Create a volume, mount the volume and configure it in docker-compose.yaml file such that the code can be edited inside the container without having to rebuild the docker image. Use the same Web application created in Task 1 to test.

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 krutika! 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"
    volumes:
      - .:/code
    environment:
      FLASK_ENV: development
  redis:
    image: "redis:alpine"
```

[The environment key sets the FLASK_ENV environment variable, which tells flask run to run in development mode and reload the code on change.]

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 and add a volume to the webapp and set its source as the directory where app.py is stored
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.
8. Now go to app.py and change the code. Now on page refresh, we can see the changes reflected as shown in the attached screenshots. We do not need to rebuild

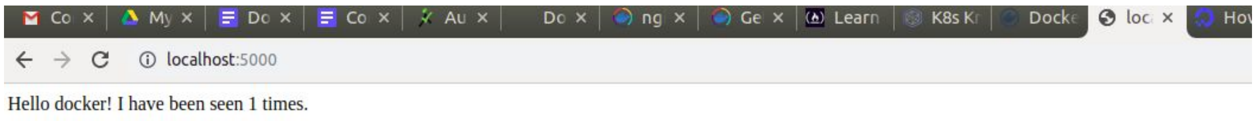
Build and run the app with Compose

```
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:~/Desktop/contain_q2$ sudo docker-compose ps
      Name                  Command              State          Ports
--------------------------------------------------------------------------------
containq2_redis_1   docker-entrypoint.sh redis ...   Up      6379/tcp
containq2_web_1     flask run                        Up      0.0.0.0:5000->5000/tcp
```

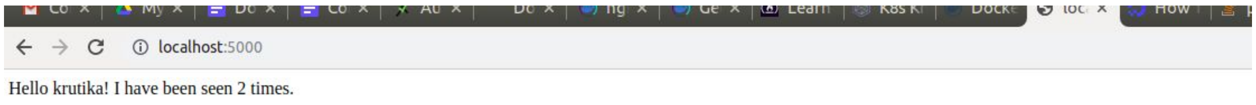
Screenshots:

Before change:



Hello docker! I have been seen 1 times.

After change:



Hello krutika! I have been seen 2 times.