

# Docker Swarm

Adfinis**sy**Group

Be smart. Think open source.

# Container Orchestration

# What is container orchestration

Basically:

Container orchestration refers to the process of organising the work of individual components and application layers.

## Why use container orchestration?

- Provisioning and deployment of containers
- Redundancy and availability of containers
- Scaling up or removing containers to spread application load evenly across host infrastructure
- Movement of containers from one host to another if there is a shortage of resources in a host, or if a host dies
- Allocation of resources between containers
- External exposure of services running in a container with the outside world
- Load balancing of service discovery between containers
- Health monitoring of containers and hosts
- Configuration of an application in relation to the containers running it

## Known container orchestrators

- [Kubernetes](#)
- [Docker Swarm](#)
- [Amazon ECS](#)
- [Azure Container Instances](#)
- [Rancher](#)
- [DC/OS](#)
- [Google Container Engine](#)
- [RedHat OpenShift](#)
- [Mesosphere Marathon](#)
- [HashiCorp Nomad](#)

# Docker Swarm

## Why use Docker Swarm?

- Cluster management integrated with Docker Engine
- Declarative service model
- Horizontal scaling
- Multi-Host networking
- Service discovery & Load balancing
- Rolling updates



## What to think before creating a swarm cluster

- At least two docker nodes
- Networking will be taken care by Swarm
- Needed open ports between nodes: TCP 2377 & TCP/UDP 7946 & UDP 4789
- Storage management isn't part of Swarm

# Create a swarm cluster

On first node:

```
docker swarm init --advertise-addr <MANAGER-IP>
```

On other nodes:

```
docker swarm join --token SWMTKN-1v-XXX <MANAGER-IP>:2377
```

## Basic swarm commands

```
$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
c8636ss0c93jb4vgi4hqptiq7 *	node-docker-01	Ready	Active	Leader	18.09.0
ui1u7tcs9sdanmbz31uaei	node-docker-02	Ready	Active		18.09.0
4sdybt048q0kba4pvukprnogu	node-docker-03	Ready	Active		18.09.0

## Promote nodes to have more than one manager

```
$ docker node promote node-docker-02
$ docker node promote node-docker-03
$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
c8636s0c93jb4vgi4hqptiq7 *	node-docker-01	Ready	Active	Leader	18.09.0
ui1u7tcs9sdanmfbzzo31uaei	node-docker-02	Ready	Active	Reachable	18.09.0
4sdybt048q0kba4pvukprnogu	node-docker-03	Ready	Active	Reachable	18.09.0

## Drain node

```
$ docker node update --availability drain node-docker-02
```

```
$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
c8636ss0c93jb4vgi4hqptiq7 *	node-docker-01	Ready	Active	Leader	18.09.0
ui1u7tcs9sdanmfbzso31uaei	node-docker-02	Ready	Drain		18.09.0
4sdybt048q0kba4pvukprnogu	node-docker-03	Ready	Active		18.09.0

## Add labels to node

```
$ docker node update --label-add disk=ssd node-docker-02
```

## Remove node from swarm

```
$ docker swarm leave  
Node left the swarm
```

## Deploy a swarm service

```
$ docker service create --replicas 1 --name helloworld alpine:3.7 ping docker.com
25x6pxxd3d8sk7r9mpqudzs5e
overall progress: 1 out of 1 tasks
1/1: running [=====]
verify: Service converged
$ docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
25x6pxxd3d8s	helloworld	replicated	1/1	alpine:3.7	



# Scale a service

```
$ docker service scale helloworld=5
helloworld scaled to 5
overall progress: 5 out of 5 tasks
1/5: running [=====>]
2/5: running [=====>]
3/5: running [=====>]
4/5: running [=====>]
5/5: running [=====>]
verify: Service converged
$ docker service ls
ID            NAME           MODE           REPLICAS        IMAGE          PORTS
25x6pxxd3d8s helloworld    replicated     5/5             alpine:latest
```

## Check where service is running

```
$ docker service ps helloworld
```

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
kynrhzr2r653	helloworld.1	alpine:latest	node-docker-01	Running	Running about a minute ago		
i23ti7nmvvy56	helloworld.2	alpine:latest	node-docker-02	Running	Running 22 seconds ago		
6fmiwpal6cus	helloworld.3	alpine:latest	node-docker-01	Running	Running 25 seconds ago		
wx963db45xrm	helloworld.4	alpine:latest	node-docker-03	Running	Running 23 seconds ago		
b15fg9ueol31	helloworld.5	alpine:latest	node-docker-02	Running	Running 22 seconds ago		

# Rolling updates

```
$ docker service update --image alpine:3.8 --update-delay 10s helloworld
helloworld
overall progress: 5 out of 5 tasks
1/5: running [=====>]
2/5: running [=====>]
3/5: running [=====>]
4/5: running [=====>]
5/5: running [=====>]
verify: Service converged
```

# Use compose files for service creation & update

Using compose file version 3

```
$ docker swarm stack deploy --compose-file docker-compose.yml myapp  
Creating network myapp_default  
Creating service myapp_web  
Creating service myapp_redis
```

# Example compose file

```
version: "3.3"

services:
  wordpress:
    image: wordpress
    ports:
      - "8080:80"
    networks:
      - overlay
    deploy:
      mode: replicated
      replicas: 2

  mysql:
    image: mysql
    volumes:
      - db-data:/var/lib/mysql/data
    networks:
      - overlay

volumes:
  db-data:

networks:
  overlay:
```

More informations: [Official Docs](#)

# Feel Free to Contact Us

[www.adfinis-sygroup.ch](http://www.adfinis-sygroup.ch)

[Tech Blog](#)

[GitHub](#)

[info@adfinis-sygroup.ch](mailto:info@adfinis-sygroup.ch)

[Twitter](#)

