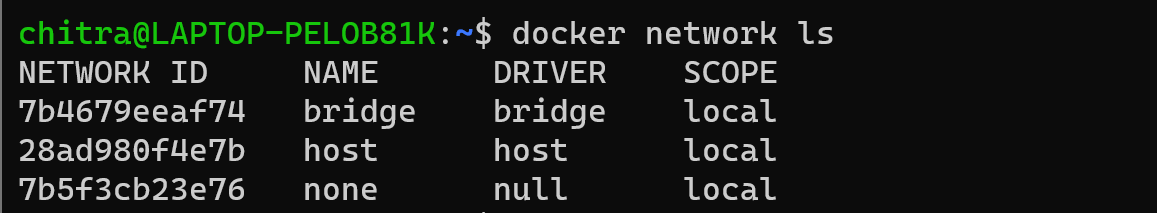
**EXPERIMENT -5: Docker Networking**

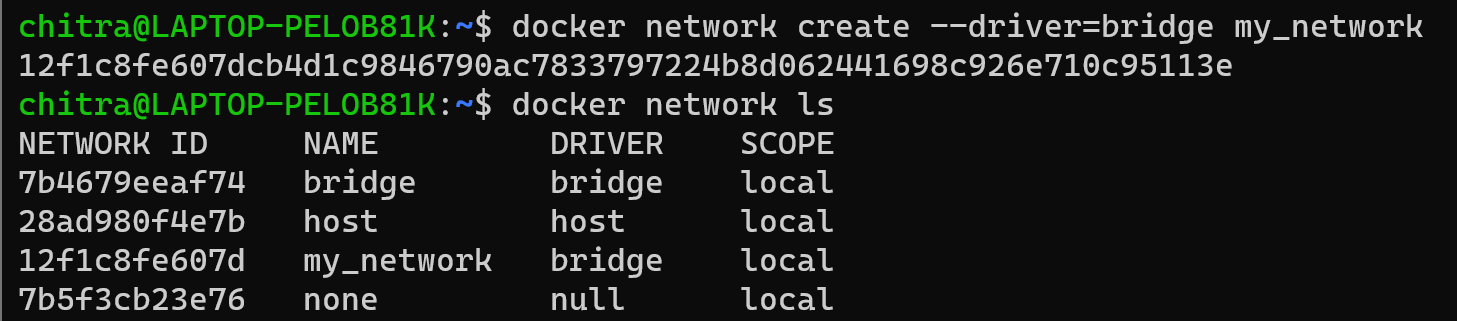
**Step-1:** Let’s first see the available network using the command:

docker network ls



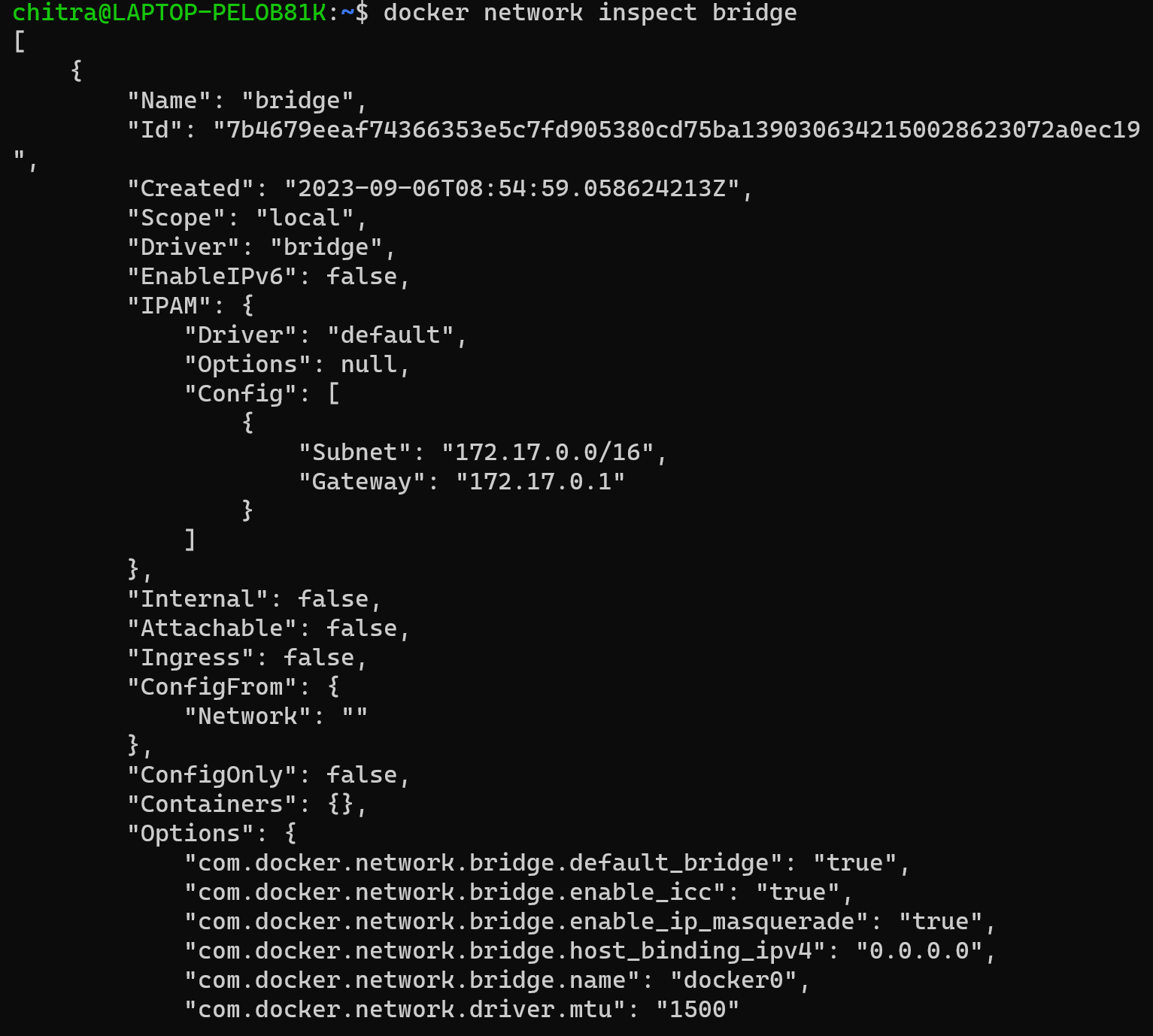
**Step-2:** Now let’s create a custom network of type bridge.

docker network create --driver=bridge my\_network



**Step-3:** Next let’s retrieve the information of the bridge network driver using the command:

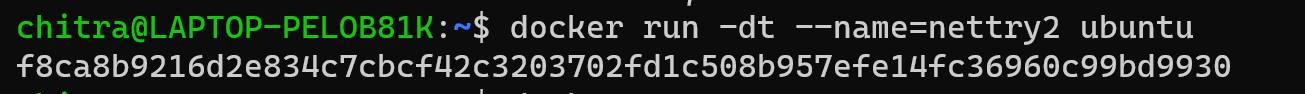
docker network inspect bridge



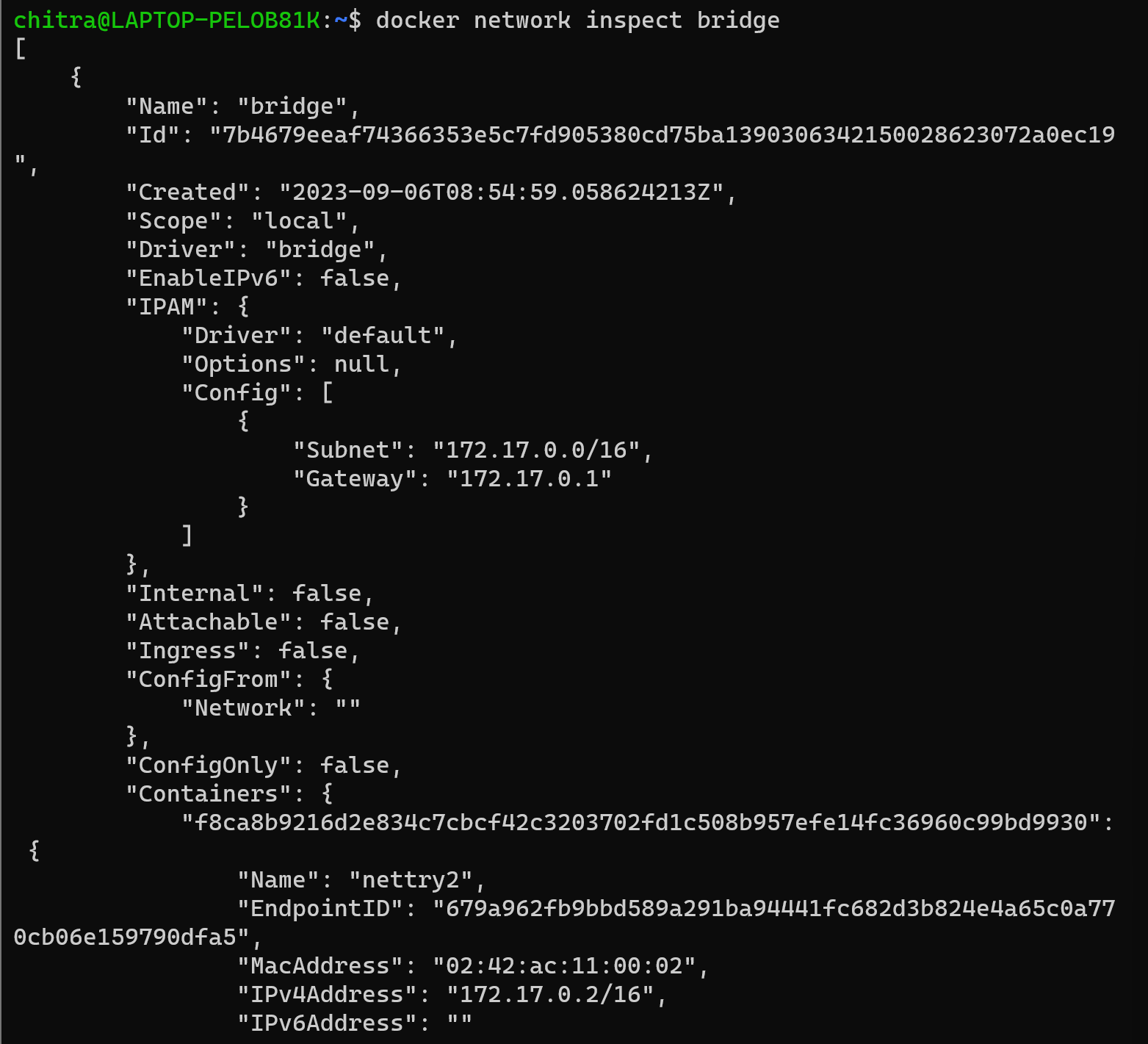
We can see in the “Containers” part that it is empty, it means that there are no containers associated to it till now. So firstly let’s create a new container.

**Step-4:** Here we are creating a new container of ubuntu image without specifying any network details.

docker run -dt --name=nettry2 ubuntu



**Step-5:** Now let’s again inspect the bridge network.

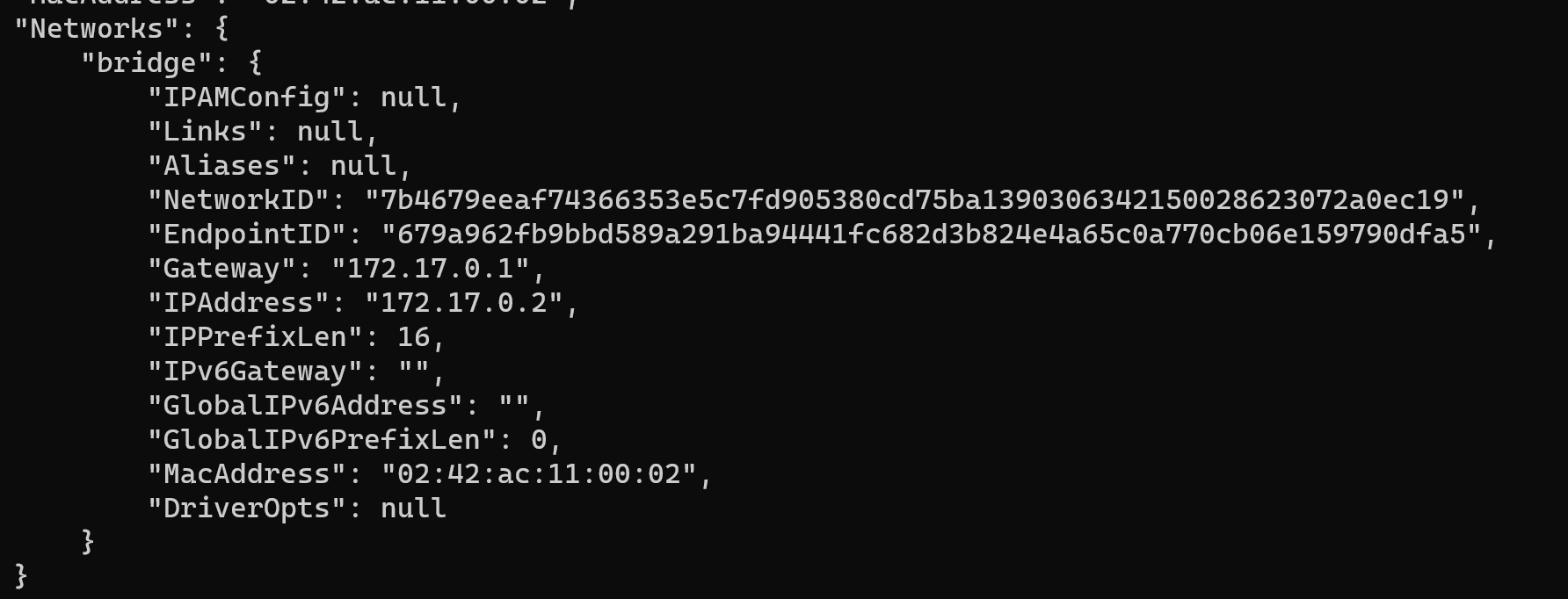


Here we can see that the container named “nettry2” is by default associated with this default bridge network.

**Step-6:** Now, check the network details of the container by inspecting it.

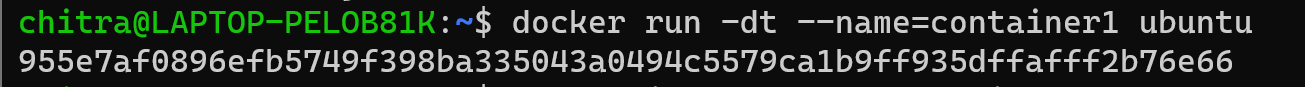


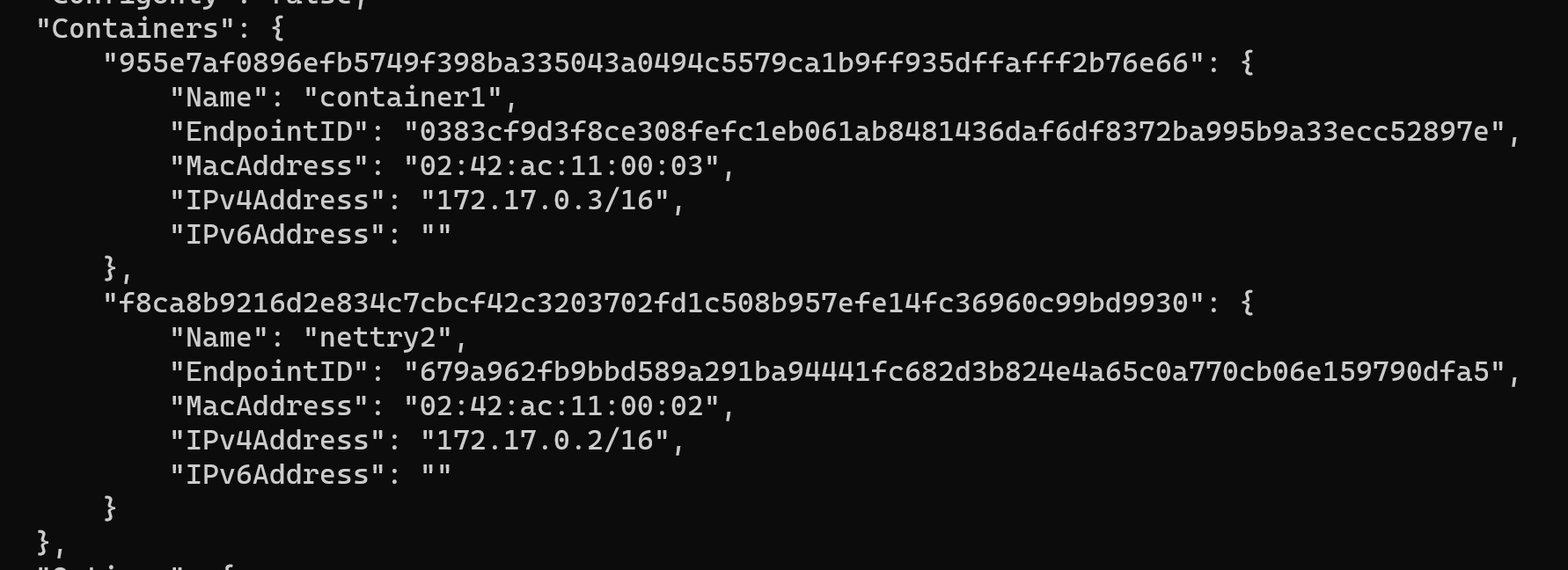
**Step-7:** We can see that a default IP address is assigned to the container from the bridge network.



**Step-8:** Now run another container and check whether inspect the bridge network again whether the second container is associated with it or not.

docker run -dt --name=container1 ubuntu

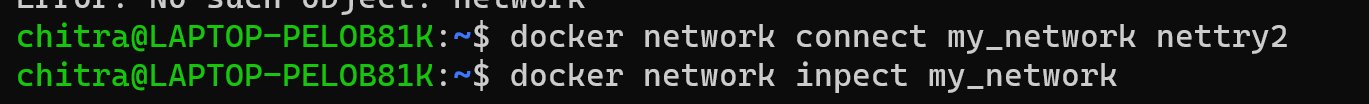




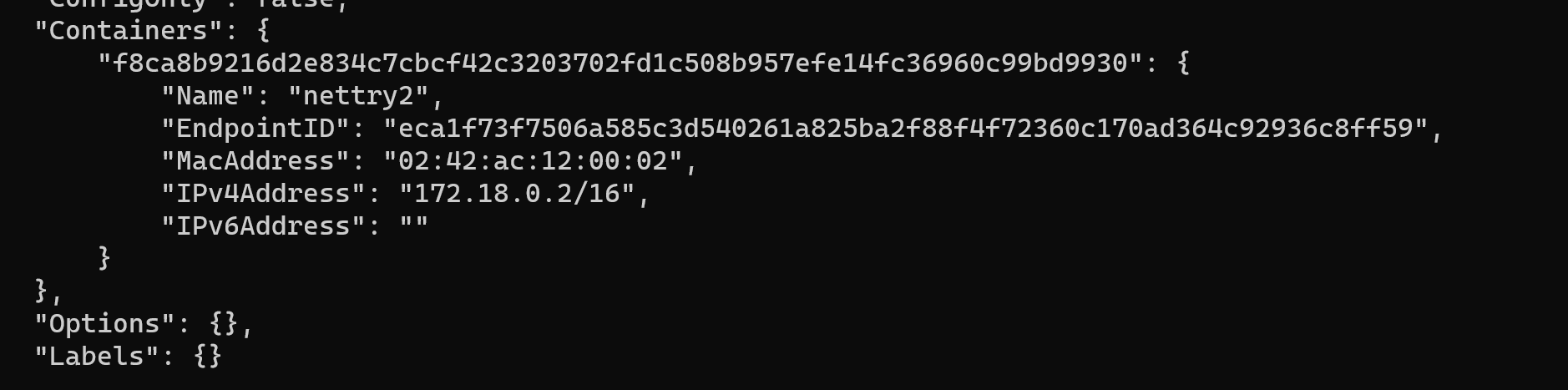
Here, we can clearly see that the two container that we just made are by default connected to the bridge network.

**Step-9:** Now, connect to the custom network, “my\_network” that we created in step one using the command and then inspect.

docker network connect my\_network nettry2



**Step-10:** We can observe here that the container “nettry2” is successfully connected to the new custom network and is assigned a different IP for this network.

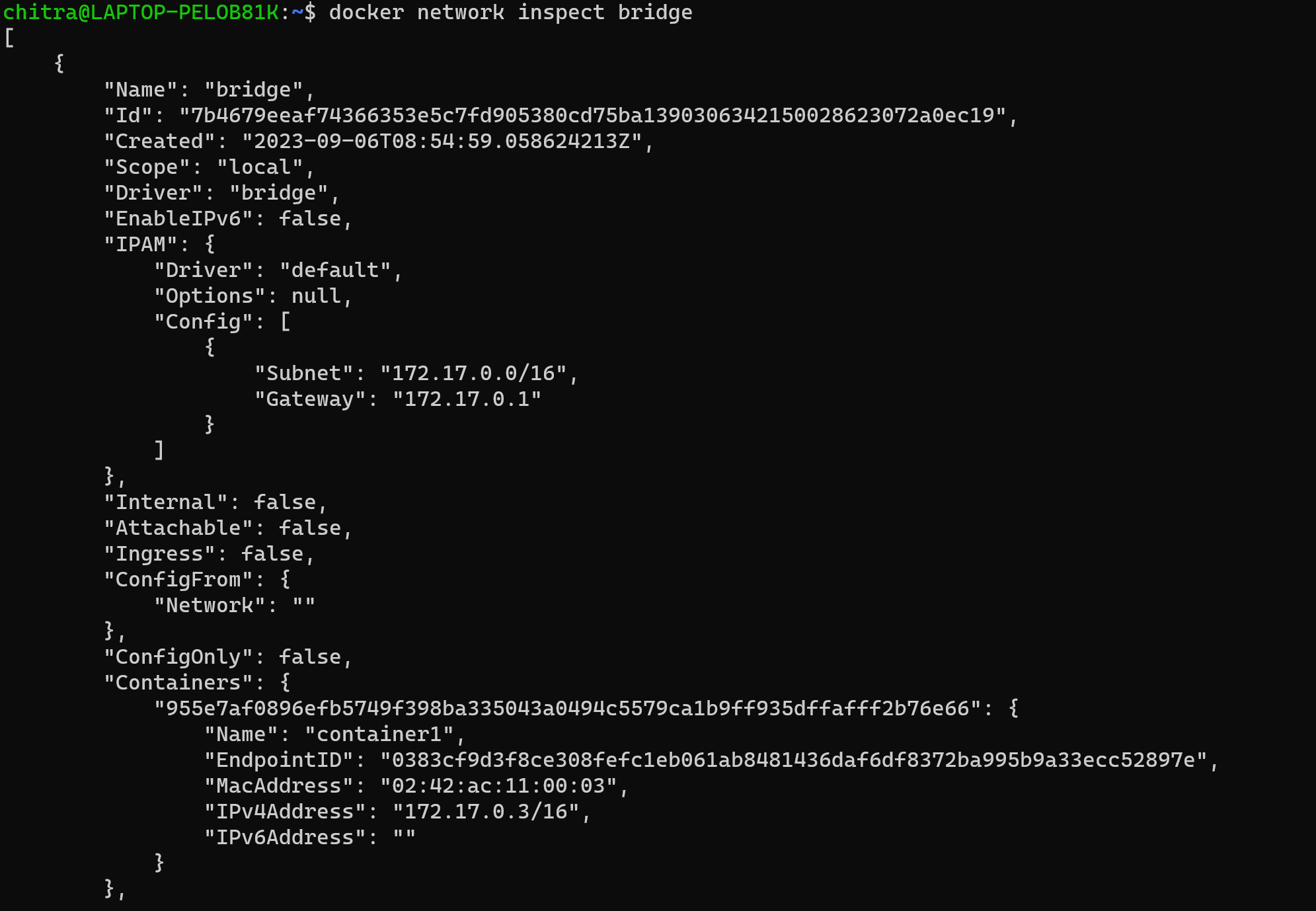


**Step-11:** We can disconnect a container from any network by using the following command

docker network disconnect bridge nettry2

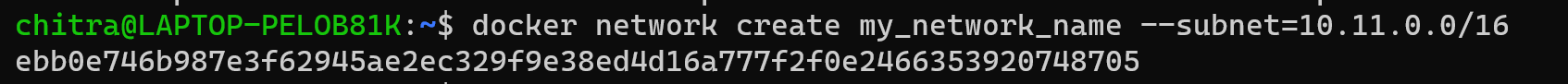


**Step-12:** Here we can see that the container is successfully disconnected from the default bridge network.



**Step-13:** Next, create a Docker network with a custom subnet range which can be further used to assign static IPs to the containers.

docker network create my\_network\_name --subnet=10.11.0.0/16



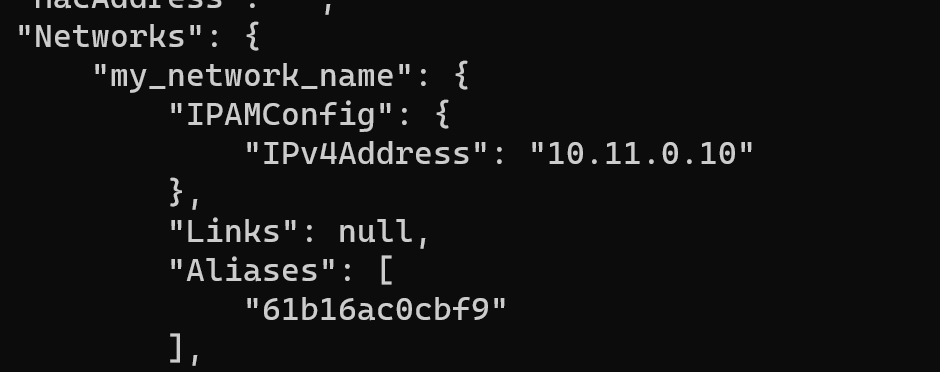
**Step-14:** Now let’s run a container and connect it to a custom network with a static IP address using the ubuntu image.

docker run -it --name=newcon --net=my\_network\_name --ip=10.11.0.10 ubuntu

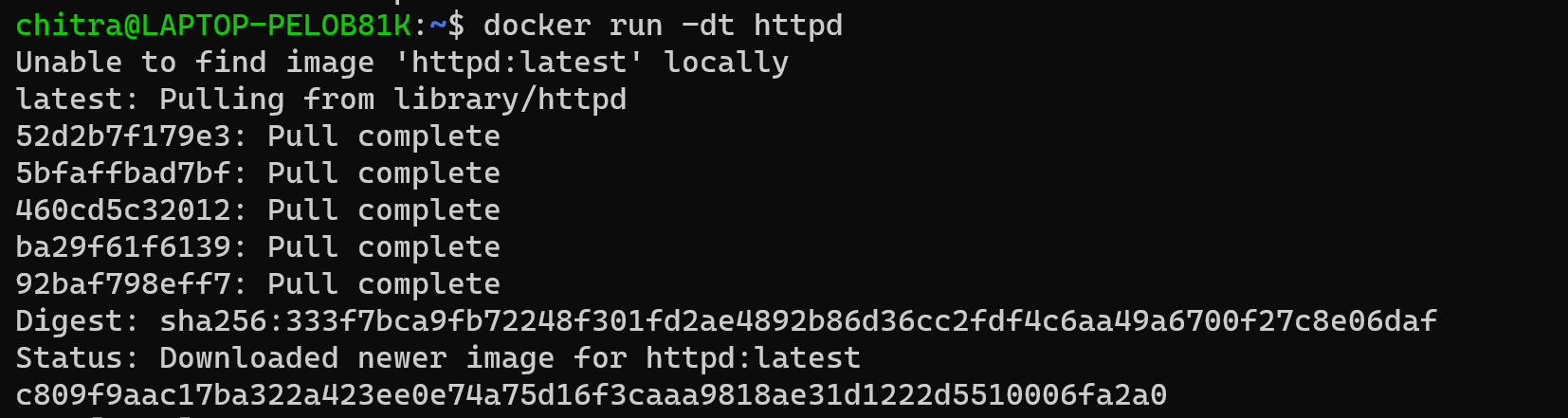


**Step-15:** By running the inspect command, we can see that the newcon container that we just created is now associated with “my\_network\_name” and has the IP assigned to it that we specified.





**Step-16:** Now let’s try to create a container of apache httpd image without specifying its name.



**Step-17:** Here we can see that it is given a default name.

