Setup and Operation Documentation

Search engine application with FSCrawler and Elastic

Contents

[Introduction](#_5y8e4o93y7eg)

[Hardware and Software Requirements](#_2prvw1b75e2i)

[Hardware](#_anvxqtiyupb6)

[Software](#_990gy5fizg0v)

[Installation/Setup](#_xr4np7lpmskg)

[Elastic, Kibana and Python front end](#_7w0k26nu2ww8)

[Docker compose to build and run Elastic and Kibana](#_rzi3l37bpv4r)

[Running FSCrawler](#_bb6lr0jopat1)

[Tools, libraries and versions](#_8l6y9t3xbg9)

[Useful commands](#_3k9383amrgtg)

[How to check the indexing is finished](#_5nm8xdrw8u6z)

[Troubleshooting](#_d5rfqscbpmvn)

[Using the Docker images in production](#_9pwlxmttgnpb)

[Set vm.max\_map\_count to at least 262144](#_iebadm4appnq)

[Linux](#_2y68cx250twr)

[macOS with Docker for Mac](#_29ff3nuwrk6o)

[Windows and macOS with Docker Desktop](#_d7wlsafrtlg4)

[Windows with Docker Desktop WSL 2 backend](#_za7edqqi2qwl)

[Good to have](#_8gspmysv3efk)

[References:](#_d7hfl7i8kvxt)

[FSCrawler](#_cbwfu5bz971e)

## 

## Introduction

This document provides guidance on the setup and operation of the FSCrawler with Elastic setup for a local file search tool. It covers topics such as hardware and software requirements, installation, configuration, and troubleshooting.

It is assumed that the user who is reading this document and will be performing installation and operations will have good knowledge of linux and docker.

## Hardware and Software Requirements

The following hardware and software are required for the software project:

### Hardware

* A computer with a minimum of 32 GB of RAM
* A hard drive with at least 256 GB of free space

### Software

* Ubuntu OS
* The software project itself
* A Java Development Kit (JDK)
* Docker is pre installed on the system
* Python 3 is pre-installed on the system. PIP needs to be installed to manage required packages for Python
* GIT cli
* Tools and library to extract and mount EWF format files
* Good text editor is preferred to edit files.

## Installation/Setup

The software project is mostly based on docker images and configuration files. The installation process is straightforward if this document is followed properly. All the required code, configuration files and specific softwares used with this project will be saved in a folder in the system. A backup copy can be kept locally or remotely in a secure place.

NOTE: We will be logging into the system via “index-root” user id. All commands must be executed with SUDO.

**Working/root project folder**: “/home/searchengine”

## Elastic, Kibana and Python front end

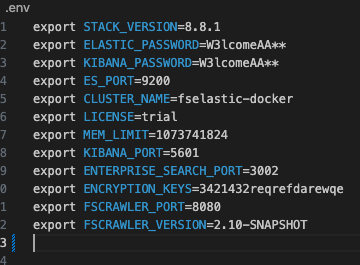
1. Go to the project root folder “/home/searchengine” . Run the following command if the folder is empty.

The setup has been done in this folder. All the code is pulled from github and saved here. For new setup, code needs to be pulled from github or copy it here from backup.

| $git clone https://github.com/hemantsh/elasticfscrawler.git |
| --- |

| $ cd elasticfscrawler |
| --- |

1. Open and check the “.env” file. This file contains key definitions that we will be using in the overall setup. Make sure all values are correct. These values will be preset on initial setup and should not be modified without the understanding of what part of application it will affect.



1. Run the following command to check docker images and containers. There should not be any images or containers for elasticsearch, kibana, enterprise search and FSCrawler in the system.

| $ docker images |
| --- |

* 1. Above command will show a list of existing images in the system. There should not be any images mentioned above when we set up for the first time.

| $ docker ps |
| --- |

* 1. Above command will give list of container and their status ( healthy etc)

1. Create the docker container/image for python elastic client backend and the web page for search.
   1. Go inside the UI folder.
   2. Setup Python and PIP: Run following commands in UI folder

| $sudo apt install python3-pip $sudo python3 -m pip install Elasticsearch $sudo python3 -m pip install flask  $sudo python3 -m pip install python-dotenv |
| --- |

* 1. Run the following command to check if all UI is working fine. This should prompt that server is running and give the url and port number. Access the page via browser, you should be able to go to search the home page. Note that, so far we have not turned the elastic server up so it will not function fully.

| $ sudo python3 setup.py |
| --- |

## Docker compose to build and run Elastic and Kibana

* 1. Go to the root of the project folder.
  2. There must be a file named “docker-compose.yaml”. This file contains the configuration to set up ElasticSearch, Kibana, EnterpriseSearch and UI for the project inside the docker container. Run the following commands.

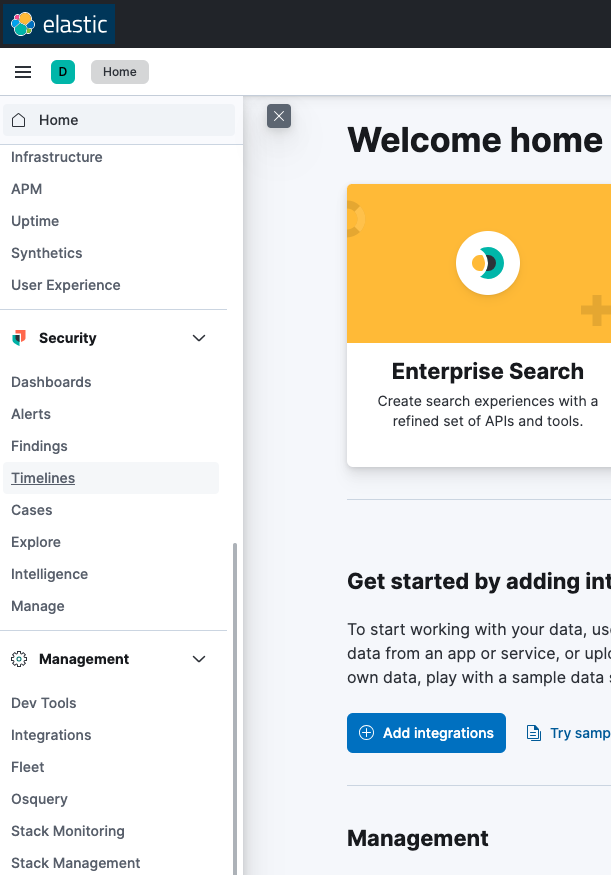
| $ docker-compose down -v  $ docker-compose up -d |
| --- |

* 1. Let the command finish. Make sure the command finishes successfully without any error messages. For any error, please refer to the troubleshooting guide in this document.
  2. Run the following command to check if the images are up and running.

| **$ docker images** |
| --- |

| hemant@Hemants-Mini FSElastic % docker images REPOSITORY TAG IMAGE ID CREATED SIZE docker.elastic.co/kibana/kibana 8.8.1 33e5d657f95c 4 weeks ago 877MB docker.elastic.co/enterprise-search/enterprise-search 8.8.1 a0155324a1d0 4 weeks ago 1.31GB docker.elastic.co/elasticsearch/elasticsearch 8.8.1 8552a239ee1e 4 weeks ago 1.34GB |
| --- |

1. Verify elastic and kibana setup
   1. Login to kibana using <http://localhost:5601> and open “Dev Tools”



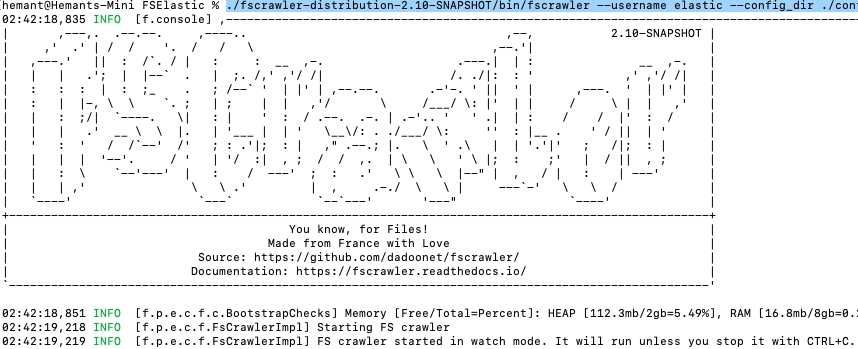
1. From dev tools, you should be able to verify the index details and perform some searches.



## Running FSCrawler

1. Go inside “fscrawler” folder
2. There will be two folders named “config” and “fscrawler-distribution-2.10-SNAPSHOT”
3. The config folder should have a pre-setup settings file ( config/<indexname>/\_settings.yaml ). This file contains important configuration for index naming and other indexing related stuff.
4. Run the command from project root folder

| **./fscrawler-distribution-2.10-SNAPSHOT/bin/fscrawler --username elastic --config\_dir ./config cases restart** |
| --- |



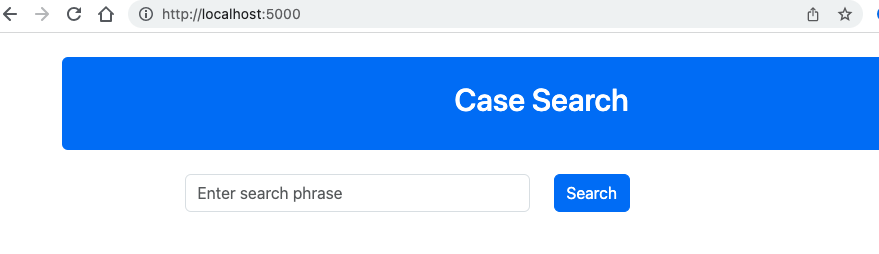
* 1. This command will start the indexing process from the root folder specified in settings and create an index with name “cases” as specified in command.
  2. Make sure to use the correct folder path in \_settings.yaml file “url” property as well as elastic url id and password. Crawler will use the path specified in “url” as the root and will scan all files and future addition to this folder and index into elastic search.
  3. Below is a sample of the settings file used by FSCrawler.



* 1. Check the index creation progress via Kibana



1. Run the search UI. Before we do this, make sure the UI application is up and running. Also elastic search must be running and have some documents indexed already.
   1. Open browser and open url http://localhost:5000
   2. This should present you with a search page.



When there are few documents in the new index created by FSCrawler, then start using the search page to test the case search.

After the software project is installed, it needs to be configured. This includes setting up the database, creating users, and configuring the application server. The configuration process is described in the project documentation.

If you encounter any problems with the software project, please refer to the troubleshooting section of the project documentation. The troubleshooting section provides tips and solutions for common problems.

## Tools, libraries and versions

| **Tool / library** | **Version** | **Link** |
| --- | --- | --- |
| Elastic | 8.8.1 | <https://www.elastic.co/guide/en/elasticsearch/reference/8.8/docker.html> |
| Kibana | 8.8.1 | <https://www.elastic.co/guide/en/kibana/current/docker.html> |
| Enterprise Search | 8.8.1 | <https://www.elastic.co/guide/en/enterprise-search/current/docker.html> |
| FSCrawler | 2.10-SNAPSHOT (Thu Jun 22 05:58:36 UTC 2023) version. Use this specific version ONLY for this setup. | <https://s01.oss.sonatype.org/content/repositories/snapshots/fr/pilato/elasticsearch/crawler/fscrawler-distribution/2.10-SNAPSHOT/fscrawler-distribution-2.10-20230622.055613-218.zip> |

## Useful commands

#### How to check the indexing is finished

In kibana run the following query.

| GET \_cat/indices/<indexname> The response will give detail of index including the number of items in the index.   yellow open idx 042NuuVOSI2iWZdgsYyN2A 1 1 64 2 1.1mb 1.1mb |
| --- |

During the indexing process, this count (the number indicated in RED) will keep increasing. Depending on the size of folder/files uploaded in the root folder, we can estimate the time remaining and then check this back after that amount of time to see if indexing is finished.

It is important to finish the indexing before we remove the files from the root folder for processing the next batch.

| docker-compose stop docker-compose up -d |
| --- |

| $docker logs <container\_id>  where container\_id is the ID of the unhealthy container |
| --- |

| $ wsl -d docker-desktop sysctl -w vm.max\_map\_count=262144 |
| --- |

| #Generate requirement file from current packages installed pip freeze > requirements.txt  #install python packages python3 -m pip install Elasticsearch python3 -m pip install flask  #Check version and other details on installed package versions python3 -m pip show Elasticsearch python3 -m pip show flask |
| --- |

Run the crawler to index into “cases” index.

| $./fscrawler-distribution-2.10-SNAPSHOT/bin/fscrawler --username elastic --config\_dir ./config cases restart |
| --- |

## Troubleshooting

This document has provided guidance on the setup and operation of the software project. For more information, please refer to the project documentation.

Try: docker logs <container\_id>, where container\_id is the ID of the unhealthy container (for example f6de943335cf).

If the container fails to start with a message about vm.max\_map\_count, refer to the following Elasticsearch documentation for platform-specific solutions: [Using the Docker images in production](https://www.elastic.co/guide/en/elasticsearch/reference/8.8/docker.html#docker-prod-prerequisites).

### **Using the Docker images in production**

The following requirements and recommendations apply when running Elasticsearch in Docker in production.

#### Set vm.max\_map\_count to at least 262144

The vm.max\_map\_count kernel setting must be set to at least 262144 for production use.

How you set vm.max\_map\_count depends on your platform.

#### Linux

To view the current value for the vm.max\_map\_count setting, run:

| grep vm.max\_map\_count /etc/sysctl.conf |
| --- |

vm.max\_map\_count=262144

To apply the setting on a live system, run:

sysctl -w vm.max\_map\_count=262144

To permanently change the value for the vm.max\_map\_count setting, update the value in /etc/sysctl.conf.

#### macOS with Docker for Mac

The vm.max\_map\_count setting must be set within the xhyve virtual machine:

1. From the command line, run:

| screen ~/Library/Containers/com.docker.docker/Data/vms/0/tty |
| --- |

1. Press enter and use sysctl to configure vm.max\_map\_count:

| sysctl -w vm.max\_map\_count=262144 |
| --- |

1. To exit the screen session, type Ctrl a d.

#### **Windows and macOS with** Docker Desktop

The vm.max\_map\_count setting must be set via docker-machine:

| docker-machine ssh |
| --- |

sudo sysctl -w vm.max\_map\_count=262144

#### **Windows with** Docker Desktop WSL 2 backend

The vm.max\_map\_count setting must be set in the "docker-desktop" WSL instance before the ElasticSearch container will properly start. There are several ways to do this, depending on your version of Windows and your version of WSL.

If you are on Windows 10 before version 22H2, or if you are on Windows 10 version 22H2 using the built-in version of WSL, you must either manually set it every time you restart Docker before starting your ElasticSearch container, or (if you do not wish to do so on every restart) you must globally set every WSL2 instance to have the vm.max\_map\_count changed. This is because these versions of WSL do not properly process the /etc/sysctl.conf file.

To manually set it every time you reboot, you must run the following commands in a command prompt or PowerShell window every time you restart Docker:

| wsl -d docker-desktop -u root |
| --- |

sysctl -w vm.max\_map\_count=262144

## 

### Good to have

* Take backup of complete project files.
* Create a code repository and save the files into the repository for extra backup and version control.
* Test out the complete setup on a machine by following this document as a guide. This will make future installation and maintenance easier.

## References:

#### FSCrawler

<https://fscrawler.readthedocs.io/en/latest/index.html>

<https://fscrawler.readthedocs.io/en/latest/admin/fs/local-fs.html>

[Using the Docker images in production](https://www.elastic.co/guide/en/elasticsearch/reference/8.8/docker.html#docker-prod-prerequisites)