Swarm Management

Create a Swarm

Creating a new swarm is a simple operation. On a host running Docker Engine 1.12 or later, run the following command:

```
docker swarm init --advertise-addr IP Address:PORT
```

Substitute IP Address with the IP address or device name of the network interface that manager nodes should use to communicate between each other and to facilitate **overlay networking**. If the system has a single network interface or IP address, you do not need to use the `--advertise-addr` option. You can optionally specify the TCP port that should be used for API communication across the swarm, by default this is achieved on port 2377.

Example:

```
docker swarm init --advertise-addr 192.168.99.100

Swarm initialized: current node (rv5w5pk1ur4czkflfxj57gozr) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3l3pecu1x9w58x9csa7ohiw9rql2pi3zmebofy4ar

To add a manager to this swarm, run 'docker swarm join-token manager' and follow
```

Tips:

- The `--advertise-addr` flag configures the manager node to publish its address as 192.168.99.100. The other nodes in the swarm must be able to access the manager at the IP address.
- The output includes the commands to join new nodes to the swarm. Nodes will join as managers or workers depending on the value for the `--token` flag.

Run the `docker node ls` command to view information about nodes:

```
$ docker node ls

ID                            HOSTNAME            STATUS              AVAILABILI
rv5w5pk1ur4czkflfxj57gozr *   docker-desktop      Ready               Active
```

Tips:

- The "*" next to the node ID indicates that you're currently connected on this node.

Swarm Management

Add nodes to the swarm

Once you've created a swarm with a manager node, you're ready to add worker nodes.

Two secret keys, or join tokens, are created when the swarm is initialized. These keys are used to authenticate nodes that are attempting to join the swarm and, depending on the key used, can control whether a node joins the swarm as a worker node or as a manager node.

```
docker swarm join-token worker

To add a worker to this swarm, run the following command:

docker swarm join--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv
    -8vxv8rssmk743ojnwacrr2e7c \
  192.168.99.100:2377
```

```
docker swarm join-token manager

To add a manager to this swarm, run the following command:

docker swarm join--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv
    -8vxv8rssmk743ojnwacrr2e7c \
192.168.99.100:2377
```

To add nodes to the swarm, on each host run the following command:

```
docker swarm join --token TOKEN HOSTNAME|IP:PORT
```

Example: Run the command produced by the `docker swarm init` output from the create a swarm step to create a second **worker node** joined to the existing swarm:

```
$ docker swarm join \
  --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv
      -8vxv8rssmk743ojnwacrr2e7c \
    192.168.99.100:2377

This node joined a swarm as a worker.
```

Substitute TOKEN with the appropriate key value to authenticate the node to the swarm as either a `manager` or `worker` node. To obtain the correct key value, you can use the `docker swarm join-token manager` or `docker swarm join-token worker` command on a management node. This command will actually provide you with the full syntax of the command required to join the system to the swarm as a node.

From the manager node, you can run docker commands that control and report on the status of member nodes within a swarm. To check which nodes belong to the swarm, you can run the following command on the manager node where you initialized the swarm:

```
docker node ls
```

Swarm Management

You can promote any worker nodes to manager status at any time by running the following command on an existing manager node:

```
docker node promote ID|HOSTNAME
```

Substitute ID|HOSTNAME with the ID or hostname of the worker node that you wish to promote. For example, a worker node is promoted based on its docker ID:

```
docker node promote 4h0eba8xkko48msqtx3talp4i
```

You can also demote manager nodes to return them back to worker-only status. In the following example, a manager node is demoted based on its hostname:

```
docker node demote clarus.com
```

To remove a node from a swarm, run the following command on the node that must be removed:

```
docker swarm leave
```

To remove a manager node from a swarm, `--force` flag is required as in `docker swarm leave --force`.

Complementary Interactive Lesson about Swarm Management