



# Docker Certified Associate

Docker Built-In Network Drivers

# Docker Network Drivers

Drivers for your Docker hosts and containers determine their behavior on each host as well as accessibility and routing to them.

## Network Driver Types

- Bridge
- None
- Host
- Overlay
- Ingress
- Docker Gateway Bridge



# Docker Network Drivers – Use Cases

Each Network Driver has a specific use case, in general:

## Bridge

- Simple to understand, use and troubleshoot, and is the default on stand-alone Docker hosts.
- Consists of a private network that is internal to the host system; all containers implemented on this host using Bridge networking can communicate.
- External access is granted by port exposure of the container's services and accessed by the host OR static routes added with the host as the gateway for that network.



# Docker Network Drivers – Use Cases

Each Network Driver has a specific use case, in general:

## None

- Used when the container(s) in question need absolutely no networking access at all.
- Containers operating on this driver can only be accessed on the host they are running on.
- These containers can be attached directly (using 'docker attach [containerid]') or executing by another command on the running container (using 'docker exec -it [containerid] [command]').
- Not commonly used.



# Docker Network Drivers – Use Cases

Each Network Driver has a specific use case, in general:

## Host

- Sometimes referred to as 'Host Only Networking'.
- Only accessible via the underlying host.
- Access to services can only be provided by exposing container service ports to the host system.



# Docker Network Drivers – Use Cases

Each Network Driver has a specific use case, in general:

## Overlay

- Allows communication among all Docker Daemons that are participating in a Swarm.
- It is a 'Swarm Scope' driver in that it extends itself (building previously non-existent networks on Workers if needed) to all daemons in the Swarm cluster.
- Allows the communication of multiple services that may have replicas running on any number of Workers in the Swarm, regardless of their origination or destination.
- Default mode of Swarm communication.



# Docker Network Drivers – Use Cases

Each Network Driver has a specific use case, in general:

## Ingress

- Special overlay network that load balances network traffic amongst a given service's working nodes.
- Maintains a list of all IP addresses from nodes that participate in that service (using the IPVS module) and when a request comes in, routes to one of them for the indicated service.
- Provides the 'routing mesh' that allows services to be exposed to the external network without having a replica running on every node in the Swarm.



# Docker Network Drivers – Use Cases

Each Network Driver has a specific use case, in general:

## Docker Gateway Bridge

- Special bridge network that allows overlay networks (including Ingress) access to an individual Docker daemon's physical network.
- Every container run within a service is connected to the local Docker daemon's host network.
- Automatically created when a Swarm is initialized or joined.





# Docker Network Drivers – Port Publishing Mode

## Two types of port publishing modes:

- **Host**

- Ports for containers are only available on the underlying host system and are NOT available for services outside of the host where that instance exists.
- Used in single host environments or in environments where you need complete control over routing.
- You are responsible for knowing where all instances are at all times, controlled with 'mode=host' in deployment.

- **Ingress**

- Since it is responsible for 'routing mesh', it makes all published ports available on all hosts (nodes or Workers) participating in the cluster so that the service is accessible from every node regardless of whether there is a service replica running on it at any given time.

