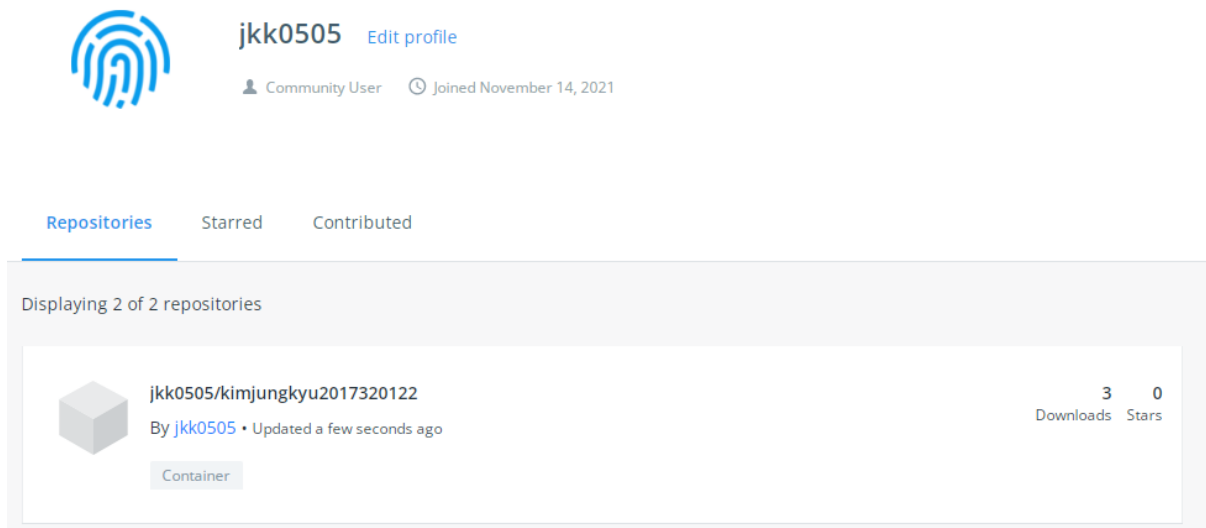


# Docker 실습 과제 보고서

2017320122 김정규

## 1. Docker Hub에 이미지 올려보기



Kimjungkyu2017320122라는 본인의 이름과 학번을 name으로 하는 이미지 파일을 생성하여 Docker Hub에 업로드하였다.

## 2. 이름과 학번을 출력하는 웹 서버를 컨테이너로 실행하고, 개인 노트북/PC에서 요청해서 결과 받기

```
[ec2-user@ip-172-31-16-174 ~]$ sudo docker pull httpd:latest
latest: Pulling from library/httpd
7d63c13d9b9b: Pull complete
ca52f3eaaa66: Pull complete
448256567156: Pull complete
21d69ac90caf: Pull complete
462e88bc3074: Pull complete
Digest: sha256:f70876d78442771406d7245b8d3425e8b0a86891c79811af94fb2e12af0fadeb
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

먼저 웹 기능을 이용하기 위해 아파치 이미지 파일을 다운받는다

```
$ sudo docker pull httpd:latest
```

```
[ec2-user@ip-172-31-19-82 ~]$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
d9523fb82f4f   httpd     "httpd-foreground"      2 minutes ago Up 2 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp
apache
[ec2-user@ip-172-31-19-82 ~]$ sudo docker inspect d9523fb82f4f | grep IP
      "LinkLocalIPv6Address": "",
      "LinkLocalIPv6PrefixLen": 0,
      "SecondaryIPAddresses": null,
      "SecondaryIPv6Addresses": null,
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "IPAMConfig": null,
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
[ec2-user@ip-172-31-19-82 ~]$ curl 172.17.0.2:80
<html><body><h1>It works!</h1></body></html>
```

먼저 현재 환경의 httpd 이미지 정보를 확인한다. 현재 포트 넘버가 과제가 요구하는 1234가 아니므로 이를 변경할 것이다.

```
$ sudo docker ps
```

```
$ sudo docker inspect d9523fb82f4f | grep IP
```

```
[ec2-user@ip-172-31-19-82 ~]$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
[ec2-user@ip-172-31-19-82 ~]$ sudo docker run -d -p 1234:80 httpd
d0a802d641c2306aa58d3a6a2b4f771ee80812695d3a8a0e8ed3b12178c6f136
[ec2-user@ip-172-31-19-82 ~]$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
d0a802d641c2   httpd     "httpd-foreground"      9 seconds ago Up 7 seconds   0.0.0.0:1234->80/tcp, :::1234->80/tcp
awesome_merkle
[ec2-user@ip-172-31-19-82 ~]$ sudo docker inspect d0a802d641c2
[
  {
    "Id": "d0a802d641c2306aa58d3a6a2b4f771ee80812695d3a8a0e8ed3b12178c6f136",
    "Created": "2021-11-14T11:54:48.890035372Z",
    "Path": "httpd-foreground",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 4466,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2021-11-14T11:54:50.254395113Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:1132a4fc88faaf5c19959f03535c1356d3004ced1978cb9c3f32e73d9c139532",
    "ResolvConfPath": "/var/lib/docker/containers/d0a802d641c2306aa58d3a6a2b4f771ee80812695d3a8a0e8ed3b12178c6f136/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/d0a802d641c2306aa58d3a6a2b4f771ee80812695d3a8a0e8ed3b12178c6f136/hostname",
    "HostsPath": "/var/lib/docker/containers/d0a802d641c2306aa58d3a6a2b4f771ee80812695d3a8a0e8ed3b12178c6f136/hosts",
    "LogPath": "/var/lib/docker/containers/d0a802d641c2306aa58d3a6a2b4f771ee80812695d3a8a0e8ed3b12178c6f136/d0a802d641c2306aa58d3a6a2b4f771ee80812695d3a8a0e8ed3b12178c6f136-json.log",
    "Name": "/awesome_merkle",
    "RestartCount": 0,
  }
]
```

원하는 포트 번호로 설정하기 위해 기존의 이미지를 없애고 포트 포워딩을 진행하여 새로운 httpd 이미지를 생성하였다. Docker의 기존 주소는 동일하나 이를 통해 퍼블릭 IP 주소와 설정한 포트 번호를 거쳐 접속이 가능하게 된다.

```
$ sudo docker stop d9523fb82f4f
```

```
$ sudo docker run -d -p 1234:80 httpd
```

```
$ sudo docker ps
```

```
$ sudo docker inspect d0a802d641c2
```

```
    "NetworkSettings": {
      "Bridge": "",
      "SandboxID": "a9c451c23b3872a700af4e7369795d073eb02a857243663bee7174bdecd4d734",
      "HairpinMode": false,
      "LinkLocalIPv6Address": "",
      "LinkLocalIPv6PrefixLen": 0,
      "Ports": {
        "80/tcp": [
          {
            "HostIp": "0.0.0.0",
            "HostPort": "1234"
          },
          {
            "HostIp": "::",
            "HostPort": "1234"
          }
        ]
      },
      "SandboxKey": "/var/run/docker/netns/a9c451c23b38",
      "SecondaryIPAddresses": null,
      "SecondaryIPv6Addresses": null,
      "EndpointID": "23630f2e489b3043e93dc94c181696090ca389aa69a65ac62cd53f785a99e215",
      "Gateway": "172.17.0.1",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "MacAddress": "02:42:ac:11:00:02",
      "Networks": {
        "bridge": {
          "IPAMConfig": null,
          "Links": null,
          "Aliases": null,
          "NetworkID": "0bddcd6fa582136c0bb9e17584e02512c9179f7e3006d3992f08455b3c9053d1",
          "EndpointID": "23630f2e489b3043e93dc94c181696090ca389aa69a65ac62cd53f785a99e215",
          "Gateway": "172.17.0.1",
          "IPAddress": "172.17.0.2",
          "IPPrefixLen": 16,
          "IPv6Gateway": "",
          "GlobalIPv6Address": "",
          "GlobalIPv6PrefixLen": 0,
          "MacAddress": "02:42:ac:11:00:02",
          "DriverOpts": null
        }
      }
    }
  }
}
]
[ec2-user@ip-172-31-19-82 ~]$ curl 172.17.0.2
<html><body><h1>It works!</h1></body></html>
[ec2-user@ip-172-31-19-82 ~]$
[ec2-user@ip-172-31-19-82 ~]$ sudo docker stop d0a802d641c2
d0a802d641c2
```

\$ curl <IPAddress>를 입력하여 설정한 서버가 잘 작동하는지 확인한다. 확인된 내용 상으로 서버에 접속하면 'It works!'라는 텍스트가 나올 것이다.

\$ curl 172.17.0.2



사이트에 연결할 수 없음

54.237.169.47에서 응답하는 데 시간이 너무 오래 걸립니다.

다음 방법을 시도해 보세요.

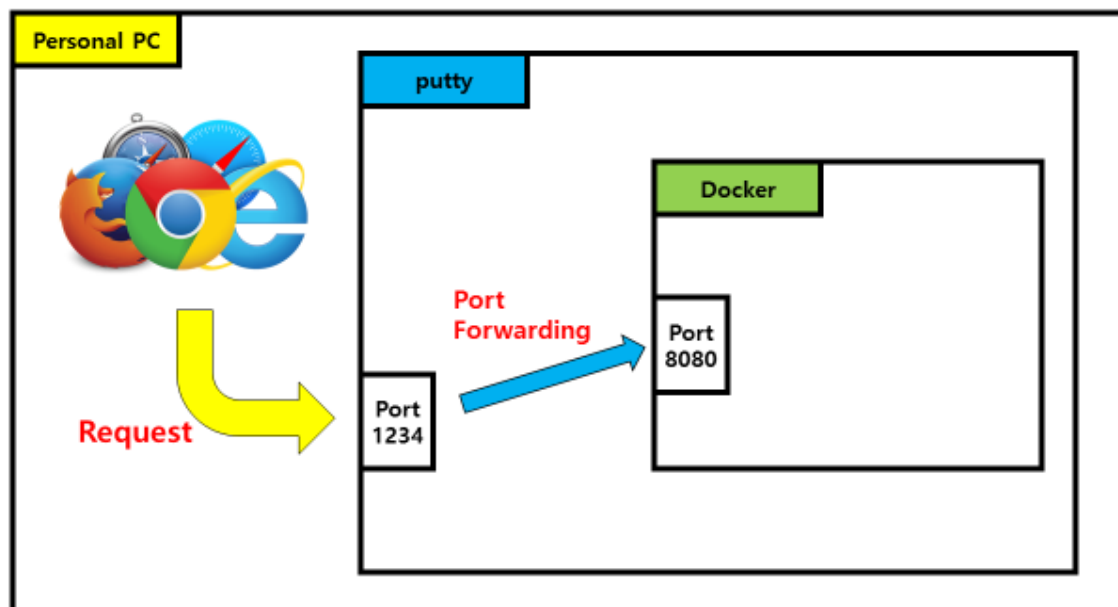
- 연결 확인
- 프록시 및 방화벽 확인
- Windows 네트워크 진단 프로그램 실행

ERR\_CONNECTION\_TIMED\_OUT

새로고침

세부정보

그러나 예상과는 다르게 서버 접속이 불가능하였다. 무엇이 문제일까?



상기 이미지는 본인이 러프하게 그린 모식도이다. 현재까지 진행한 내용으로 저 그림을 그대로 따라한 것과 같다. 그러나 서버 접속이 불가능하기에 AWS EC-2 의 보안 그룹에 문제가 있을지도 모른다.

세부 정보

보안

네트워킹

스토리지

상태 검사

모니터링

태그

▼ 보안 세부 정보

IAM 역할

-

보안 그룹

sg-0e3a402acae5aabb2 (launch-wizard-2)

소유자 ID

985893411276

▼ 인바운드 규칙

Q 필터 규칙

보안 그룹 규칙 ID	포트 범위	프로토콜	원본	보안 그룹
sgr-031ce3b04e8272921	443	TCP	0.0.0.0/0	launch-wizard-2
sgr-09ea3ef0948d7170f	80	TCP	0.0.0.0/0	launch-wizard-2
sgr-0a6d96013c5ecce94	80	TCP	::/0	launch-wizard-2
sgr-024080d2e6dd10760	22	TCP	0.0.0.0/0	launch-wizard-2

EC2 > 보안 그룹 > sg-0e3a402acae5aabb2 - launch-wizard-2

sg-0e3a402acae5aabb2 - launch-wizard-2

작업 ▼

세부 정보

보안 그룹 이름

launch-wizard-2

보안 그룹 ID

sg-0e3a402acae5aabb2

설명

launch-wizard-2 created 2021-11-14T20:13:03.433+09:00

VPC ID

vpc-099f9adb67974c27c

소유자

985893411276

인바운드 규칙 수

4 권한 항목

아웃바운드 규칙 수

1 권한 항목

인바운드 규칙

아웃바운드 규칙

태그

이제 Reachability Analyzer를 사용하여 네트워크 연결을 확인할 수 있습니다.

Reachability Analyzer 실행

인바운드 규칙 (4)

태그 관리

인바운드 규칙 편집

Q 보안 그룹 규칙 필터

< 1 >

	Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위	소스	설명
<input type="checkbox"/>	-	sgr-031ce3b04e82729...	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-09ea3ef0948d7170f	IPv4	HTTP	TCP	80	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0a6d96013c5ecce94	IPv6	HTTP	TCP	80	::/0	-
<input type="checkbox"/>	-	sgr-024080d2e6dd10...	IPv4	SSH	TCP	22	0.0.0.0/0	-

상기 캡처와 같이 보안 그룹 규칙을 모두 다시 확인하였으며 인바운드 규칙에 특이사항은 없는 것으로 확인되었으나 여전히 서버 접속 오류의 원인을 찾을 수 없었고 더 이상 과제 진행이 불가하였다.