# Docker Networking - Using a Custom Bridge Network

## 1. Default Networks

Checked existing networks:

Bash

```
docker network ls
```

Output showed:

- `bridge` → Default bridge network.
- `host` → Shares host's networking stack.
- `none` → Isolated network (no connectivity).

 **By default**, containers on the **default bridge** cannot resolve each other by name, which makes microservice communication tricky.

## 2. Creating a Custom Bridge Network

Command:

Bash

```
docker network create currency-network
```

Result: A new user-defined bridge network `currency-network` was created.

Verified:

Bash

```
docker network ls
```

Output confirmed `currency-network` exists alongside default networks.

## 3. Running Containers on the Custom Network

Befor running these command we have to stop and remove both the container.

Currency Exchange service:

Bash

```
docker run -d -p 8000:8000
  --name=currency-exchange
  --network=currency-network
  souravdevopsdev/currency-exchange:0.0.1
```

Currency Conversion service:

Bash

```
docker run -d -p 8100:8100
  --env CURRENCY_EXCHANGE_SERVICE_HOST=http://currency-exchange
  --name=currency-conversion
  --network=currency-network
  souravdevopsdev/currency-conversion:0.0.1
```

### Key Points

i. Both containers are attached to the same **user-defined bridge network** (`currency-network`).

ii. In this network, containers can resolve each other by **container name** (e.g., `currency-exchange`).

iii. The environment variable `CURRENCY_EXCHANGE_SERVICE_HOST=http://currency-exchange` tells the conversion service how to reach the exchange service.

# 4. Verification

Checked running containers:

Bash

```
docker container ls
```

Output showed both services running:

`currency-exchange` on port 8000.

`currency-conversion` on port 8100.

Stopped containers when done:

bash

```
docker container stop d02dfda54a40 628b6613838b
```

# 5. Key Learnings

A. **Default bridge network**: Containers can't talk by name, only by IP (not ideal for microservices).

B. **User-defined bridge network**: Containers can communicate using their names, making service discovery easier.

C. **Environment variables**: Still needed to tell the application which service to call, but now the hostname (`currency-exchange`) works because of the custom network.

**Best practice**: Use user-defined networks instead of `--link` (which is legacy and deprecated). This approach is closer to how Kubernetes handles service discovery.