# GUI

# Setup

## VM allocation

For MS Windows users, using wsl subsystems

* Open cmd, run
* wsl
* then 🡪 sudo sysctl -w vm.max\_map\_count=264144

## Elasticsearch and kibana

Setting elasticsearch and kibana on docker

1. docker pull docker.elastic.co/elasticsearch/elasticsearch:8.4.3
2. docker network create elastic
3. Start Elasticsearch in Docker. A password is generated for the elastic user and output to the terminal, plus an enrollment token for enrolling Kibana.
   1. docker run --name es01 --net SE\_elastic -p 9200:9200 -p 9300:9300 -it docker.elastic.co/elasticsearch/elasticsearch:8.4.3

------------------------------------------------------------------------------------------------------------------------

-> Elasticsearch security features have been automatically configured!

-> Authentication is enabled and cluster connections are encrypted.

-> Password for the elastic user (reset with `bin/elasticsearch-reset-password -u elastic`):

0K81s\_pf\_hHC\*aiYz1xg

-> HTTP CA certificate SHA-256 fingerprint:

dfdc80fe337cbbea445d305f0bafccc0d0925e5606792e162ce56b601251f87d

-> Configure Kibana to use this cluster:

\* Run Kibana and click the configuration link in the terminal when Kibana starts.

\* Copy the following enrollment token and paste it into Kibana in your browser (valid for the next 30 minutes):

eyJ2ZXIiOiI4LjQuMyIsImFkciI6WyIxNzIuMTkuMC4yOjkyMDAiXSwiZmdyIjoiZGZkYzgwZmUzMzdjYmJlYTQ0NWQzMDVmMGJhZmNjYzBkMDkyNWU1NjA2NzkyZTE2MmNlNTZiNjAxMjUxZjg3ZCIsImtleSI6ImxIX3Bhb1FCeERaWV9HOTg5UmxKOk1qR1B3UWJEU0tDT2VMY0dwN1VwaEEifQ==

-> Configure other nodes to join this cluster:

\* Copy the following enrollment token and start new Elasticsearch nodes with `bin/elasticsearch --enrollment-token <token>` (valid for the next 30 minutes):

eyJ2ZXIiOiI4LjQuMyIsImFkciI6WyIxNzIuMTkuMC4yOjkyMDAiXSwiZmdyIjoiZGZkYzgwZmUzMzdjYmJlYTQ0NWQzMDVmMGJhZmNjYzBkMDkyNWU1NjA2NzkyZTE2MmNlNTZiNjAxMjUxZjg3ZCIsImtleSI6ImtuX3Bhb1FCeERaWV9HOTg5UmxJOkM1bTJyUVZKUUNtMGZqVEEyWDFCcEEifQ==

If you're running in Docker, copy the enrollment token and run:

`docker run -e "ENROLLMENT\_TOKEN=<token>" docker.elastic.co/elasticsearch/elasticsearch:8.4.3`

------------------------------------------------------------------------------------------------------------------------

* copy the generated enrollment token for adding new Elasticsearch nodes
  + docker cp es01:/usr/share/elasticsearch/config/certs/http\_ca.crt ./Certs
* verify that you can connect to your Elasticsearch cluster
  + curl --cacert Certs/http\_ca.crt -u elastic https://localhost:9200
  + curl --ssl-no-revoke --cacert Certs/http\_ca.crt -u elastic <https://localhost:9200>
* docker pull docker.elastic.co/kibana/kibana:8.4.3
* docker run --name kib-01 --net SE\_elastic -p 5601:5601 docker.elastic.co/kibana/kibana:8.4.3

# System

# References

* Elastic Search
  + Elastic Search Setup
    - <https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html>
  + Elastic search setup with docker
    - https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html#\_enroll\_additional\_nodes
  + Elastic search python API
    - <https://elasticsearch-py.readthedocs.io/en/v8.4.3/api.html#indicess>
  + Building a Search Engine using Elasticsearch in 15 minutes
    - <https://pub.towardsai.net/building-a-search-engine-using-elasticsearch-in-15-minutes-69b010ea420>
  + Use Fuzzy Query for search functionality
    - https://hackernoon.com/hosw-to-use-fuzzy-query-matches-in-elasticsearch-dh1h3167
* Kibana
  + Kibana Setup
    - https://www.elastic.co/guide/en/kibana/current/docker.html
* Error solutions
  + <https://stackoverflow.com/questions/51445846/elasticsearch-max-virtual-memory-areas-vm-max-map-count-65530-is-too-low-inc>
  + <https://answers.microsoft.com/en-us/windows/forum/all/ssl-error-preventing-connection-in-windows-10/192436e5-e37b-4b4c-a4e0-c4ec744b0f5c>
  + <https://www.elastic.co/guide/en/elasticsearch/client/python-api/master/connecting.html>
* Haystack + ElasticSearch
  + <https://medium.com/nerd-for-tech/semantic-search-with-elastic-search-and-pre-built-nlp-models-part-1-you-got-a-question-600e7564890>
  + <https://snrspeaks.medium.com/building-a-faster-and-accurate-search-engine-on-custom-dataset-with-transformers-d1277bedff3d> (better link)
* Pre-processing
  + Pre-processing Arabic text for machine-learning using the camel-tools Python package
    - <https://towardsdatascience.com/arabic-nlp-unique-challenges-and-their-solutions-d99e8a87893d>
* Optimization
  + faster and more optimized processes
    - <https://towardsdatascience.com/fuzzy-matching-at-scale-84f2bfd0c536>
    - <https://stackoverflow.com/questions/52673285/performance-of-pandas-apply-vs-np-vectorize-to-create-new-column-from-existing-c>
* BM25
  + <https://towardsdatascience.com/how-to-build-a-search-engine-9f8ffa405eac#:~:text=use%20Elasticsearch.%20This%20search%20engine%20was%20powered%20by%20incredibly%20simple%20term%2Dfrequency%2C%20inverse%20document%20frequency%20(or%20tf%2Didf)>
* TF-IDF + word vectors
  + <https://towardsdatascience.com/supercharging-word-vectors-be80ee5513d#:~:text=model%20which%20can%20handle%20out%2Dof%2Dvocabulary%20words>
* Kaggle approaches
  + <https://www.kaggle.com/code/greegtitan/creating-simple-search-engine>
* Advanced Solutions (choose either approach)
  + Complete AI Based Search Engine with Elasticsearch, Kubeflow and Katib
    - <https://towardsdatascience.com/building-a-complete-ai-based-search-engine-with-elasticsearch-kubeflow-and-katib-590c7b27eb8f>
    - <https://towardsdatascience.com/tutorial-basic-kubeflow-pipeline-from-scratch-5f0350dc1905>
  + smart search engine (Superfast searching, BM25, fastText Word vectors)
    - <https://towardsdatascience.com/how-to-build-a-smart-search-engine-a86fca0d0795>