with memory map area	7
2.2.4	Possible error with Healthcheck	7
2.2.5	Possible error with UIDs or GIDs	8
2.3	Manual Installation	8
2.3.1	InfluxDB Install	8
2.3.2	Grafana Install	9

1 Introduction

This guide should be used alongside the `mistral_quick_start` guide. For more information about Mistral please refer to the demo videos and user manuals. Mistral was originally developed by Ellexus, now an Altair company.

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

1.1 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, InfluxDB and Splunk. For customers who do not already have a log-aggregation infrastructure in place we recommend starting with Elasticsearch, and the Grafana dashboard. However this guide is for InfluxDB and Grafana.

1.2 Install InfluxDB and Grafana

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

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

Add a `plugin:` section to your `mistral_config.yaml` file, either in the Mistral installation directory or elsewhere on your system (for example, in `/etc` with other configuration files). We recommend you base this section on the `docs/samples/influxdb/v2_plugin.conf` file provided, with the correct plugin path, hostname/IP address, bucket, organisation, error log path, executable path and token.

Once this is done you can now run the same test as above, but with an additional variable defined:

```
#!/bin/sh
```

```
## Mistral launch script
```

```
export ALTAIR_LICENSE_PATH=<path to license file>
export MISTRAL_LOG=<path to output log file>
export MISTRAL_CONFIG=<path>/mistral_config.yaml
```

Now run `ls` under Mistral again to test the settings

```
$ source ./mistral_start.sh
$ <mistral path>/bin/mistral.sh /bin/ls
$ cat $MISTRAL_LOG
```

There should be no Mistral errors reported in `syslog` and this time there should be no new lines in the Mistral log because the data should now be going to the database. Mistral will only fall back to writing to the log file if there is a problem passing data to the database.

You should now check if data has been added to the InfluxDB database. The web interface for InfluxDB can be accessed by pointing your browser at `<server>:8086`. Once you have connected to InfluxDB you can search for Mistral data under the Data Explorer option and querying the `mistral` bucket within the correct time frame. You should be able to see that there is data, just from selecting fields to query on, by selecting the bandwidth measurement in the first filter.

1.3 Set-up the Grafana dashboard

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

First set-up the Mistral InfluxDB 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 following file as a new dashboard in Grafana:

- `docs/samples/influxd/fluxql_grafana.json`

This can be done by going to <http://grafana url/dashboard/import>. See [Importing sample grafana dashboards](#) for further details.

Check that the time range at the top right of the dashboards 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 dashboards to make sure that you only have the information that you are interested in.

The duration graphs in this dashboard will not be populated with data by default. If you are interested in collecting this information then you will need to enable duration measurements. This can be achieved by modifying your Mistral configuration file, setting these duration settings:

```
# Mistral configuration items to enable duration metering, so that  
# there is data for the performance dashboard.  
read:  
    duration: yes  
write:  
    duration: yes  
metadata:  
    duration: yes
```

2 Installing InfluxDB and Grafana

For an evaluation InfluxDB 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 InfluxDB 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.

2.1 Installing with Docker Compose

The easiest way to install InfluxDB with Grafana is through the provided `mistral-influxdb-container-setup.sh` script in `sbin`. This script is a wrapper for `docker compose` so you must have this installed to setup in this way. It has a single required argument specifying where the Mistral license file is:

```
./mistral-influxdb-container-setup.sh [-o /path/to/place/influx_token]
    [-e /path/to/mistral_influxdb]
    [-h <hostname>]
    [-d]
    [-r]
    [-c <container run time>]
    [-T]
    [-t timeframe]
    [-C /path/to/influx-certificate]
    [-K /path/to/influx-certificate-key]
    -l /path/to/mistral_license
```

Please check the usage message for all documented options and arguments. Before running this script, the `etc/influxdb/.env` file provided should be filled in with appropriate default values. Each option is documented as a comment above it in the file. When each option has been set, simply run the shell script. The script must be run in the directory it is provided in, and all provided files should also be present as-is.

Should any of the container setup phases fail then you can see the output by switching to the `etc/influxdb` directory and using `docker compose logs` or `docker compose up`. The `-r` option to `mistral-influxdb-container-setup.sh` may be useful in the event of failure to ensure a clean restart is done, as otherwise InfluxDB will persist in its setup.

If supplying your own certificate, it and the matching key must be available inside the `idbsetup` container. To do this, add an item to the volumes of the container `idbsetup`, like so:

```
idbsetup:
  ...
  volumes:
    - influxcerts:/etc/ssl/influxdb2/certs:rw
    - /path/to/dir/with/certificate:/path/to/dir/with/certifiacte:ro
  ...
```

The `ro` ensures it is read only. Also make sure that the path is different to every other volume and that the source (path outside the container) and the destination (inside the container) are the same. This is so that the script can both use the certificate outside the container with Mistral and pass the path as an environment variable to the inside of the `idbsetup` container, so that it can copy it somewhere all the other containers can use it from.

If your key is in a different directory, do the same for it as well. Then run the setup script with the `-C` and `-K` options pointing to the full path for your certificate and key respectively.

This install process will attempt to download and install Third Party content. Altair is not responsible for content developed and hosted by Third Parties. ALTAIR DISCLAIMS ANY AND ALL LIABILITY, INCLUDING ANY EXPRESS OR IMPLIED WARRANTIES, WHETHER ORAL OR WRITTEN, FOR SUCH THIRD-PARTY CONTENT. IN USING THIS SERVICE TO DOWNLOAD AND INSTALL THIS THIRD-PARTY CONTENT YOU ACKNOWLEDGE THAT NO REPRESENTATION HAS BEEN MADE BY US AS TO THE FITNESS OF THE THIRD-PARTY SERVICES FOR YOUR INTENDED PURPOSE.

2.2 Installing with Podman Compose

The `mistral-influxdb-container-setup.sh` script includes a `-c` option to utilize podman and podman-compose for installing Influxdb and Grafana. This script has been tested on Ubuntu 20.04.5 and Rocky Linux 8.9 with `-p` option as shown in the command below.

```
./mistral-influxdb-container-setup.sh -c podman -l <ALTAIR_LICENSE_PATH> \
-e <MISTRAL_INFLUXDB_PLUGIN_PATH>
```

There is a possibility of error based on the system configuration with podman and podman-compose, which can be resolved with the following recommendations.

2.2.1 Possible error with subnet allocation

If there is an error with subnet, then remove `mistral_influxdb` network and create manually with available private subnet as follow and run the `setup.sh` script once again.

```
ERRO[0000] "plugin type="bridge" failed (add): cni plugin bridge failed:
range set 0 overlaps with 1"
```

```
sudo podman network rm mistral_influxdb
sudo podman network create --subnet X.X.X.X/16 mistral_influxdb
```

2.2.2 Possible error with resource limits

If there is an error with resource limits, then update `/etc/security/limits.conf` and restart the system. The error happened in Ubuntu 20.04.5 by default.

```
ERROR: Cannot start service influxdb: OCI runtime create failed:
setting rlimits for ready process caused: error setting rlimit type 8:
operation not permitted: unknown
```

```
$ cat <<EOF | sudo tee -a /etc/security/limits.conf
*                soft    nofile    1024000
*                hard    nofile    1024000
*                soft    memlock   unlimited
*                hard    memlock   unlimited
influxdb         soft    nofile    1024000
```

```
influxdb      hard    nofile      1024000
influxdb      soft    memlock     unlimited
influxdb      hard    memlock     unlimited
root          soft    nofile      1024000
root          hard    nofile      1024000
root          soft    memlock     unlimited
EOF
```

```
$ sudo reboot
```

2.2.3 Possible error with memory map area

If there is an error with memory map area, then update `/etc/sysctl.conf` and apply the configuration in system. The parameters `net.ipv4.ip_forward`, `net.ipv4.tcp_retries2` and `vm.swappiness` are not directly related to `vm.max_map_count`, they are often configured together to optimize system performance. The error happened in Rocky Linux 8.9 by default.

```
ERROR: bootstrap check failure [1] of [1]: max virtual memory areas
vm.max_map_count [65530] is too low, increase to at least [262144]
```

```
$ cat <<EOF | sudo tee -a /etc/sysctl.conf
vm.max_map_count=262144
net.ipv4.ip_forward=1
net.ipv4.tcp_retries2=5
vm.swappiness=1
EOF
```

```
$ sudo sysctl -p
$ sudo systemctl restart network
```

2.2.4 Possible error with Healthcheck

If there is an error with healthcheck as shown below, then remove healthcheck sections from the `docker-compose.yml` file for all the services. After making these changes, run the `mistral-influxdb-container-setup.sh` script again. The `mistral-influxdb-container-setup.sh` script manages the healthcheck and container synchronization to ensure proper operation. The error happened in Rocky Linux 8.9 by default.

Traceback (most recent call last):

```
File "/bin/podman-compose", line 8, in <module>
    sys.exit(main())
File "/usr/local/lib/python3.8/site-packages/podman_compose.py", line 3504, in
↳ main
    asyncio.run(async_main())
File "/usr/lib64/python3.8/asyncio/runners.py", line 44, in run
    return loop.run_until_complete(main)
File "/usr/lib64/python3.8/asyncio/base_events.py", line 616, in
↳ run_until_complete
    return future.result()
```

```

File "/usr/local/lib/python3.8/site-packages/podman_compose.py", line 3500, in
↳ async_main
    await podman_compose.run()
File "/usr/local/lib/python3.8/site-packages/podman_compose.py", line 1743, in
↳ run
    retcode = await cmd(self, args)
File "/usr/local/lib/python3.8/site-packages/podman_compose.py", line 2500, in
↳ compose_up
    podman_args = await container_to_args(compose, cnt, detached=args.detach)
File "/usr/local/lib/python3.8/site-packages/podman_compose.py", line 1205, in
↳ container_to_args
    raise ValueError("'CMD_SHELL' takes a single string after it")
ValueError: 'CMD_SHELL' takes a single string after it
Failed to start docker containers with error code 1.

```

2.2.5 Possible error with UIDs or GIDs

If there is an issue with UIDs or GIDs in Rocky Linux 8.9, then run the `mistral-influxdb-container-setup.sh` script with `sudo` permission or migrate podman storage to ensure it executes correctly.

```

bash Error: copying system image from manifest list: writing blob: adding layer
... processing tar file(potentially insufficient UIDs or GIDs available in user
namespace (requested 0:42 for /etc/shadow): Check /etc/subuid and /etc/subgid if
configured locally and run "podman system migrate": lchown /etc/shadow: invalid
argument): exit status 1

```

```

sudo ./mistral-influxdb-container-setup.sh -c podman -l <ALTAIR_LICENSE_PATH> \
    -e <MISTRAL_PLUGIN_PATH>

```

```

podman system migrate
./mistral-influxdb-container-setup.sh -c podman -l <ALTAIR_LICENSE_PATH> \
    -e <MISTRAL_PLUGIN_PATH>

```

2.3 Manual Installation

2.3.1 InfluxDB Install

InfluxDB can be installed using an `.rpm` or `.deb`, the instructions are available here:

<https://docs.influxdata.com/influxdb/latest/install/?t=Linux>

2.3.1.1 InfluxDB configuration

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

```

$ firewall-cmd --zone=public --permanent --add-port=8086/tcp
$ firewall-cmd --reload

```


Once InfluxDB is installed and running, go to the web interface:

<http://localhost:8086>

From there you will be able to complete the setup. Create a default bucket called `mistral` and an access token which has read and write access.

2.3.2 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
```

2.3.2.1 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.

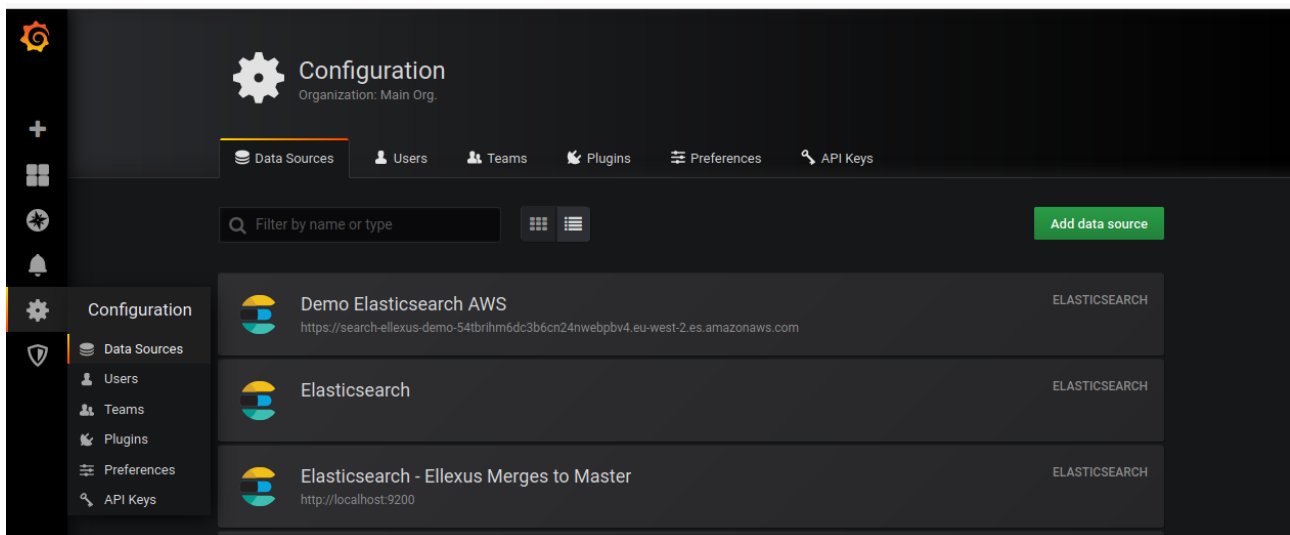


Figure 1: Grafana data sources menu

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

Enter the following into the new data source:

Name: `Mistral`
Query Language: `Flux`
HTTP:
URL: `http://<hostname>:8086`
Auth:
Leave the default settings
InfluxDB details:
Organization: `<Your organisation name>`
Token: `<InfluxDB API Token>`
Default Bucket: `mistral`

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 dashboards.

2.3.2.2 Importing sample Grafana dashboards

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 files in `docs/samples/influxdb/grafana/dashboards`.