Create a Custom Docker Bridge Network

Step 1: Verify Existing Docker Networks

Check what networks already exist:

docker network ls

You will see something like:

NETWORK ID NAME DRIVER SCOPE

abcd1234efgh bridge bridge local

ijkl5678mnop host host local

qrst9876uvwx none null local

Step 2: Create a Custom Bridge Network

docker network create \

--driver bridge \

--subnet 192.168.100.0/24 \

--gateway 192.168.100.1 \

custom-bridge-net

Explanation:

- --driver bridge → Specifies it as a bridge network
- --subnet → Defines a custom IP range for containers
- --gateway → The gateway IP address inside the network
- custom-bridge-net → Your custom network name

Step 3: Inspect the Custom Network

docker network inspect custom-bridge-net

This gives detailed info about your network, including its name, IPAM config, connected containers, etc.

Step 4: Run Containers Using the Custom Network

You can run containers and attach them to this network using --network:

docker run -dit --name container1 --network custom-bridge-net alpine sh

docker run -dit --name container2 --network custom-bridge-net alpine sh

Now, both container1 and container2 are on the same custom network.

Step 5: Test Communication Between Containers

Exec into one container and ping the other by its name:

docker exec -it container1 sh

ping container2

You should see replies, meaning the containers are communicating over your custom network.

Step 6: Remove Containers and Network (Optional)

To stop and remove containers:

docker rm -f container1 container2

To delete the custom bridge network:

docker network rm custom-bridge-net

Summary

Task	**Command Example**
List Docker networks	docker network ls
Create a custom bridge network	docker network create --driver bridge --subnet 192.168.100.0/24 --gateway 192.168.100.1 custom-bridge-net
Inspect network	docker network inspect custom-bridge-net
Run container on custom network	docker run -dit --name container --network custom-bridge-net alpine sh
Ping another container	ping container_name
Remove container	docker rm -f container_name
Remove custom network	docker network rm custom-bridge-net