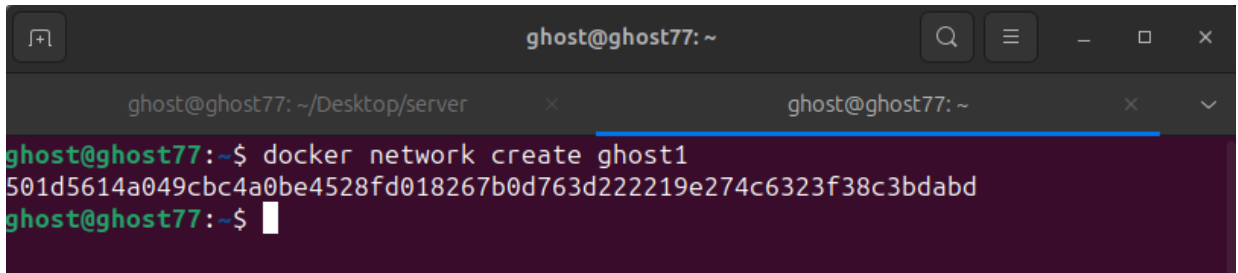


6. Create a Docker Container Network

To create a Docker container network, you can use the Docker CLI or Docker Compose. I'll provide instructions for creating a bridge network using the Docker CLI:

1. Open a terminal or command prompt.
2. Run the following command to create a Docker bridge network:

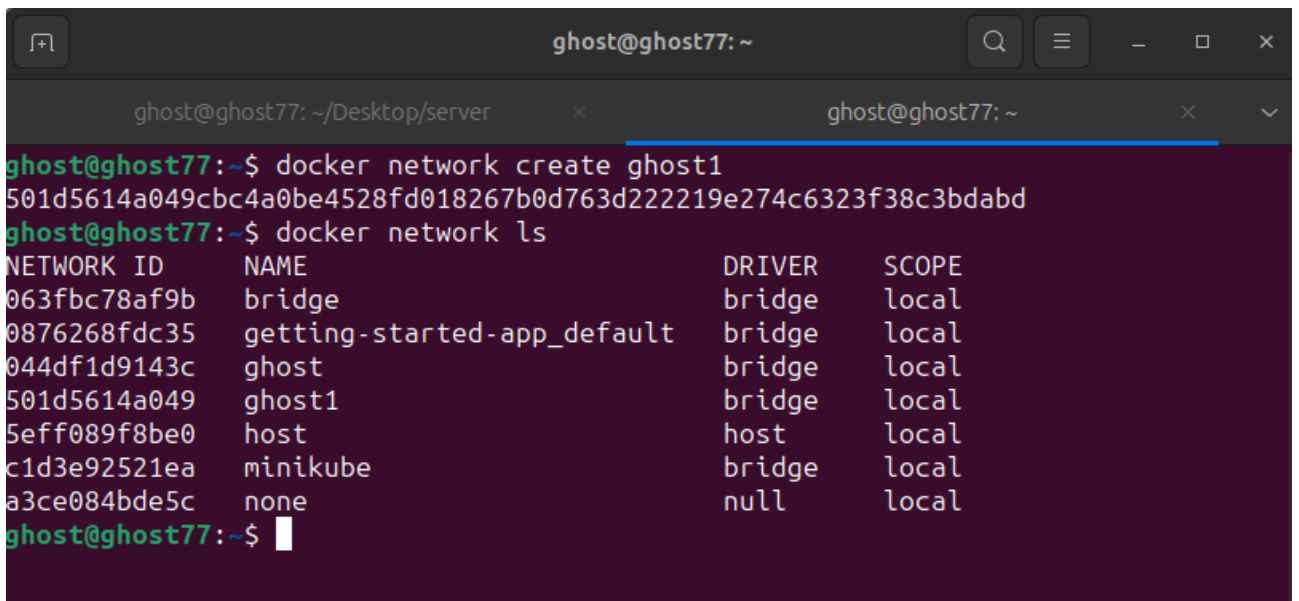
A terminal window titled 'ghost@ghost77: ~' with two tabs. The active tab shows the command 'docker network create ghost1' and its output, a long hexadecimal ID: '501d5614a049cbc4a0be4528fd018267b0d763d222219e274c6323f38c3bdabd'.

```
ghost@ghost77: ~$ docker network create ghost1
501d5614a049cbc4a0be4528fd018267b0d763d222219e274c6323f38c3bdabd
ghost@ghost77: ~$
```

Replace `your-network-name` with the desired name for your network. For example, you can use a name like `my-network`.

By default, the bridge network driver is used, which allows containers to communicate with each other on the same host.

- Verify that the network has been created by running the following command:

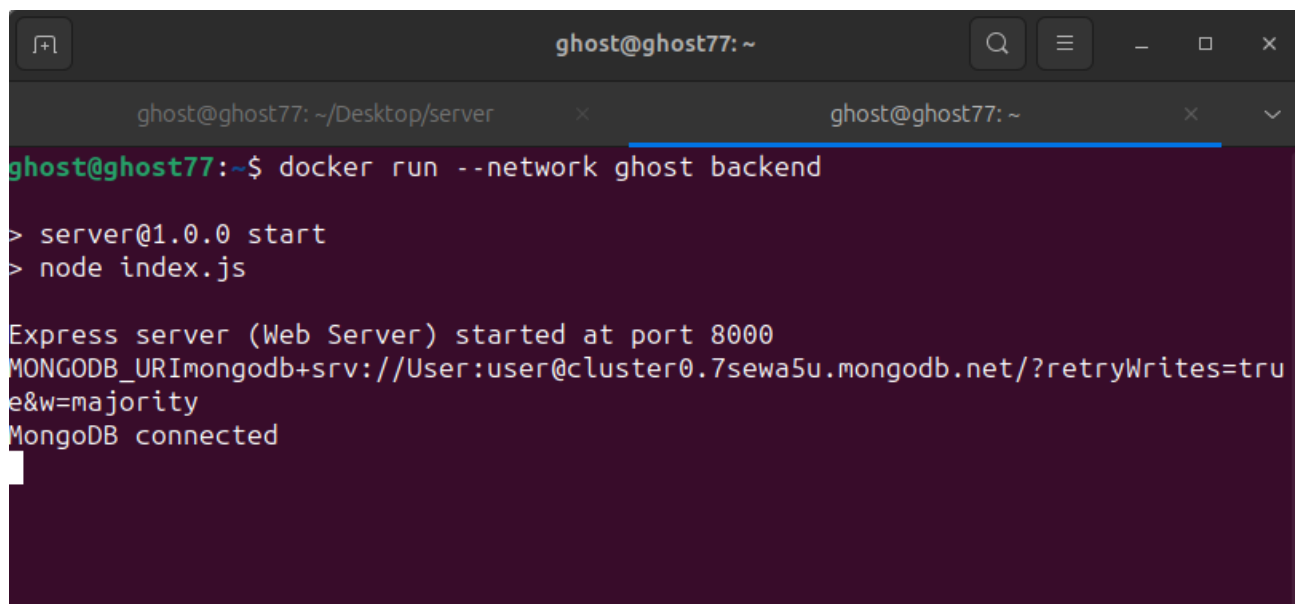
A terminal window titled 'ghost@ghost77: ~' with two tabs. The active tab shows the command 'docker network ls' and its output, a table listing all Docker networks on the system.

```
ghost@ghost77: ~$ docker network create ghost1
501d5614a049cbc4a0be4528fd018267b0d763d222219e274c6323f38c3bdabd
ghost@ghost77: ~$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
063fbc78af9b        bridge              bridge              local
0876268fdc35        getting-started-app_default  bridge              local
044df1d9143c        ghost               bridge              local
501d5614a049        ghost1              bridge              local
5eff089f8be0        host                host                local
c1d3e92521ea        minikube             bridge              local
a3ce084bde5c        none                null                local
ghost@ghost77: ~$
```

This command lists all the Docker networks on your system.

You should see your newly created network in the list along with its name, ID, and other details.

To use this network when running Docker containers, specify the network name with the `--network` option. For example:

A terminal window titled 'ghost@ghost77: ~' with two tabs. The active tab shows the command 'docker run --network ghost backend' being executed. The output shows a web server starting on port 8000 and connecting to a MongoDB database. The terminal has a dark purple background with white and green text.

```
ghost@ghost77:~$ docker run --network ghost backend
> server@1.0.0 start
> node index.js

Express server (Web Server) started at port 8000
MONGODB_URImongodb+srv://User:user@cluster0.7sewa5u.mongodb.net/?retryWrites=tru
e&w=majority
MongoDB connected
```

Replace your `-network-name` with the name of your network and your `-image-name` with the name of the Docker image you want to run.

By creating a Docker container network, you enable communication between containers within the network, allowing them to interact and exchange data efficiently.