Port mapping

Port mapping in Docker is the process of assigning a port on the host machine to inside a container application, to access the application from outside of docker.

- By default, the container's internal port is hidden inside Docker.

Why Port Mapping

- Containers are isolated Each Docker container runs its own network environment.
- Without port mapping, container apps cannot be accessed from outside Docker.
- This allows external access (like from browsers) to the container app.

How does the traffic flow from outside to inside the container via port mapping

External request:
You access the host machine's port, for example, http://localhost:8080.

Host machine:
Your host machine listens on port 8080.
Docker port mapping:
Docker forwards any traffic coming to host port 8080 to the container's internal port, for example, port 80.

Inside the container:
The application inside the container listens on port 80 and responds.
Response sent back:
The response travels back the same way from the container's port 80 → host port 8080 → your browser or external system.

In Docker, there are two main types of port mapping:

1. Port Forwarding (-P)

- Docker automatically assigns a random port on your host machine to the container's exposed port.
- You use the -P (capital P) option.
- Port Forwarding is also known as Dynamic port forwarding.

Example:

```
docker run -P nginx
```

- After running the container, use this command to hecck the random port is assigned or not.

```
docker port nginx
```

- Now we can see that the port like Example :

```
80/tcp -> 0.0.0.0:49153
```

➡ This means:

- The container's port 80 is mapped to host port 49153.
- You can access the app using: http://localhost:49153

Alternative: Use docker ps

```
docker ps
```

Look in the "PORTS" column. It will show something like:

```
0.0.0.0:49153->80/tcp
```

✅ **Use When:**

- You don't care which host port is used.
- You're testing or running multiple containers without port conflicts.

2. Port Assigning (-p)

- You manually specify which host port should connect to which container port (will assign a custom port).
- Use the -p (small p) option.

Example:

```
docker run -p 8080:80 nginx
```

- Now we can the port using

```
docker port nginx
```

(or)

```
docker ps
```

Output :

```
80/tcp -> 0.0.0.0:8080
```

- This maps host port 8080 → container port 80.

✅ **Use When:**

- You want predictable, fixed ports (e.g., in production).

LAB :

create a custom Apache HTTP Server (httpd) image, assign a port, and deploy a colorful HTML page :

🔧 Step-by-Step Guide :

step 1: Create Your Project Structure

```
mkdir Basic-Docker-Project
cd Basic-Docker-Project
```

```
vi index.html
```

- In the index.html file we can paste a basic html code, you can refer this code.

```
<!DOCTYPE html>
<html> <head>

    <title>Colorful Apache Page</title>
    <style>
        body {
            background: linear-gradient(to right, #ff758c, #ff7eb3);
            color: white;
            font-family: Arial, sans-serif;
```

```
            text-align: center;
            padding-top: 100px;
        }
        h1 {
            font-size: 3rem;
            text-shadow: 2px 2px #000;
        }
        p {
            font-size: 1.2rem;
        }
    </style>
</head>
<body>
    <h1>Keep Learning Keep Growing</h1>
    <p>Happy Learning</p>
</body>
</html>
```

step2: Run Apache container with your HTML mounted and port mapped

```
docker run -d -p 8080:80 -v $(pwd):/usr/local/apache2/htdocs/ --name my-httpd-container httpd
```

- -d → run container detached (in background)
- -p 8080:80 → map your machine's port 8080 to the container's port 80 (Apache's default)
- -v $(pwd):/usr/local/apache2/htdocs/ → mount your current folder (my-httpd-site) as the web root in the container
- httpd → official Apache HTTP Server image.

Step 3: Open your browser

```
http://localhost:8080
```

Check the port using :

```
docker ps (or) docker port httpd
```

Step-By-Step process:

```
ubuntu@ip-172-31-25-247:~$ mkdir Basic-Docker-Project
ubuntu@ip-172-31-25-247:~$ ls
Basic-Docker-Project
ubuntu@ip-172-31-25-247:~$
ubuntu@ip-172-31-25-247:~$
ubuntu@ip-172-31-25-247:~$ cd Basic-Docker-Project/
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$ vi index.html
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$ vi index.html
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$ docker run -d -p 8080:80 -v $(pwd):/usr/local/apache2/htdocs/ --name my-httpd-container httpd
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping":
dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$ sudo chmod 666 /var/run/docker.sock
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$ docker run -d -p 8080:80 -v $(pwd):/usr/local/apache2/htdocs/ --name my-httpd-container httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
254e724d7786: Pull complete
10d01782dc02: Pull complete
4f4fb700ef54: Pull complete
4ceeea7b3d76: Pull complete
0ff470512d2f: Pull complete
ba78a05e3b3c: Pull complete
Digest: sha256:c11efd67f6308f2c25965e4e9d13ded15e7c45c0367b95f619a16e03c6c1e2b1
Status: Downloaded newer image for httpd:latest
a14ae48bced18b4f009e40326444386a9de3d809cc8ea456831a3c782a94361e
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$ docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED         STATUS         PORTS                                   NAMES
a14ae48bced1   httpd     "httpd-foreground"   6 minutes ago   Up 6 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   my-httpd-container
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$ docker port my-httpd-container
80/tcp -> 0.0.0.0:8080
80/tcp -> [::]:8080
ubuntu@ip-172-31-25-247:~/Basic-Docker-Project$
```

- Index.html file which conatins html-code

```html
<!DOCTYPE html>
<html>
<head>
    <title>Keep Learning Keep Growing</title>
    <style>
        body {
            background: linear-gradient(to right, #ff758c, #ff7eb3);
            color: white;
            font-family: Arial, sans-serif;
            text-align: center;
            padding-top: 100px;
        }
        h1 {
            font-size: 3rem;
            text-shadow: 2px 2px #000;
        }
        p {
            font-size: 1.2rem;
        }
    </style>
</head>
<body>
    <h1>Keep Learning Keep Growing</h1>
    <p>Happy Learning</p>
</body>
</html>
```

- After Running a container we can access the application using Ip-address, we can this output.