

LEGACY: Installing Elastic Search

We don't currently support running Elasticsearch locally w/ the Atlantis Search Solution. This article pertains to running Elasticsearch locally with our legacy search solution. We have kept the article around because we may support running elastic search locally in the future and this article would be a good starting point for doing that.

Install Using Docker

Taken from [Elastic Search with Plugins in Docker](#)

1. Download Docker Desktop (Windows): [Docker Desktop](#)
2. Create a Dockerfile. This file can be saved anywhere you choose. There is many ways of doing this, but what needs to be done is to create a file that does not have an extension on it and call it Dockerfile. One way is to open an editor(VS Code, Notepad++, Sublime Text, etc) then save the file with that name and no extension. Sometimes you have to save the name with quotes around it (exp. "Dockerfile").
3. Once file is created enter the following code into the Dockerfile:

```
FROM elasticsearch:6.8.23
```

```
RUN /usr/share/elasticsearch/bin/elasticsearch-plugin install --batch ingest-attachment
```

4. Open power shell and navigate to the file location of the Dockerfile. Then run the following to build the image in docker:

```
docker build -t elasticsearch-ingest-attachment .
```

5. Next in power shell run the following to create the container and run for the first time:

```
docker run --name elastic-search-6.8.23 -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" elasticsearch-ingest-attachment
```

6. Elastic search is now up and running. The only problem is it is running in power shell and will stay up on the screen. To resolve this please close the power shell which will stop elastic search. Then open Docker, in the container menu find the elastic search container

and hit the start button. Elastic search will be then good to go.

Note: This install of Elastic Search includes the Ingest add-on that is required for Rubex.

For Testing if it is up and running. Run the following code in power shell: `curl -GET "localhost:9200"`

The following will then display which shows it is up and running:

```
{
  "name" : "64adb598c086",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "_jyHQb0LTesiJxenN4MO1g",
  "version" : {
    "number" : "6.8.23",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "ef48eb35cf30adf4db14086e8aabd07ef6fb113f",
    "build_date" : "2020-03-26T06:34:37.794943Z",
    "build_snapshot" : false,
    "lucene_version" : "8.4.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

DEBUGGING:

Sometimes elastic search can get into readonly mode. You can tell if this has happened when you can connect to elasticsearch successfully, but it fails to index or purge any new data. You can verify this is the issue by checking the error messages in utopia, or in the docker elasticsearch logs you can find this string.

```
max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]
```

To increase the max memory allocated to this container, run the following commands in PowerShell:

```
wsl -d docker-desktop
```

```
sysctl -w vm.max_map_count=262144
```

Stop the elasticsearch container and restart it.

NOTE:

If the database is still in read-only mode, run the following curl command from the docker terminal.

```
curl -XPUT -H "Content-Type: application/json" http://localhost:9200/nodebaseindexitemindex/_settings -d '{"index.blocks.read_only_allow_delete": null}'
```

If this STILL doesn't work (or the command fails), [download and unzip Kaizen.zip](#)

Then, once unzipped run kaizen.bat, and connect to localhost elasticsearch, then delete the index and re-run utopia to re-create the index.

It should work...

Revision #16

Created 21 July 2022 16:10:30 by Caleb Baker

Updated 12 August 2024 14:25:42 by Quinn Godfrey