## **🚢 Docker Network Commands for Bridge Driver**

### **1️⃣ List all networks**

docker network ls

👉 Shows all networks, including bridge, overlay, host, etc.

### **2️⃣ Inspect a specific bridge network**

docker network inspect bridge

👉 Details about the default bridge network or any other user-defined bridge network.

### **3️⃣ Create a user-defined bridge network**

docker network create --driver bridge my-bridge-network

👉 Creates a new bridge network with a custom name.

**Optional flags**:

* --subnet (specify a custom subnet)
* --gateway (specify a custom gateway)
* --ip-range (limit IP allocation)

Example:

docker network create --driver bridge --subnet 172.25.0.0/16 my-bridge-network

### **4️⃣ Connect a container to a bridge network**

docker network connect my-bridge-network my-container

👉 Attaches an existing container to the bridge network.

### **5️⃣ Disconnect a container from a bridge network**

docker network disconnect my-bridge-network my-container

👉 Detaches a container from the bridge network.

### **6️⃣ Remove a bridge network**

docker network rm my-bridge-network

👉 Deletes a user-defined bridge network (must be unused by containers).

### **7️⃣ Prune unused networks**

docker network prune

👉 Removes all unused bridge (and other driver) networks.

### **8️⃣ Run a container with a specific bridge network**

docker run -d --name my-container --network my-bridge-network nginx

👉 Starts a container attached to the specified bridge network.

### **9️⃣ List all bridge networks (filtered)**

docker network ls --filter driver=bridge

👉 Shows only bridge networks.

### **1️⃣0️⃣ Check container IP address within a bridge network**

docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' my-container

👉 Retrieves the container’s IP address in the bridge network.