Final output from -> curl localhost:8001

Hello from 172.19.0.1:55724
to 172.19.0.2:3000
Hello from 172.18.0.3:57712
to 172.18.0.2:3000

Explanation why the addresses and port-numbers are like they are

Mapping of IP addresses,

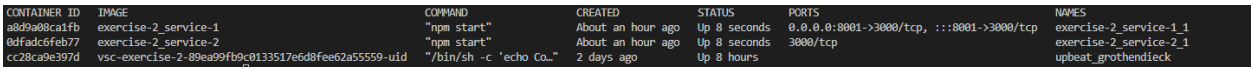
Component	Network 1 (default in compose file) IP	Network 1 (private in compose file) IP
Host	172.19.0.1	172.18.0.1
Service-1	172.19.0.2	172.18.0.3
Service-2	Not present in the network	172.18.0.2

My first observation is that I do not see local port 8001 on service 1 output. The reason for that may be because I am creating a port mapping on docker-compose to expose service 1 on port 8001 while the container itself exposes traffic on port 3000.

My second observation is that the IP address subnet / range is different for both service 1 and service 2. The reason for that is, I created two networks in docker compose stack. One network which only has service 1 bridged to host and other has service 1 and service 2 in an isolated network. However, I do not know how to not have the second 'isolated' network to not have bridge to host.

The larger port numbers on remote sockets are outgoing ports which are selected by TCP stack for outgoing connections. Further calls to the API kept changing those ports as TCP does not immediately reuse the port numbers.

Output from -> docker container ls



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a8d9a08ca1fb	exercise-2_service-1	"npm start"	About an hour ago	Up 8 seconds	0.0.0.0:8001->3000/tcp, :::8001->3000/tcp	exercise-2_service-1_1
0dfadc6feb77	exercise-2_service-2	"npm start"	About an hour ago	Up 8 seconds	3000/tcp	exercise-2_service-2_1
cc28ca9e397d	vsc-exercise-2-89ea99fb9c0133517e6d8fee62a55559-uid	"/bin/sh -c 'echo Co…"	2 days ago	Up 8 hours		upbeat_grothendieck

The readable text was being scrambled here so the screenshot above, but to simplify, here is the output of -> docker container ls --format '{{.ID}} : {{.Image}} : {{.Command}} : {{.Ports}}'

a8d9a08ca1fb : exercise-2_service-1 : "npm start" : 0.0.0.0:8001->3000/tcp, :::8001->3000/tcp

0dfadc6feb77 : exercise-2_service-2 : "npm start" : 3000/tcp

cc28ca9e397d : vsc-exercise-2-89ea99fb9c0133517e6d8fee62a55559-uid : "/bin/sh -c 'echo Co…" :

Note: the last container is my docker dev container.

Output from -> docker network ls

NETWORK ID NAME DRIVER SCOPE

ecb54575de1c bridge bridge local

47dfc7d29f70 exercise-2_default bridge local

5763847e3030 exercise-2_private bridge local

2e9dbc3a70f4 host host local

e1f086c0389d none null local

Note: This may not be the output you expect, as my stack is,

Windows -> WSL (contains docker installation) -> Docker Dev Container (Node.js) -> docker-compose up –build
The outputs are from commands executed on WSL.