After running docker-compose up –build, we can output the result by creating the curl request to http://localhost:8001/

```
risa@Risas-MBP ~ % curl http://localhost:8001/
Hello from 192.168.48.1:61500 to 192.168.48.3:8001
Hello from 192.168.48.3:53184 to 192.168.48.2:8002
```

Docker services run on local computers, so they share the same internal network environment. The IP addresses can be different each time docker-compose build is run. Docker defined a network called backend so that two Docker services could communicate to each other.

- 192.168.48.1 IP address is a local, private, or gateway IP address. 192.168.48.1 is my router's address that computer devices connected to the network will use to send data requests over the internet. So when we curl to service1 from my computer, it will show the remote address for service1 is 192.168.48.1 and the port which is used to send curl request to service1.
- 192.168.48.3 is a service1's IP address, and the port is defined in the index.js file from service1 which is mapped from Docker to local host.

```
const http = require('http');
const IP = require('ip');

const ipAddress = IP.address();
const hostname = ipAddress;
const port = 8001;
```

- 192.168.48.2 is a service2's IP address, and the port is defined in the index.js file from service2 which is mapped from Docker to local host.

```
const http = require('http');
const IP = require('ip');

const ipAddress = IP.address();
const hostname = ipAddress;
const port = 8002;
```

To see all the containers, we can run docker container ls

```
risa@Risas-MBP docker-compose % docker container ls
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS          PORTS                    NAMES
83b797d375d4   docker-compose_service1   "docker-entrypoint.s…"   32 seconds ago   Up 31 seconds   0.0.0.0:8001->8001/tcp   service1
9f2186f2dce9   docker-compose_service2   "docker-entrypoint.s…"   32 seconds ago   Up 32 seconds                            service2
risa@Risas-MBP docker-compose %
```

To see all the networks, we can run docker network ls

```
risa@Risas-MBP docker-compose % docker network ls
NETWORK ID     NAME                      DRIVER    SCOPE
0a61e4e0bbec   bridge                    bridge    local
ffe87a1c40b2   docker-compose_backend    bridge    local
be30f6ccf10e   host                      host      local
c4c85ddd1073   none                      null      local
risa@Risas-MBP docker-compose %
```