

TEB SOLUTION DOCUMENT

INTRO

This project contains 3 main python scripts and 1 bash script to start all necessary platforms.

Those scripts are:

logger.py : This script creates logs in wanted pattern with 2mb log file sizes continuously.

Reader.py: This script reads the logs converts them to the proper format and converts them into kafka producer messages with using necessary python modules.

Consumer.py : This script consumes the logs from kafka consumer then, converts them to proper format and sends them to elasticsearch server.

Start.sh: This script starts all the platforms and scripts.

In this project I used different platforms for different purposes.

Platforms in this project are :

Apache Kafka – publish and consume purposes

Elasticsearch – database purposes

Kibana - Visualization purposes

To make all those platforms reachable in my solution I dockerized them into a single container.

I dockerized my solution into an ubuntu container because of my experience in ubuntu.

USAGE

Those commands below should be written to console.

To run the project you should build it with docker build commands.

Then you can run the project by writing:

```
#docker run -it -p 9200:9200 -p 5601:5601 <image-name>
```

(Port 9200 is for ElasticSearch, Port 5601 is for Kibana)

Then you will have connection to the terminal of container.

To run everything at once you should just write:

```
#!/start.sh ----> This script will start everyting. It is compulsory.
```

After this command you should be wait around 20-30 seconds to all platforms start and be ready.

Then, you can see the output of the project by reaching from the browser:

#localhost:5601

This will open the Kibana app. Simply click the “Dashboards” from the menu on the left then click “Teblogs” in next screen.

It will open a line graph in main windows. If you would like to change the refresh rate of graph simply click the auto-refresh button from top right and make it 5 seconds to see the graph change more continuously.

TO EXAMINE CODES

To examine my work, you should be connecting to container with:

```
# docker run -it -p 9200:9200 -p 5601:5601 <image-name>
```

Then, you can just write #cat command to see contents of any script.

Warning:

This docker image won't be containing text editors from the beginning.

So, if you would like to examine codes in a text editor you can download the “nano” via

```
#apt-get install nano
```

If you would like to reach data in Elasticsearch you can simply write

In your console:

```
#curl -H 'Content-Type: application/json' -X GET https://localhost:9200/teblogs/\_search?pretty  
to console.
```

In your browser:

Also, you can discover the data by clicking “Discover” button in Kibana platform.

If you would like to see continuous change you can see it in Dashboard with making auto-refresh 5 seconds.

In container's console:

When you run the “start.sh” in container's console you can see the messages coming from reader.py and consumer.py

Thank you for your time.