

CO515 Advanced Computer Networks: Selected Topics

Docker Networking Lab 02

Overlay Networking

An “overlay network” is a virtual network that runs on top of a different network. Devices in that network are unaware that they are in an overlay. Traditional VPNs, for instance, are overlay networks running over the Internet.

The overlay network driver creates a distributed network among multiple Docker daemon hosts. This network sits on top of (overlays) the host-specific networks, allowing containers connected to it (including swarm service containers) to communicate securely when encryption is enabled.

Lab Task 01

1. In this step you'll initialize a new Swarm, join a single worker node, and verify the operations worked.

- 1.1. Run `docker swarm init --advertise-addr $(hostname -i)`
- 1.2. In the first terminal copy the entire `docker swarm join ...` command that is displayed as part of the output from your terminal output. Then, paste the copied command into the second terminal.
- 1.3. Run a `docker node ls` to verify that both nodes are part of the Swarm. Check whether both nodes have joined the Swarm and are ready and active.

2. Create an overlay network

- 2.1. Create a new overlay network called “overnet” by running `docker network create -d overlay overnet`
- 2.2. Use the `docker network ls` command to verify the network was created successfully.
Notice how the new “overnet” network is associated with the overlay driver and is scoped to the entire Swarm.
NOTE: The other new networks (ingress and docker_gwbridge) were created automatically when the Swarm cluster was created.
- 2.3. Run the same `docker network ls` command from the second terminal.
- 2.4. Use the `docker network inspect <network>` command to view more detailed information about the “overnet” network. Run this command from the first terminal.

3. Create a service

Now that we have a Swarm initialized and an overlay network, it's time to create a service that uses the network.

- 3.1. Execute the following command from the first terminal to create a new service called myservice on the overnet network with two tasks/replicas.

```
docker service create --name myservice \  
--network overnet \  
--replicas 2 \ ubuntu  
sleep infinity
```

- 3.2. Verify that the service is created and both replicas are up by running `docker service ls`. Examine the output.
- 3.3. Verify that a single task (replica) is running on each of the two nodes in the Swarm by running `docker service ps myservice`.
- 3.4. Now that the second node is running a task on the "overnet" network it will be able to see the "overnet" network. Lets run `docker network ls` from the second terminal to verify this.
- 3.5. Use `docker network inspect overnet` on the second terminal to get more detailed information about the "overnet" network and obtain the IP address of the task running on the second terminal.

4. Test the network

To complete this step you will need the IP address of the service task running on node2 that you saw in the previous step

- 4.1. Execute the `docker network inspect overnet` command from the first terminal.
- 4.2. Log on to the service task. Run `docker exec -it <CONTAINER ID> /bin/bash`
- 4.3. Install the ping command and ping the service task running on the second node used in step 3 above from the `docker network inspect overnet` command.
- 4.4. Ping to the ip address obtained in step 3 above

5. Test service discovery

Now that you have a working service using an overlay network, let's test service discovery.

- 5.1. If you are not still inside of the container, log back into it with the `docker exec -it <CONTAINER ID> /bin/bash` command.
Run `cat /etc/resolv.conf` from inside of the container.
- 5.2. Try and ping the "myservice" name from within the container by running `ping -c5 myservice`
Obtain the returning ip address.
Next let's verify that this address is the virtual IP (VIP) assigned to the `myservice` service.
- 5.3. Type the `exit` command to leave the `exec` container session and return to the shell prompt of your Docker host.

- 5.4. Inspect the configuration of the “myservice” service by running `docker service inspect myservice`. Lets verify that the VIP value matches the value returned by the previous `ping -c5 myservice` command.
- 5.5. Towards the bottom of the output you will see the VIP of the service listed. Does the VIP listed here match the value returned by the `ping -c5 myservice` command.?

Cleaning Up

Lets clean up the service we created, the containers we started, and finally disable Swarm mode.

1. Execute the `docker service rm myservice` command to remove the service called myservice.
2. Execute the `docker ps` command to get a list of running containers.
3. You can use the `docker kill <CONTAINER ID ...>` command to kill the ubuntu and nginx containers we started at the beginning.
4. Finally, remove node1 and node2 from the Swarm. Use the `docker swarm leave --force` command to do that. Lets run `docker swarm leave --force` on node1.
5. Run `docker swarm leave --force` on node2.

Now the service is cleaned up.

Submission

Submit a report named Lab02_EYYXXX.pdf (XXX is your E Number, YY is your batch) with the screenshots and explanations necessary for all the questions in the lab tasks.