# Logjam: User Guide

# Logjam.sh

Our project Logjam comes with a logjam.sh script that serves as a wrapper for our project and allows users to more easily interact with Logjam’s different functionalities. This includes creating the Logjam Docker image, spawning master and slave nodes for the elasticsearch cluster, and ingesting log bundles. To use the script, one can utilize the following flags:

* -b (build image)
* -c <name> (create elasticsearch master container <name>)
* -r <name> (create elasticsearch slave <name>)
* -i <container> <path to folder to ingest> (ingest log data on <container>)
* -d <container> (delete duplicates database on <container>)
* -e (creates the elasticsearch index)

# Manually running Logjam

If you’d like to run Logjam without the wrapper script, you can follow these steps:

## Building the Logjam Image

To use Logjam, you’ll have to create a container(s) using the Logjam Docker image. To build the image, you can simply run the following (inside the directory with the dockerfile):

sudo docker build -t logjam\_image .

If you’d like to edit the dockerfile, see the developer’s guide.

## Creating the Logjam Docker Containers

### Master Node

To create a container that has the full ELK stack - Elasticsearch, Logstash, and Kibana - use the following:

sudo docker run -p 5601:5601 -p 9300:9300 -p 9200:9200 -p 5044:5044 -v /home/admin/logjam/config/elasticsearch.yml:/etc/elasticsearch/elasticsearch.yml -v elk-data:/var/lib/elasticsearch -v /mnt/logjam/:/mnt/logjam/ --name <container> logjam\_image &

### Slave Node

These containers act as worked to help load balance. The slave containers only have elasticsearch installed on them. To manually create a worker container, run the following:

sudo docker run --rm=true -p 9200:9200 -p 9300:9300 -e LOGSTASH\_START=0 -e KIBANA\_START=0 -v /home/admin/logjam/config/elasticsearch-slave.yml:/etc/elasticsearch/elasticsearch.yml -v elk-data:/var/lib/elasticsearch -v /mnt/logjam/:/mnt/logjam/ --name <container> logjam\_image &

## Log Bundle Ingestion

To manually ingest logs, you can run the following two commands. The first command sets up the duplicates database, and can be skipped if it has already been set up.

sudo docker exec <container python /logjam/setup.py

The second command actually ingests the log bundles on a certain container at a specified path. You can add “-v” to include verbose output.

sudo docker exec <container> python /logjam/ingest.py <path>

# Kibana

Kibana is a visualization tool that allows users to perform queries on data ingested in Logstash and create various charts and graphs. By default, each new instance of Kibana does not come pre-installed with our dashboard and must be configured by the user (please see the Kibana section of the Installation Guide for more information).

From the “Dashboard” menu, you can access our “Common Failure/Error Concentration” dashboard, containing visualizations detailing common errors found throughout StorageGRID. These include aggregations for I/O, connection issues, and hard drive partition errors. Most importantly, our dashboard comes with an overview chart, detailing each ingested log’s timestamp, filename, category StorageGRID version, and categorize time. For the categorize time, this information provides a unique identifier for each log, linking back to its absolute path in our SQLite database.

Each visualization can be easily edited from the dashboard. By clicking the “Edit” option from the top-right menu, you can reorder position and size for each visualization. Further, by clicking the gear icon in the top-right corner of each visualization, you can edit its individual parameters. For specific parameter tuning, please refer to the official Kibana documentation for queries[1] and visualizations[2].

# References

[1] (https://www.elastic.co/guide/en/beats/packetbeat/current/kibana-queries-filters.html)

[2] https://www.elastic.co/guide/en/kibana/current/visualize.html