Introduction

4 Tasks where given in the link. These will be described here (together)

Installing Elasticsearch and Kibana in Docker

Set up and Installation

Step 1: Use the following commands or run **install-docker.sh (Task 1)**:
[Considered Ubuntu as base OS]

If docker is already installed on os. Remove it first and then install

```
sudo apt-get purge -y docker-engine docker docker.io docker-ce docker-ce-cli
sudo apt-get autoremove -y --purge docker-engine docker docker.io docker-ce
sudo rm -rf /var/lib/docker /etc/docker
sudo rm /etc/apparmor.d/docker
sudo groupdel docker
sudo rm -rf /var/run/docker.sock

sudo apt-get update
sudo apt install docker.io
sudo snap install docker
```

Step 2: Create a File named docker-compose.yml in the Task 1 directory.
Use the following commands

```
version: "3.7"
services:
```

```
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.12.0
    container_name: elasticsearch
    restart: always
    environment:
      - xpack.security.enabled=false
      - discovery.type=single-node
    ulimits:
      memlock:
        soft: -1
        hard: -1
      nofile:
        soft: 65536
        hard: 65536
    cap_add:
      - IPC_LOCK
    volumes:
      - elasticsearch-data-volume:/usr/share/elasticsearch/data
    ports:
      - "2048:9200"
  kibana:
    container_name: kibana
    image: docker.elastic.co/kibana/kibana:7.12.0
    restart: always
    environment:
      SERVER_NAME: kibana
      ELASTICSEARCH_HOSTS: http://elasticsearch:9200
    ports:
      - "4096:5601"
    depends_on:
      - elasticsearch
volumes:
  elasticsearch-data-volume:
    driver: local
```

Step 3: Save the File and run the following command under the same directory terminal.
Or run run-elk.sh(Task2)
sudo docker-compose -f docker-compose.yml up -d

Step 4: Last step is to finally see the end results.
Go to http://localhost:4096/ (localhost server we provided for Kibana) and http://localhost:2048/(Elasticsearch host server) to check if it's working fine.
Will get something like this on 2048:


```
{
        name: "6cb2f9947017",
        cluster_name: "docker-cluster",
        cluster_uuid: "gya89K-bTXS33rECCDnE8w",
        version:
        {
                number: "7.12.0",
                build_flavor: "default",
                build_type: "docker",
                build_hash: "4e6e4eab2297e949ec994e688dad46290d018022",
                build_date: "2022-01-06T23:43:02.825887787Z",
                build_snapshot: false,
                lucene_version: "8.8.0",
                minimum_wire_compatibility_version: "6.8.0",
                minimum_index_compatibility_version: "6.0.0-beta1"
        },
        tagline: "You Know, for Search"
}
```


Index Some data

Use python faker and random package and generate 10 FAKE student details i.e. name, age, gender, address, contact.

```
PUT /students
{
        "mappings":
        {

          "properties":
          {
            "name":     { "type": "text"  },
            "age" :         { "type": "integer" },
            "gender":{ "type": "keyword" },
            "address":       { "type": "text" },
            "contact": { "type": "text" }
          }

        }}
```

Though I have used python elasticsearch client.
Run the script named **index_data.py** on Task 3

This will create index if not exists and pull data from **FAKE_STUDENTS.py .**
Here custom wait function **wait here()** is used placed in index_data script