Docker Swarm Networking and Stack

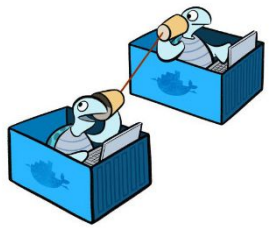
Table of Contents

1

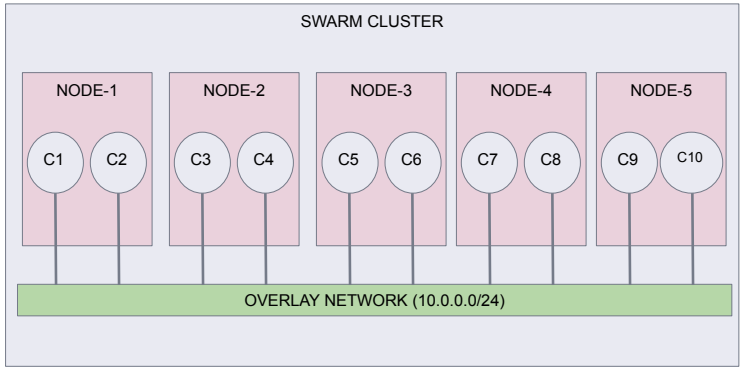
Overlay Network

Overlay Network

- **Overlay** networks connect multiple Docker daemons together and enable swarm services to communicate with each other.
- You can also use overlay networks to facilitate communication between a swarm service and a standalone container, or between two standalone containers on different Docker daemons.

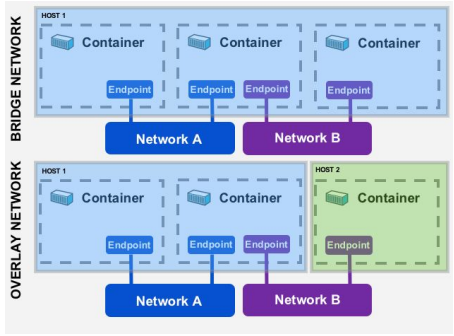


Overlay Network



Overlay Network

The **overlay** network driver creates a distributed network among multiple Docker daemon hosts.

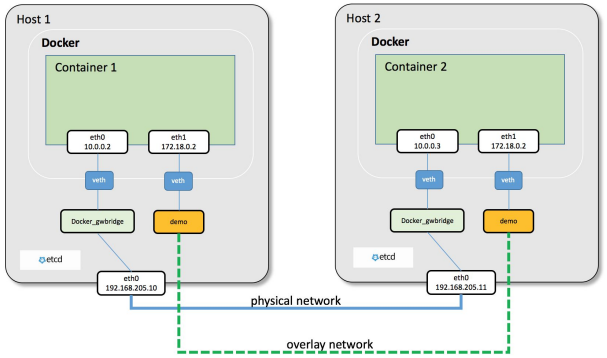


Overlay Network

When we initialize a swarm or join a Docker host to an existing swarm, two new networks are created on that Docker host:

- An overlay network called **ingress**, which handles control and data traffic related to swarm services. When you create a swarm service and do not connect it to a user-defined overlay network, it connects to the **ingress** network by default.
- A bridge network called **docker_gwbridge**, which connects the individual Docker daemon to the other daemons participating in the swarm.

Overlay Network



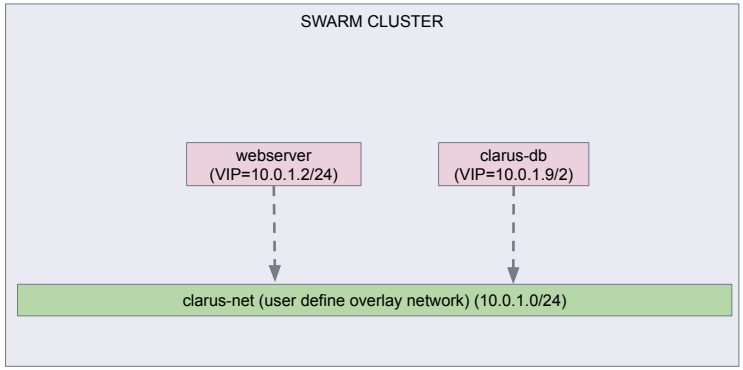
Overlay Network

Firewall rules for Docker daemons using overlay networks:

We need the following ports open to traffic to and from each Docker host participating on an overlay network:

- TCP port 2377 for cluster management communications
- TCP and UDP port 7946 for communication among nodes
- UDP port 4789 for overlay network traffic

User-defined Overlay Network



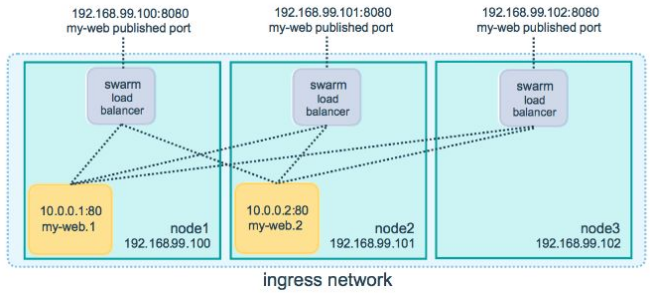
2

Swarm Mode Routing Mesh

Swarm Mode Routing Mesh

- Docker Engine swarm mode makes it easy to publish ports for services to make them available to resources outside the swarm.
- All nodes participate in an ingress routing mesh.
- The **routing mesh** enables each node in the swarm to accept connections on published ports for any service running in the swarm, even if there's no task running on the node.
- The routing mesh routes all incoming requests to published ports on available nodes to an active container.

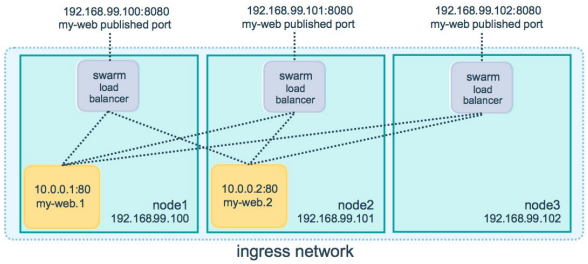
Swarm Mode Routing Mesh



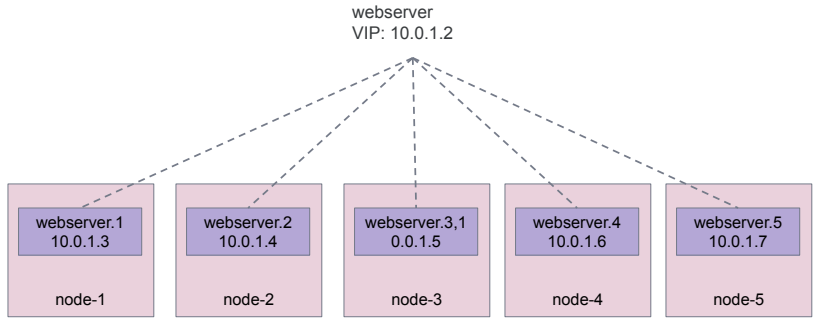
3 Swarm Load balancing

Swarm Load balancing

The swarm manager uses **ingress load balancing** to expose the services you want to make available externally to the swarm.



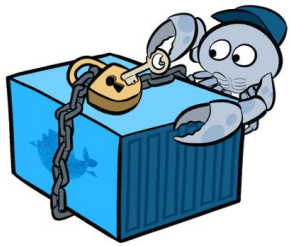
Swarm Load balancing



4 Docker secret

Docker secret

- In terms of Docker Swarm services, a **secret** is a blob of data, such as a **password, SSH private key, SSL certificate**, or another piece of data that should not be transmitted over a network or stored unencrypted in a Dockerfile or in your application's source code.



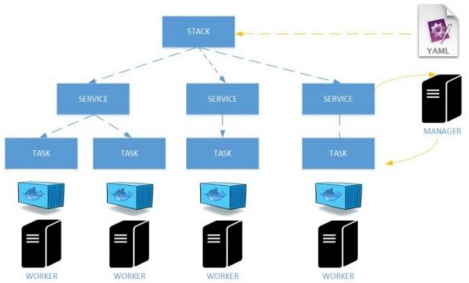
Docker secret

- You can use Docker secrets to centrally manage this data and securely transmit it to only those containers that need access to it.
- Secrets are **encrypted** during transit and at rest in a Docker swarm.

5 Docker stack

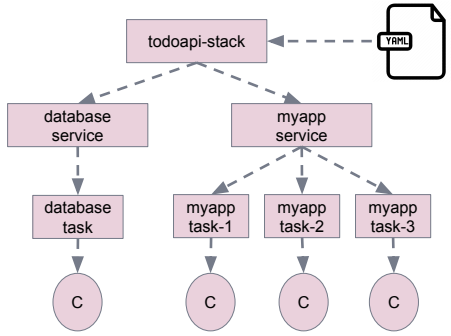
Docker stack

Docker stack is a collection of services that make up an application in a specific environment.



Docker stack

Docker stack is a collection of services that make up an application in a specific environment.



docker stack Commands

Command	Description
docker stack deploy	Deploy a new stack or update an existing stack
docker stack ls	List stacks
docker stack ps	List the tasks in the stack
docker stack rm	Remove one or more stacks
docker stack services	List the services in the stack

THANKS!

Any questions?

You can find me at:

- alex.d@clarusway.com