A concise guide to manage Docker networking — suitable for local setups and project containers.

## 🧱 Basic Network Commands

Command	Description
`docker network ls`	List all Docker networks
`docker network inspect <network-name>`	Inspect details of a network
`docker network create <network-name>`	Create a custom bridge network
`docker network rm <network-name>`	Remove a Docker network

🔌 Create Custom Networks

```
# Bridge (default type)

docker network create --driver bridge my_bridge_net

# Host network (shares host's network)

docker network create --driver host my_host_net
```

🔗 Connect & Disconnect Containers

```
# Connect container to a specific network

docker network connect <network-name> <container-name>

# Disconnect container from a network

docker network disconnect <network-name> <container-name>
```

🧪 Example: Using a Custom Bridge Network

```
# Step 1: Create network

docker network create my_net

# Step 2: Run MySQL container

docker run -d --name mysql_container --network my_net mysql

# Step 3: Run WordPress container on same network

docker run -d --name wp_container --network my_net wordpress
```

🌐 Inspect Network Details

```
docker network inspect my_net
```

📍 Start Container with Specific Network

```
docker run -d --name nginx_app --network my_net nginx
```

🔍 Get Container IP

```
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' <container-name>
```

🧰 Advanced: Custom Subnet and Gateway

```
docker network create \
  --driver=bridge \
  --subnet=192.168.100.0/24 \
  --gateway=192.168.100.1 \
  my_custom_net
```

🧹 Remove Unused Networks

```
docker network prune
```

🧪 Debug Networking

```
docker exec -it <container> ping <other-container-name>

docker exec -it <container> curl <other-container-name>:<port>
```