Docker Swarm Cheat Sheet
by Gaurav Pandey (gauravpandey44) via cheatography.com/69622/cs/20590/

**Basic**

Swarm is Docker’s in-built containers orchestrator solution, its main purpose is to manage containers in a computer cluster, i.e. a set of connected computers that work together.

Swarm comes built into the Docker Engine, you don’t need to install anything to get started.

In Docker, there are many layers of encapsulation: the OS kernel, containers, tasks that encapsulate containers as units of work, services or Pods that represent application components, and stacks (collection of services) that represent full applications.

**docker swarm commands**

docker swarm init
- Initialize a swarm

docker swarm join --token<manager-token> 10.1.0.2:2377
- Join an existing swarm as manager node

docker swarm join --token<worker-token> 10.1.0.2:2377
- Join a swarm as a worker node

docker swarm leave
- Leave the swarm

**docker stack commands**

docker stack deploy nodeapp
- Deploy the stack using docker-compose file, Swarm does not support the build option if defined in the Compose file (but docker compose up uses it).

```
docker stack ls
```
- Shows all stacks along with list of services in the stack

```
docker stack services <stack-name>
```
- List the services in the stack.

```
docker stack ps <stack-name>
```
- List all the tasks in the stack.

```
docker stack rm <stack-name>
```
- Removes the stack.

Manage Docker stacks (is a collection of services that make up an application in a specific environment).

**docker node commands**

docker node ls
- List nodes in the swarm

docker node ps
- List tasks running on one or more nodes, defaults to current node

```
docker node rm [OPTIONS] NODE [NODE...]
```
- Remove one or more nodes from the swarm

```
docker node demote NODE [NODE...]
```
- Demote one or more nodes from manager in the swarm

```
docker node promote NODE [NODE...]
```
- Promote one or more nodes to manager in the swarm

Sponsored by ApolloPad.com
Everyone has a novel in them. Finish Yours!
https://apollopad.com
Common Terms/Key concepts

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>is a physical or virtual machine (running an instance of the Docker Engine.)</td>
</tr>
<tr>
<td>Manager nodes</td>
<td>perform swarm management and orchestration duties. By default manager nodes are also worker nodes.</td>
</tr>
<tr>
<td>Worker nodes</td>
<td>execute tasks.</td>
</tr>
<tr>
<td>Cluster</td>
<td>one or more nodes grouped together.</td>
</tr>
<tr>
<td>Swarm</td>
<td>is a type of cluster in docker terminology.</td>
</tr>
<tr>
<td>Docker Swarm</td>
<td>(not part of docker engine) is a separate product which you can use to cluster multiple Docker hosts. Prior to Docker version 1.12 it was the only native Docker option for clustering hosts, and it needed a lot of additional setup for distributed state, service discovery and security.</td>
</tr>
<tr>
<td>Swarm Mode</td>
<td>With Docker 1.12, (is built into Docker Engine) To run a cluster you just need to install Docker on multiple machines, run docker swarm init to switch to Swarm Mode and docker swarm join to add more nodes to the cluster. State, discovery and security are all included with zero setup.</td>
</tr>
<tr>
<td>Stack</td>
<td>is a collection of services that make up an application in a specific environment.</td>
</tr>
<tr>
<td>service</td>
<td>it defines the blueprint about which image to use and which commands to execute, ports, networks, replicas, etc inside future running containers.</td>
</tr>
<tr>
<td>task</td>
<td>in swarm model, task is actually invoked inside a container. When service is created the swarm manager starts a task (or its replicas) inside various containers.</td>
</tr>
</tbody>
</table>

Swarm service diagram

Application

**similar commands**

**Init host as a swarm manager node:**
docker swarm init

**Deploy application:**
docker stack deploy -c docker-compose.yml myApp

**List services:**
docker service ls
docker stack services myApp

**List tasks:**
docker service ps myApp_web
docker container ls -q
docker stack ps myApp

**Stop application:**
docker stack rm myApp

**Take down swarm**
docker swarm leave --force

By Gaurav Pandey (gauravpandey44)

cheatography.com/gauravpandey44/

Published 22nd September, 2019.
Last updated 22nd September, 2019.
Page 2 of 2.

Sponsored by ApolloPad.com

Everyone has a novel in them. Finish Yours!
https://apollopad.com