Create a web application in Flask which maintains a hit counter in redis, as whenever user hits the page counter increments and gets updated in redis. Build this environment using docker compose

App.py

```
import time
import redis
from flask import Flask
import logging
from logging.handlers import RotatingFileHandler

app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)

def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
            app.logger.info('Website is loaded or reloaded')

        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)

@app.route('/')
def hello():
    count = get_hit_count()
    app.logger.info('count is {}.\n'.format(count))
    return 'Hello World! I have been seen {} times.\n'.format(count)

if __name__ == "__main__":
    handler = RotatingFileHandler('info.log', maxBytes=10000, backupCount=1)
    handler.setLevel(logging.INFO)
    app.logger.addHandler(handler)
    app.run(host="0.0.0.0", debug=True)
```

[*redis is the hostname of the redis container on the application's network. We use the default port for Redis, 6379*.]

Requirements.txt

```
flask
redis
```

Dockerfile

```
FROM python:3.7-alpine
WORKDIR /code
ENV FLASK_APP app.py
ENV FLASK_RUN_HOST 0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
COPY . .
CMD ["flask", "run"]
```

[

This tells Docker to:

1. *Build an image starting with the Python 3.7 image.*
2. *Set the working directory to /code.*
3. *Set environment variables used by the flask command.*
4. *Install gcc so Python packages such as MarkupSafe and SQLAlchemy can compile speedups.*
5. *Copy requirements.txt and install the Python dependencies.*
6. *Copy the current directory . in the project to the workdir . in the image.*
7. *Set the default command for the container to flask run.*

]

docker-compose.yaml

```
version: '3'
services:
  web:
    build: .
    ports:
     - "5000:5000"
  redis:
    image: "redis:alpine"
```

[This Compose file defines two services: web and redis.]

STEPS

1: Create a folder inside your working directory (Can work without creating a directory)
2: Add the app.py file with the code to print the required output
3: Add the requirements.txt and specify the dependencies to be installed
4: Add the Dockerfile with the command to start the flask application.
5: Now outside this directory add the docker-compose.yml file and add the services redis and webapp in it
6: run the command

```
docker-compose up
```

This creates the containers specified in the docker-compose.yml file
7: Navigate to the said route and refresh the page, and the counter will get updated on every refresh. But any changes to the app.py will not be reflected directly. we need to stop the containers and rebuild

Build and run your app with Compose

```
krutika@Quantiphi-930:~/Desktop/Containarization$ sudo docker-compose up
Building web
Step 1/9 : FROM python:3.7-alpine
 ---> 930a7e894675
Step 2/9 : WORKDIR /code
 ---> Running in a399e049cbef
Removing intermediate container a399e049cbef
 ---> 8ca37ba72ad1
Step 3/9 : ENV FLASK_APP app.py
 ---> Running in fef65ef1a298
Removing intermediate container fef65ef1a298
 ---> ff0f1c09fe90
Step 4/9 : ENV FLASK_RUN_HOST 0.0.0.0
 ---> Running in e94fc11b5fe1
Removing intermediate container e94fc11b5fe1
 ---> 3c6115a9913b
Step 5/9 : RUN apk add --no-cache gcc musl-dev

krutika@Quantiphi-930:~$ sudo docker ps -a
[sudo] password for krutika:
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS
PORTS                    NAMES
5f44c1fd3f06        containarization_web   "flask run"              23 minutes ago      Up 23 minutes
0.0.0.0:5000->5000/tcp   containarization_web_1
fea775f85a59        redis:alpine           "docker-entrypoint.s…"   23 minutes ago      Up 23 minutes
6379/tcp                 containarization_redis_1
41381ef7e32c        q2:latest              "python app.py"          21 hours ago        Exited (0) About
an hour ago                      krutika
```

ebb11422bacb 28422a27ebfc "python app.py" 22 hours ago Created krutika1