

실습4 Report

2021320314 고건영

PART B

1. docker ps

- v1

```
ubuntu@VM-2-79-ubuntu:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c72f84f5282d frontend:v1 "python app.py" 6 minutes ago Up 6 minutes 0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp frontend
de77114475a4 backend:v1_ "python app.py" 6 minutes ago Up 6 minutes 0.0.0.0:5001->5001/tcp, [::]:5001->5001/tcp backend
```

- v2

```
ubuntu@VM-2-79-ubuntu:~/project$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
00d6a02295ba frontend:v2 "python app.py" About a minute ago Up About a minute 0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp frontend
e9b04cde759b backend:v2 "python app.py" About a minute ago Up About a minute 0.0.0.0:5001->5001/tcp, [::]:5001->5001/tcp backend
```

2. docker volume

- v1

```
ubuntu@VM-2-79-ubuntu:~/project$ sudo docker exec backend cat /data/message.txt
Hello! This is the v1 version message.ubuntu@VM-2-79-ubuntu:~$
```

- v2

```
ubuntu@VM-2-79-ubuntu:~/project$ sudo docker exec backend cat /data/message.txt
Hello! This is the v2 version message. (updated at 2025-11-24 08:14:13)ubuntu@VM-2-79-ubuntu:~/project$
```

3. Frontend webpage

- v1

Frontend Service

Current Message:

Hello! This is the v1 version message.

Type new message

- v2

← → ⌂ △ 주의 요함 43.128.145.7:5000

Frontend Service v2

Current Message:

Hello! This is the v2 version message.

Last Updated At:

2025-11-24 08:14:13

Update Message

Type new message

4. Backend API (webpage)

- v1

← → ⌂ △ 주의 요함 43.128.145.7:5001/api/message
pretty print 적용
{"message": "Hello! This is the v1 version message."}

- v2

← → ⌂ △ 주의 요함 43.128.145.7:5001/api/message
pretty print 적용
{"message": "Hello! This is the v2 version message. (updated at 2025-11-24 08:14:13)"}

← → ⌂ △ 주의 요함 43.128.145.7:5001/api/health
pretty print 적용
{"status": "healthy"}

5. docker network

- v1

```
[ubuntu@VM-2-79-ubuntu:~$ sudo docker network inspect appnet
[
  {
    "Name": "appnet",
    "Id": "190c5e1e621711161c7a8657fef8b6154344f780ee1ef62467ce72b35b59c6b7",
    "Created": "2025-11-24T14:32:25.456388466+08:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "IPRange": "",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Options": {},
    "Labels": {},
    "Containers": {
      "c72f84f5282d55758f4747a860745ea15cbc032ec46ee573dd9f8c1fe67ffb89": {
        "Name": "frontend",
        "EndpointID": "f3951ff77e5fc1c4d52ecb25eaa431186dade405486d9a60e3658035995104e1",
        "MacAddress": "b2:73:a3:45:bf:4d",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
      },
      "de77114475a4edcca051500184c229cc4e0d5b352495ff3e40816c890c1c95e1": {
        "Name": "backend",
        "EndpointID": "74131364fc4f723d83402d2050cca2405c42301d657f790dddebe9374b6a8551",
        "MacAddress": "8a:d6:86:3a:d5:50",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      }
    },
    "Status": {
      "IPAM": {
        "Subnets": {
          "172.18.0.0/16": {
            "IPsInUse": 5,
            "DynamicIPsAvailable": 65531
          }
        }
      }
    }
  }
]
```

- v2

```
[ubuntu@VM-2-79-ubuntu:~/project]$ sudo docker network inspect appnet
[{"Name": "appnet",
 "Id": "190c5e1e621711161c7a8657fef8b6154344f780ee1ef62467ce72b35b59c6b7",
 "Created": "2025-11-24T14:32:25.456388466+08:00",
 "Scope": "local",
 "Driver": "bridge",
 "EnableIPv4": true,
 "EnableIPv6": false,
 "IPAM": {
     "Driver": "default",
     "Options": {},
     "Config": [
         {
             "Subnet": "172.18.0.0/16",
             "IPRange": "",
             "Gateway": "172.18.0.1"
         }
     ]
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
     "Network": ""
 },
 "ConfigOnly": false,
 "Options": {},
 "Labels": {},
 "Containers": {
     "00d6a02295ba1c8d49f1f3430ff492ff2c8b6389fbf6e0a1e9020e1206e44915": {
         "Name": "frontend",
         "EndpointID": "950faf76b65e889e42623823f40ec93f3fdc39412e0f6bdd541044aacc552086",
         "MacAddress": "ae:60:92:91:ab:53",
         "IPv4Address": "172.18.0.3/16",
         "IPv6Address": ""
     },
     "e9b04cde759bc412b586ca6fdf2181a2cd131fecdcf4a9b347a699a22c1056bd": {
         "Name": "backend",
         "EndpointID": "ea8606fe409de6622a23bc1541e54d6b31affda7f319d50343c20dfd2d497f93",
         "MacAddress": "6e:ad:65:86:bb:b7",
         "IPv4Address": "172.18.0.2/16",
         "IPv6Address": ""
     }
 },
 "Status": {
     "IPAM": {
         "Subnets": {
             "172.18.0.0/16": {
                 "IPsInUse": 5,
                 "DynamicIPsAvailable": 65531
             }
         }
     }
 }
}]
```

6. Docker Hub repositories

kohgeonyeong/backend

Last pushed 2 minutes ago · Repository size: 51.9 MB · ⭐0 · ↓20

Add a description
Add a category

General Tags Image Management BETA Collaborators Webhooks Settings

Sort by Newest Filter tags Delete

TAG			
Digest	OS/ARCH	Last pull	Compressed size
e6482a8889c7	linux/amd64	less than 1 day	47.35 MB

TAG			
Digest	OS/ARCH	Last pull	Compressed size
4ae718d74909	linux/amd64	less than 1 day	47.35 MB

kohgeonyeong/frontend

Last pushed 2 minutes ago · Repository size: 55.1 MB · ⭐0 · ↓17

Add a description
Add a category

General Tags Image Management BETA Collaborators Webhooks Settings

Sort by Newest Filter tags Delete

TAG			
Digest	OS/ARCH	Last pull	Compressed size
75f8bac8196c	linux/amd64	less than 1 day	48.97 MB

TAG			
Digest	OS/ARCH	Last pull	Compressed size
902a85b0c777	linux/amd64	less than 1 day	48.97 MB

7. Test output (PART C)

```

[ubuntu@VM-2-79-ubuntu:~/project$ curl http://43.128.145.7:5000
<!DOCTYPE html>
<html>
<head>
    <title>Frontend Service v2</title>
</head>
<body>
    <h1>Frontend Service v2</h1>

    <h2>Current Message:</h2>
    <p id="current-message">Hello! This is the v2 version message.</p>

    <h2>Last Updated At: </h2>
    <p> 2025-11-24 08:14:13</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>

[ubuntu@VM-2-79-ubuntu:~/project$ curl http://43.128.145.7:5001/api/message
{"message": "Hello! This is the v2 version message. (updated at 2025-11-24 08:14:13)"}

[ubuntu@VM-2-79-ubuntu:~/project$ curl http://43.128.145.7:5001/api/health
{"status": "healthy"}]

```

PART D

1. How the frontend communicates with the backend

공유 네트워크 `appnet` 을 설정한 덕분에 컨테이너 간 통신은 Docker 내부 DNS를 기반으로 이루어진다. 따라서 실제 IP 주소를 몰라도 서비스 이름만으로 API를 호출할 수 있으며, 프론트엔드는 Docker 네트워크 안에서 `backend`라는 이름을 도메인처럼 사용해 `http://backend:5001/api/message` 와 같은 주소로 백엔드에 요청을 보낸다. 프론트엔드에서 GET 요청을 보낼 경우 백엔드에 저장된 메시지를 JSON으로 받아오고, POST 요청 시에는 새로운 메시지를 JSON 형태로 전달한다.

2. Why Docker needs a shared network

Docker에서는 서로 다른 컨테이너들이 같은 네트워크 환경에 연결되어야만 서로를 인식하고 통신할 수 있다. 기본 bridge 네트워크에서는 컨테이너 이름 기반 통신이 어렵거나 불가능하여 IP 주소를 직접 관리해야 하는 불편함이 있다. 이때 공유 네트워크를 사용하면 Docker가 컨테이너 이름을 내부 DNS로 자동 등록해주어, 프론트엔드는 `backend`라는 이름만으로 백엔드를 찾아갈 수 있다.

3. What the volume is used for

Volume은 백엔드가 저장하는 메시지 파일(`/data/message.txt`)을 컨테이너 외부에 보관하여, 컨테이너를 재시작하거나 삭제 후 다시 생성하더라도 기존 메시지가 유지되도록 한다. 컨테이너 내부 파일 시스템은 컨테이너가 종료되면 사라지는 구조이기 때문에, 메시지처럼 유지되어야 하는 데이터 등을 volume에 저장하여 보존할 수 있다.

4. What you changed for v2

백엔드에서는 `update_message()` 함수에서 메시지를 저장하기 전에 `datetime.now().strftime()` 을 사용해 타임스탬프 문자열을 생성하고, 전달받은 메시지 뒤에 "(updated at ...)" 형식으로 붙여 저장하도록 수정하고, 새로운 `/api/health` 엔드포인트를 추가하여 서비스 상태를 체크 할 수 있게 구현했다. 프론트엔드에서는 `index()` 함수에서 백엔드로부터 받은 전체 메시지를 "(updated at" 기준으로 분리하여 본문 메시지와 시간 값을 추출하고, HTML 템플릿에서 `{{ last_updated }}` 를 표시하는 영역을 추가했다.