

# Lab Docker (Default)

## Lab (Default) :

```
mkdir -p ynov-docker/{backend,cache}
```

```
touch backend/app.py backend/requirements.txt backend/Dockerfile cache/redis.conf  
cache/Dockerfile compose.yaml Caddyfile
```

## Backend

Create the file in backend/app.py :

```
import time  
import redis  
from flask import Flask  
  
app = Flask(__name__)  
cache = redis.Redis(host='redis', port=6379)  
  
def get_hit_count():  
    retries = 5  
    while True:  
        try:  
            return cache.incr('hits')  
        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()  
    return 'Helloooooo World! I have been seen {} times.\n'.format(count)  
  
if __name__ == "__main__":  
    app.run(host="0.0.0.0", port=5000)
```

Create the file in backend/requirements.txt :

```
Flask  
redis
```

Create the file in backend/Dockerfile :

```
# Use a lightweight Python base image
FROM python:3.10-slim

# Set environment variables to prevent Python from buffering stdout/stderr
ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

# Set work directory
WORKDIR /app

# Install dependencies
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

# Copy application code
COPY app.py .

# Expose the port that the Flask app will run on
EXPOSE 5000

# Run the Flask application
CMD ["python", "app.py"]
```

## Frontend

Create the file Caddyfile :

```
:80 {
    reverse_proxy backend:5000
}
```

## Compose

Create the file compose.yaml :

```
services:
  caddy:
    image: caddy:latest
    container_name: caddy
    ports:
      - "80:80"
    volumes:
      - "./Caddyfile:/etc/caddy/Caddyfile"
    networks:
      - app_network

  backend:
    build:
      context: ./backend
      dockerfile: Dockerfile
    container_name: flask_app
    restart: unless-stopped
    networks:
      - app_network

  redis:
```

```
image: redis:alpine
container_name: redis
restart: unless-stopped
networks:
  - app_network

networks:
  app_network: {}
```

Now run :

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

Your caddy should be accessible on <http://localhost> and redirecting to Flask.

---

Revision #5

Created 1 September 2024 14:58:29 by Alek

Updated 14 November 2024 08:12:56 by Alek