

클라우드 컴퓨팅 과제2 (1203)

2022320009 이수현

PartB

1. docker ps

```
C:\Users\shlee\CCHW2\backend>docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED        STATUS        PORTS          NAMES
6187ea62f6f6   backend:v2  "python app.py"  4 minutes ago  Up 4 minutes  0.0.0.0:5001->5001/tcp  backend
d74f27f156f1   frontend:v2 "python app.py"  11 minutes ago  Up 11 minutes  0.0.0.0:5000->5000/tcp  frontend
```

2. volume content

A. v1

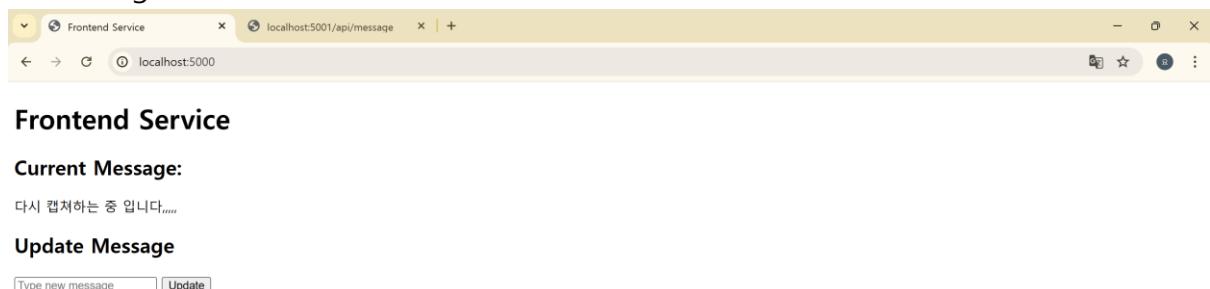
```
C:\Users\shlee\CCHW2\frontend>docker exec backend cat /data/message.txt
다시 캡처하는 중 입니다.....
```

B. v2

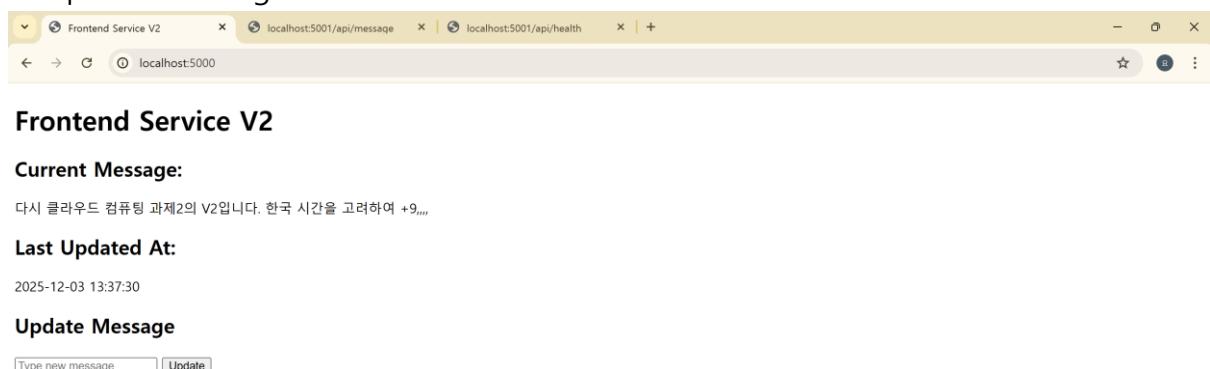
```
C:\Users\shlee\CCHW2\backend>docker exec backend cat /data/message.txt
다시 클라우드 컴퓨팅 과제2의 V2입니다. 한국 시간을 고려하여 +9,,, (updated at 2025-12-03 13:37:30)
```

3. frontend webpage

A. v1 message



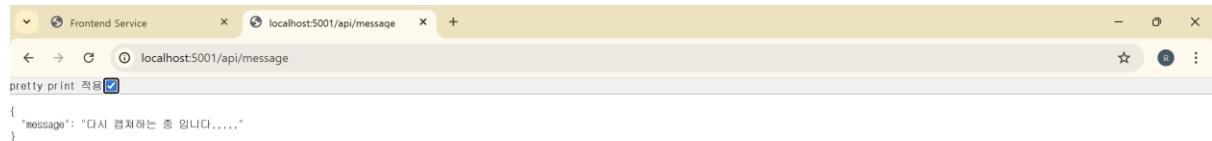
B. v2 updated message



4. browser screenshot

A. GET /api/message

i. V1



```
{ "message": "다시 래피는 중입니다...." }
```

ii. V2



```
{ "message": "다시 클라우드 컴퓨팅 과제2의 V2입니다. 한국 시간을 고려하여 +9.... (updated at 2025-12-03 13:37:30)" }
```

B. GET /api/health (V2)



```
{ "status": "healthy" }
```

5. Network

```
C:\Users\shlee\CCHW2\backend>docker network inspect appnet
[{"Name": "appnet", "Id": "81cea74fb94dfb306152faad7e76cc424f3dbf0cebc6a00c8d6850a89ba7ca6", "Created": "2025-12-03T03:22:28.292934209Z", "Scope": "local", "Driver": "bridge", "EnableIPv6": false, "IPAM": {"Driver": "default", "Options": {}, "Config": [{"Subnet": "172.20.0.0/16", "Gateway": "172.20.0.1"}]}, "Internal": false, "Attachable": false, "Ingress": false, "ConfigFrom": {"Network": ""}, "ConfigOnly": false, "Containers": {"6187ea62f6f677b1613677c356a59b4c9bae76d4768c211ee8a7392a15b7985d": {"Name": "backend", "EndpointID": "bbfbcbc1e2d93dbb20b974bd8b8081cb4bba63b98c68997d151b888b655c257367", "MacAddress": "02:42:ac:14:00:02", "IPv4Address": "172.20.0.2/16", "IPv6Address": ""}, "d74f27f156f18d3f2c224f6ec198985490b6f7e9501a7232d181a121e77761e1": {"Name": "frontend", "EndpointID": "8dbd1913010be2b1bbfd5d88189d688cd4fa46e8518c4dbc2bcbd78b26476d", "MacAddress": "02:42:ac:14:00:03", "IPv4Address": "172.20.0.3/16", "IPv6Address": ""}}, "Options": {}, "Labels": {}}]
```

6. Docker hub pages

A. Frontend

The screenshot shows the Docker Hub repository page for `hyund/frontend`. The sidebar on the left includes options like Repositories, Hardened Images, Collaborations, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main content area displays the repository details: `hyund/frontend`, pushed less than a minute ago, with a size of 61.2 MB. It features tabs for General, Tags (selected), Image Management (Beta), Collaborators, Webhooks, and Settings. Under the Tags tab, there are two entries: `v2` (pushed less than a minute ago) and `v1` (pushed about 2 hours ago). Each entry shows a digest, OS/ARCH (linux/amd64), and the last pull time (less than 1 day). Docker commands for pulling the image are provided for each tag.

TAG	Digest	OS/ARCH	Last pull	Compressed size
<code>v2</code>	05c96b76c111	linux/amd64	less than 1 day	48.98 MB
<code>v1</code>	5f65df5fc9e2	linux/amd64	less than 1 day	48.98 MB

B. Backend

The screenshot shows the Docker Hub repository page for `hyund/backend`. The sidebar on the left is identical to the frontend. The repository details show it was pushed 4 minutes ago with a size of 51.9 MB. The Tags tab is selected, showing two entries: `v2` (pushed 4 minutes ago) and `v1` (pushed about 1 hour ago). Each entry provides a digest, OS/ARCH (linux/amd64), and the last pull time (less than 1 day). Docker commands for pulling the image are available for both tags.

TAG	Digest	OS/ARCH	Last pull	Compressed size
<code>v2</code>	d7cfbe11f5db	linux/amd64	less than 1 day	47.4 MB
<code>v1</code>	927190f2ac6b	linux/amd64	less than 1 day	47.4 MB

PartC

```
C:\Users\shlee\CCHW2\frontend>curl http://172.29.160.1:5000
<!DOCTYPE html>
<html>
<head>
<title>Frontend Service V2</title>
</head>
<body>
<h1>Frontend Service V2</h1>

<h2>Current Message:</h2>
<p id="current-message">다시 클라우드 컴퓨팅 과제 2의 v2입니다. 한국 시간을 고려하여 +9,,,,</p>

<h2>Last Updated At:</h2>
<p id="last-updated">2025-12-03 13:37:30</p>

<h2>Update Message:</h2>
<form action="/update" method="post">
  <input type="text" name="new_message" placeholder="Type new message" required>
  <button type="submit">Update</button>
</form>
</body>
</html>
C:\Users\shlee\CCHW2\frontend>curl http://172.29.160.1:5001/api/message
{"message": "\ub2e4\uc2dc \ud074\ub77c\uc6b0\ub4dc \ucfc\uc81c2\uc758 V2\uc785\ub2c8\ub2e4. \ud55c\uad6d \uc2dc\uac84\uc744 \uaace0\ub824\ud558\uc5ec +9,,,,"}
updated at 2025-12-03 13:37:30"

C:\Users\shlee\CCHW2\frontend>curl http://172.29.160.1:5001/api/health
{"status": "healthy"}
```

PartD

1. How the frontend communicates with the backend

프론트엔드가 requests 라이브러리를 사용하여 백엔드 API주소로 HTTP요청을 보내고 그에 백엔드가 응답하면, 프론트는 그 응답에서 원하는 대답을 parsing하여 표시한다. 이때 보통의 프로젝트와 달랐던 점은 docker를 쓰기 때문에 IP주소가 아닌 hostname을 사용한다는 점이다. 실제로 BACKEND_URL = <http://backend:5001>인 이 주소가 백엔드 기본 url로, docker안에 있는 내장된 DNS가 자동으로 이에 맞는 컨테이너 IP주소로 변환해주기 때문에 가능하다. 장점은 컨테이너가 재시작되어 IP가 바뀌더라도 프론트엔드의 코드변경없이 진행할 수 있다는 점이다.

2. Why Docker needs a shared network

컨테이너 각각은 독립된 환경이기 때문에 서로 통신을 할 수 없는 상태이다. 그래서 서로 통신할 수 없기 때문에 appnet과 같은 shared network를 생성하여 두 개의 컨테이너를 같은 가상 네트워크에 묶어주어야한다. 덕분에 컨테이너 간에 포트 접근이 허용되고, IP주소 대신 hostname을 통한 내부 통신이 가능해진다.

3. What the volume is used for

컨테이너는 삭제되거나 재시작될 경우 내부 데이터가 모두 사라지는 Volatile이다. 그래서 백엔드 서비스가 저장하는 message.txt를 계속 저장하기 위해서는 다른 기능이 필요한데, docker volume을 이용해 호스트 시스템에 데이터를 저장하는 것이다. 이렇게 하면 컨테이너가 삭제되거나 재시작과는 상관없이 데이터는 호스트 시스템의 저장공간에 안전하게 유지된다.

4. What you changed for v2

백엔드의 경우 메시지를 저장할 때 그 때의 timestamp를 함께 저장하도록 하였다. 또한 timedelta를 이용하여 +9시간을 해주어 한국 시간으로 바꿔주었고, /api/health 엔드포인트를 추가했다. 프론트엔드도 요구된대로 제목을 V2로 바꾸고, 백엔드로부터 받는 응답에서 (updated at)을 기준으로 잘라서 text와 timestamp로 나누어 화면에 별도로 표시되도록 변경하였다.