# Task:

Dockerization of open source zabbix application using docker and docker compose.

# Solution:

**1. Install Docker:**

- Installed Docker on the VM (Ubuntu) using the official Docker documentation (docs.docker.com)

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
 echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

- Verified Docker installation:

```
docker --version
```

**2. Install Docker Compose:**

- Installed Docker Compose using the official Docker documentation.

```
sudo apt-get update
sudo apt-get install docker-compose-plugin
```

- Verified Docker Compose installation.

```
docker-compose --version
```

**3. Clone the Open Source Zabbix Code:**

- Cloned the Zabbix code from GitHub using the provided URL

```
docker clone https://github.com/lukecyca/pyzabbix.git
```

- The Zabbix code was successfully cloned to the VM.

**4. Create docker-compose.yml File:**

- Created a docker-compose.yml file for the Zabbix application with the following content:

```
version: "3.8"

services:
  postgres-server:
    image: postgres:13-alpine
    environment:
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: zabbix

  zabbix-server:
    image: zabbix/zabbix-server-pgsql:alpine-${ZABBIX_VERSION:-6.2}-latest
    ports:
      - 10051:10051
    volumes:
      - /etc/localtime:/etc/localtime:ro
    environment:
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: zabbix
      ZBX_CACHEUPDATEFREQUENCY: 1
    depends_on:
      - postgres-server

  zabbix-web:
    image: zabbix/zabbix-web-nginx-pgsql:alpine-${ZABBIX_VERSION:-6.2}-latest
    ports:
      - 8888:8080
    volumes:
      - /etc/localtime:/etc/localtime:ro
    environment:
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: zabbix
    depends_on:
      - postgres-server
      - zabbix-server
    healthcheck:
      test: ["CMD", "curl", "-f", "http://localhost:8080/"]
      interval: 10s
      timeout: 5s
```

```
      retries: 3
      start_period: 30s
```

**5. Build and Start the Docker Containers:**

- Used Docker Compose to build and start the containers

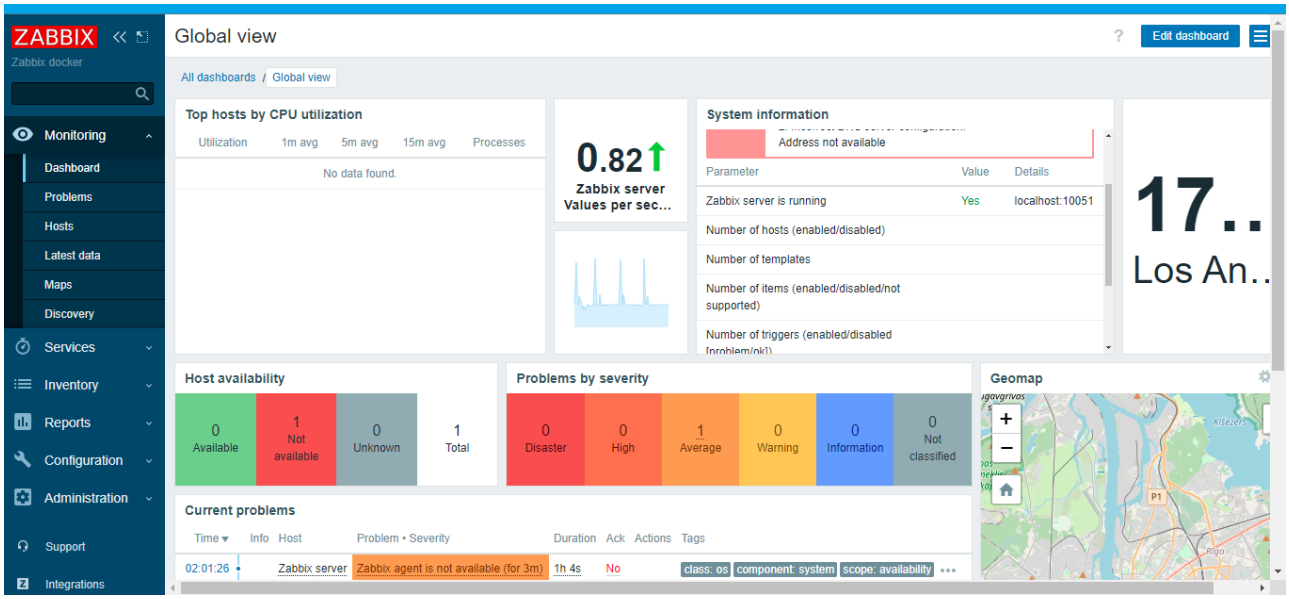
```
docker-compose up --build
```

- Verified that all containers are up and running

```
docker ps
```

**6. Access Zabbix:**

- Accessed the Zabbix application via the web browser at `http://<IP_address>:8888`
- Logged in using the Zabbix default credentials to see the Zabbix interface.




## Key Point:

Dockerfile was not required because the docker-compose.yml file provided all necessary configurations for deploying the Zabbix application. The file defined services for PostgreSQL, Zabbix server, and web interface, handling all settings and dependencies directly. This streamlined the deployment process, making a separate Dockerfile unnecessary.

## Conclusion:

The open-source Zabbix application was successfully dockerized and deployed using Docker Compose. All components, including the PostgreSQL database, Zabbix server, and Zabbix web interface, were configured

and run in Docker containers. The application was accessible via a web browser.The setup is now fully operational and ready for use.