

Docker: swarm mode

主讲：郭宁@mobike

2018-10-24

目录

- Docker swarm mode

- 是什么?

- 怎么用?

- 核心概念

- 搭建 Swarm 集群

- 部署服务 Service

- 服务升级 (滚动升级)

- 什么原理?

- 参考资料

Swarm mode: 是什么

- 关于 Swarm mode 是什么，关注 2 个方面：
 - **Swarm 简介**：方向上，说明 Swarm 解决什么问题
 - **Swarm 新特性**：细节上，提供哪些功能，能够解决什么问题

Swarm mode 简介

- **swarm 的作用：**
 - Docker Engine 内置（原生）的 **集群管理** 和 **服务编排** 工具。
- 关于 **swarm** 的**历史演进**：
 - **Docker Swarm：**
 - 在 **Docker 1.12** 版本之前，是独立的组件
 - 独立于 Docker Engine 之外，需要**独立安装**；
 - **swarm mode：**
 - 在 **Docker 1.12+**(涵盖1.12)，内置到了 Docker Engine 之中
 - 无需独立安装；
- 官方建议：
 - 如果要使用 swarm，直接升级到 **Docker 1.12+** 的新版本，使用 **swarm mode** 即可。

Swarm 新特性

- Swarm mode 内置 kv 存储功能，提供了众多的新特性：
 - 去中心化：具有容错能力
 - 启动服务节点，运行时可以设置不同角色：manager、worker
 - 声明式的服务模型，可以直接定义一个应用，包含哪些服务
 - 监听 worker 的状态，如果 Container 挂掉，Container 会自动重启
 - 服务发现：内置了服务发现
 - 负载均衡：服务可以对接外部的 LB，swarm 也支持指定 service 在 node 上的分布
 - 路由网格：overlay 网络，虚拟网络，就近服务注册和服务发现，以及服务路由
 - 动态伸缩：服务层面，容器数量的伸缩
 - 滚动更新：服务部署的粒度，细化到容器
 - 安全传输：使用 TLS 协议进行安全的通信
- 这些特性，使得 Docker 原生的 Swarm 集群，具备与 **Mesos**、**Kubernetes** 竞争的实力。

目录

- Docker swarm mode
 - 是什么?
 - 怎么用?
 - 核心概念
 - 搭建 Swarm 集群
 - 部署服务 Service
 - 服务升级 (滚动升级)
 - 什么原理?
- 参考资料

Swarm mode: 怎么用

- 关于 Swarm mode 的使用，关注几个问题：
 1. **核心概念**：Swarm mode 场景下，存在哪些对象？相互之间的关系
 2. **搭建 Swarm 集群**：多个 Docker 服务节点，组成一个集群
 3. **部署服务 Service**：在 Swarm 集群上，部署服务、管理服务
 4. **服务升级（滚动升级）**：在 Swarm 集群中，滚动升级、以及回滚操作

目录

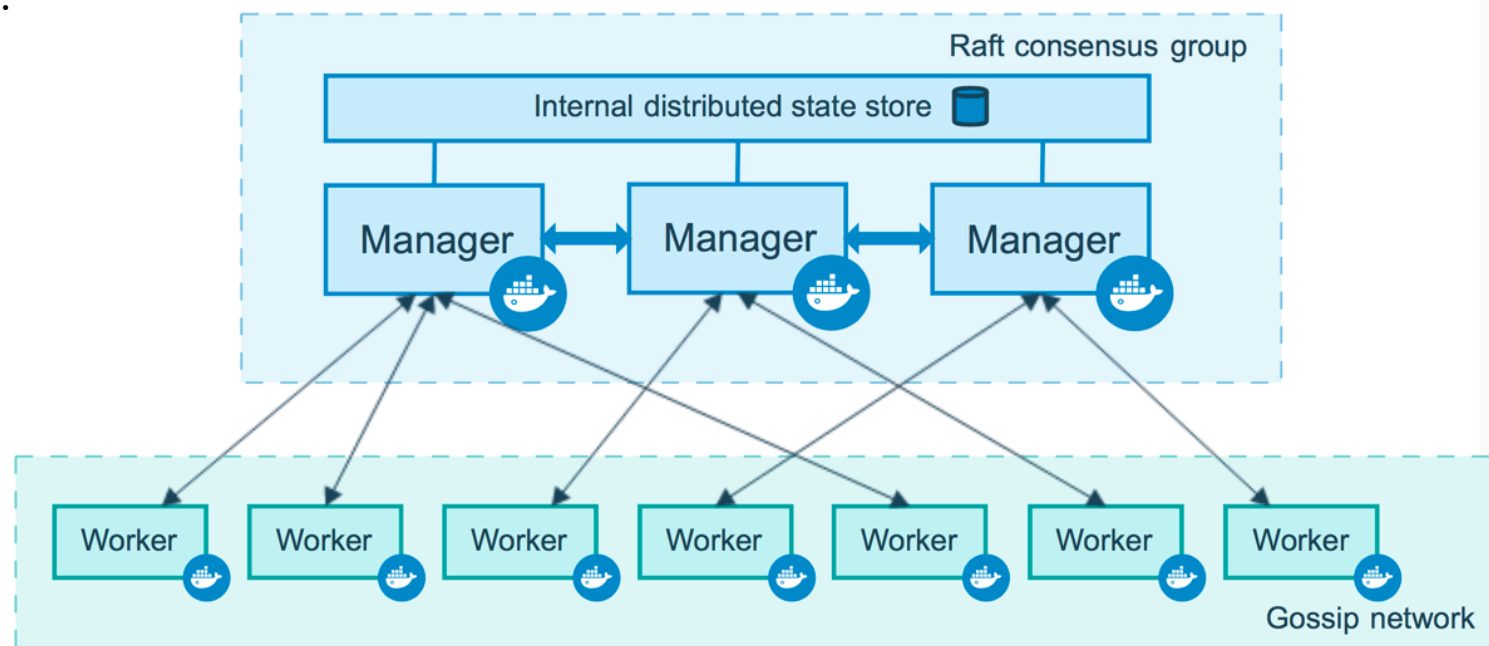
- Docker swarm mode
 - 是什么?
 - 怎么用?
- 核心概念
- 搭建 Swarm 集群
- 部署服务 Service
- 服务升级 (滚动升级)
- 什么原理?
- 参考资料

核心概念

- Swarm mode 几个**核心概念**:
 - **Docker 节点**: Docker 服务器, 分为不同的**角色** (**物理维度**划分)
 - **Service 服务**: 服务 和 任务 (**逻辑维度**划分)
- 详细信息, 参考:
 - [how-swarm-mode-works](#)

Docker 节点

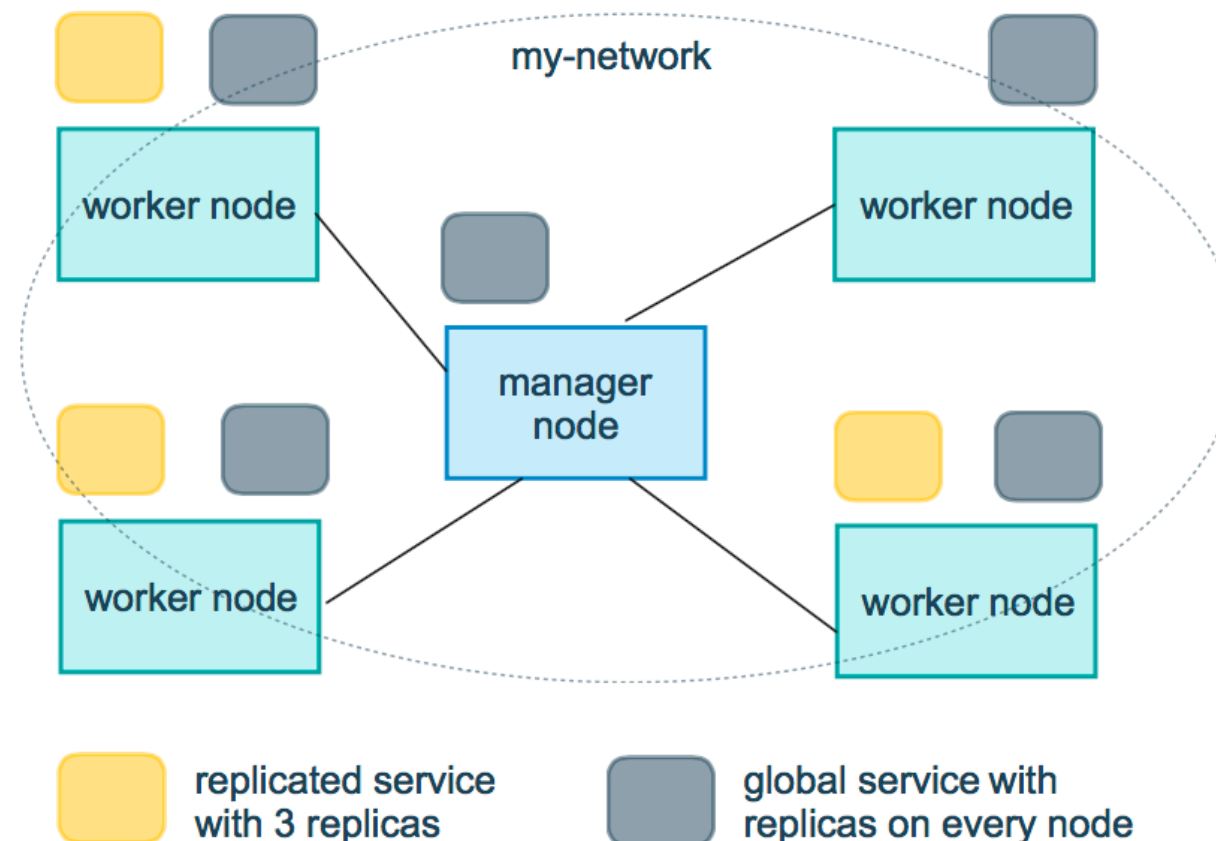
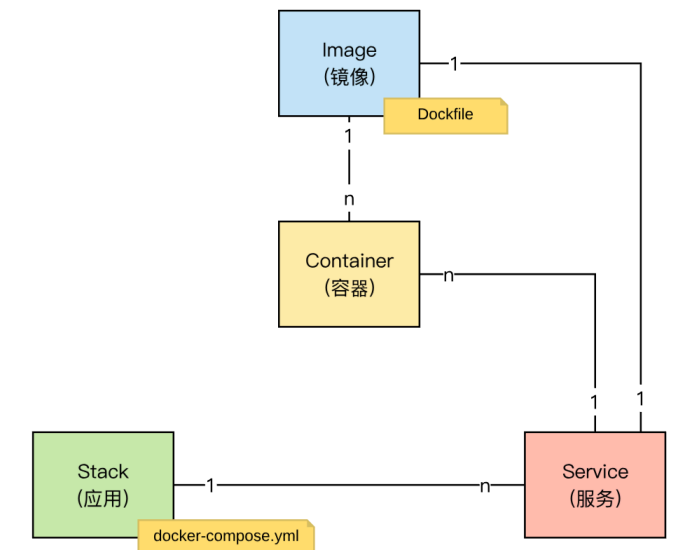
- **Docker 节点**: Docker node
 - 本质: Docker 服务器, Docker daemon
 - **2 种角色**: 管理节点 (manager) 和工作节点 (worker)
 - 管理节点 manager:
 - **管理** swarm 集群
 - 可以存在多个 manager 节点, 但依赖 **Raft** 协议, **只有一个 leader**
 - docker swarm **管理命令**, 只能在 manager 上执行
 - 工作节点 worker:
 - **执行**任务
 - manager 节点, 默认, 也可以作为 worker 节点
 - docker machine 创建 Docker 节点, **join 到 swarm 集群中**, 并设置自己角色
 - 同一个 Docker node, 可以选择 **3 种角色**:
 - manager
 - worker
 - 同时是 manager 和 worker



服务 (Service) 和任务 (Task)

Docker 技术原理：关键概念

- 服务和任务 (Service & Task) :
 - Service, 一个服务, 可能涵盖多个容器
 - Task, 是指 worker 节点上, 创建一个容器的任务
- 关系:
 - Service 创建多个容器, 每个容器对应一个 Task
- Service 分类:
 - **replicated service**: 容器数量固定, 跟 worker node 数量无关;
 - **global service**: 每个 worker node , 运行一个容器, 容器数量跟 worker node 绑定;



目录

- Docker swarm mode
 - 是什么?
 - 怎么用?
 - 核心概念
- 搭建 Swarm 集群
- 部署服务 Service
- 服务升级 (滚动升级)
- 什么原理?
- 参考资料

搭建 Swarm 集群

- 目标：
 - 搭建一个 swarm 集群，本质，**Docker 服务器的集群**。
- 下面的场景中，将进行如下操作：
 1. 创建节点：创建一个 Docker 集群：1 个 manager、2 个 worker；
 2. 角色初始化：manager 节点，进行 swarm 初始化
 3. 增加 worker 节点：worker 节点，加入 swarm 集群
- 细节参考：
 - Getting started with swarm mode
 - 创建 Swarm 集群

1. 创建一个 Docker 节点, 命名为 manager
docker-machine create -d virtualbox manager

2. 登录 manager 节点, 完成 swarm 模式初始化
docker-machine ssh manager

在 manager 节点上, 完成下述 swarm 初始化
docker swarm init --advertise-addr 192.168.99.100

...

Swarm initialized: current node (j9iopsxdcrrwm0ayughex405zh) is now a manager.

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

```
docker swarm join --token SWMTKN-1-4jp7adsqdwjw1kkijnstkvj1t3xkcmmmr6oqgmahz0tmqkv7-12mk3k2zv97hju7gafpq1ia7f 192.168.99.100:2377
```

To add a manager to **this** swarm, run 'docker swarm join-token manager' and follow the instructions.

...

3. 创建 2 个 worker 节点
docker-machine create -d virtualbox worker1
docker-machine create -d virtualbox worker2

4. 分别 ssh 登录两个 worker 节点, 并执行 swarm join 命令
docker-machine ssh worker1
docker-machine ssh worker2

...

```
docker swarm join --token SWMTKN-1-4jp7adsqdwjw1kkijnstkvj1t3xkcmmmr6oqgmahz0tmqkv7-12mk3k2zv97hju7gafpq1ia7f 192.168.99.100:2377
```

...

5. 查看 Docker 集群状态 (ssh 登录到 manager 节点, 执行下述命令)
\$ docker node ls

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
j9iopsxdcrrwm0ayughex405zh *	manager	Ready	Active	Leader	18.06.0-ce
ecx168drw3tc0ct4hzb9kf90j	worker1	Ready	Active		18.06.0-ce
sco7gy7tzx9ciu9gb5pk8zu0r	worker2	Ready	Active		18.06.0-ce

目录

- Docker swarm mode
 - 是什么?
 - 怎么用?
 - 核心概念
 - 搭建 Swarm 集群
 - 部署服务 Service
 - 服务升级 (滚动升级)
 - 什么原理?
- 参考资料

部署服务 Service

- 两种场景下，部署服务：
 - 单个镜像
 - **docker-compose** 配置「一组服务」
- 详细信息，参考：
 - [Getting started with swarm mode](#)

部署服务： 单个镜像

- 使用 **docker service** 命令，具体：
 - docker service create： 基于单个镜像，**创建**服务，一次只能创建一个服务.
 - docker service ls： **查看**所有 service 列表
 - docker service ps [service]： **查看**具体 service 详情
 - docker service logs [service]： **查看**具体 service 运行的 **log**
 - docker service scale [service]=[num]： 服务**伸缩**
 - docker service rm [service]： **删除**服务

1. 登录 manager 节点

```
docker-machine ssh manager
```

2. 创建 service

```
docker service create --replicas 3 -p 80:80 --name nginx nginx:1.13.7-alpine
```

```
docker service create --replicas 3 -p 80:80 --name nginx nginx:1.13.7-alpine
```

```
wzdkv1925fxqt5iz5f7dthf6w
```

```
overall progress: 3 out of 3 tasks
```

```
1/3: running [=====>]
```

```
2/3: running [=====>]
```

```
3/3: running [=====>]
```

```
verify: Service converged
```

3. 查询 service 列表

```
docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
wzdkv1925fxq	nginx	replicated	3/3	nginx:1.13.7-alpine	*:80->80/tcp

4. 查询单个 service 详情

```
docker service ps nginx
```

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR
yhp4qvcpn5dk	nginx.1	nginx:1.13.7-alpine	manager	Running	Running 11 minutes ago	
babdsuk5w9an	nginx.2	nginx:1.13.7-alpine	worker1	Running	Running 11 minutes ago	
x6siyj94d3id	nginx.3	nginx:1.13.7-alpine	worker2	Running	Running 11 minutes ago	

5. 服务伸缩

```
docker service scale nginx=4
```

```
nginx scaled to 4
```

```
overall progress: 4 out of 4 tasks
```

```
1/4: running [=====>]
```

```
2/4: running [=====>]
```

```
3/4: running [=====>]
```

```
4/4: running [=====>]
```

```
verify: Service converged
```

6. 删除服务

```
docker service rm nginx
```

部署服务： docker-compose

- 目标：
 - 使用 docker compose 可以一次配置，**启动**多个服务、多个容器；
 - 在 swarm 模式下，也可以使用 ***docker-compose.yml*** 来配置、启动服务。
- **Note:** docker service create 一次只能创建一个服务，借助 docker compose 可以创建多个服务。
- 基本步骤：
 1. **定义服务**：在 manager 节点，准备 docker-compose.yml 文件
 2. **启动服务**：使用 docker stack deploy 命令部署

- **步骤 1:** 在 manager 节点, 创建下述 docker-compose.yml 文件
- 关于上述 docker-compose.yml 文件说明
 - 在 Swarm 集群「manager 节点」新建该文件
 - 其中的 visualizer 服务, 提供一个可视化页面, 我们可以从浏览器中很直观的查看集群中各个服务的运行节点;

```
version: "3"

services:
  wordpress:
    image: wordpress
    ports:
      - 80:80
    networks:
      - overlay
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
    deploy:
      mode: replicated
      replicas: 3

  db:
    image: mysql
    networks:
      - overlay
    volumes:
      - db-data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
    deploy:
      placement:
        constraints: [node.role == manager]

  visualizer:
    image: dockersamples/visualizer:stable
    ports:
      - "8080:8080"
    stop_grace_period: 1m30s
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock"
    deploy:
      placement:
        constraints: [node.role == manager]

volumes:
  db-data:
networks:
  overlay:
```

- **步骤 2:**
- 在 manager 节点上,
执行 docker stack 命令

```
# 1. 基于 docker compose 创建 stack  
docker stack deploy -c docker-compose.yml wordpress_stack
```

```
Creating network wordpress_stack_overlay  
Creating network wordpress_stack_default  
Creating service wordpress_stack_wordpress  
Creating service wordpress_stack_db  
Creating service wordpress_stack_visualizer
```

```
# 2. 查看 stack  
docker stack ls
```

```
# 3. 移除 stack  
docker stack rm [stack]
```

部署服务： docker-compose

- 任何一个 Docker 服务器节点上，浏览器访问 8080 端口，即可查看到「服务-节点」的分布情况.



目录

- Docker swarm mode
 - 是什么?
 - 怎么用?
 - 核心概念
 - 搭建 Swarm 集群
 - 部署服务 Service
 - 服务升级 (滚动升级)
 - 什么原理?
- 参考资料

服务升级（滚动升级）

- 在 swarm mode 下，如何进行滚动升级？
 1. 滚动升级如何操作？
 2. 如何回滚？
- Note：使用 `docker service update` 进行滚动升级


```
# 1. 创建 nginx 服务
docker service create --replicas 3 -p 80:80 --name nginx nginx:1.13.7-alpine
```

```
# 2. 滚动升级 nginx 服务到 1.13.12
docker service update --image nginx:1.13.12-alpine nginx
```

```
# 3. 查看滚动升级结果
docker service ps nginx
```

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
kct3ksqxljpu	nginx.1	nginx:1.13.12-alpine	worker1	Running	Running 19 seconds ago		
vcbkaz7q8kya	_ nginx.1	nginx:1.13.7-alpine	worker1	Shutdown	Shutdown 36 seconds ago		
qlyxhdee0i6c	nginx.2	nginx:1.13.12-alpine	worker2	Running	Running 56 seconds ago		
dshb5x0lo61s	_ nginx.2	nginx:1.13.7-alpine	worker2	Shutdown	Shutdown about a minute ago		
s5es0n52tzqx	nginx.3	nginx:1.13.12-alpine	manager	Running	Running 40 seconds ago		
t9mczkuy8dd6	_ nginx.3	nginx:1.13.7-alpine	manager	Shutdown	Shutdown 53 seconds ago		

```
# 3. 回滚
docker service rollback nginx
```

```
# 4. 查看回滚记录
docker service ps nginx
```

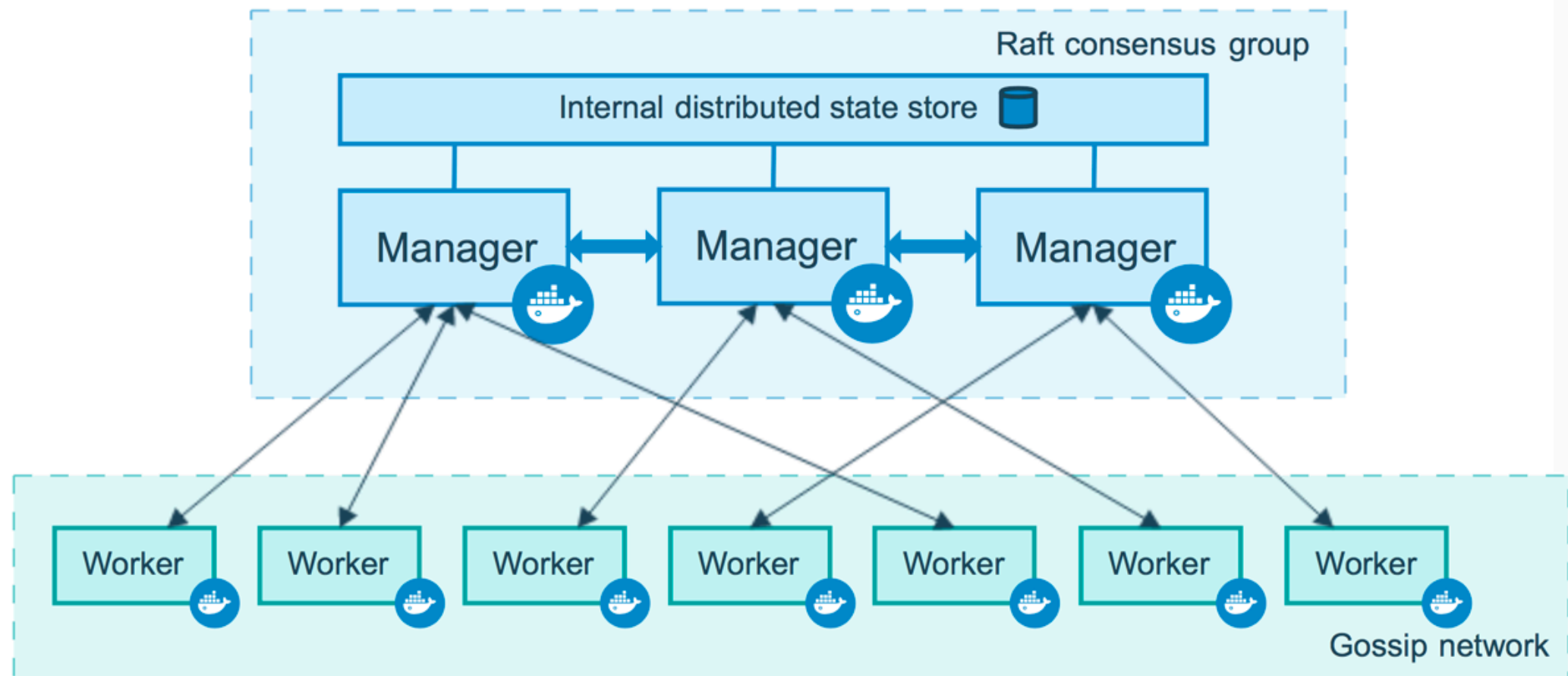
ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
uru5lwrvmjn5	nginx.1	nginx:1.13.7-alpine	worker1	Running	Running 44 seconds ago		
kct3ksqxljpu	_ nginx.1	nginx:1.13.12-alpine	worker1	Shutdown	Shutdown 45 seconds ago		
vcbkaz7q8kya	_ nginx.1	nginx:1.13.7-alpine	worker1	Shutdown	Shutdown about a minute ago		
kxzfpu30mwp1	nginx.2	nginx:1.13.7-alpine	worker2	Running	Running 48 seconds ago		
qlyxhdee0i6c	_ nginx.2	nginx:1.13.12-alpine	worker2	Shutdown	Shutdown 49 seconds ago		
dshb5x0lo61s	_ nginx.2	nginx:1.13.7-alpine	worker2	Shutdown	Shutdown 2 minutes ago		
4yewijsw2ryy	nginx.3	nginx:1.13.7-alpine	manager	Running	Running 40 seconds ago		
s5es0n52tzqx	_ nginx.3	nginx:1.13.12-alpine	manager	Shutdown	Shutdown 41 seconds ago		
t9mczkuy8dd6	_ nginx.3	nginx:1.13.7-alpine	manager	Shutdown	Shutdown 2 minutes ago		

目录

- Docker swarm mode
 - 是什么?
 - 怎么用?
 - 核心概念
 - 搭建 Swarm 集群
 - 部署服务 Service
 - 服务升级 (滚动升级)
 - 什么原理?
- 参考资料

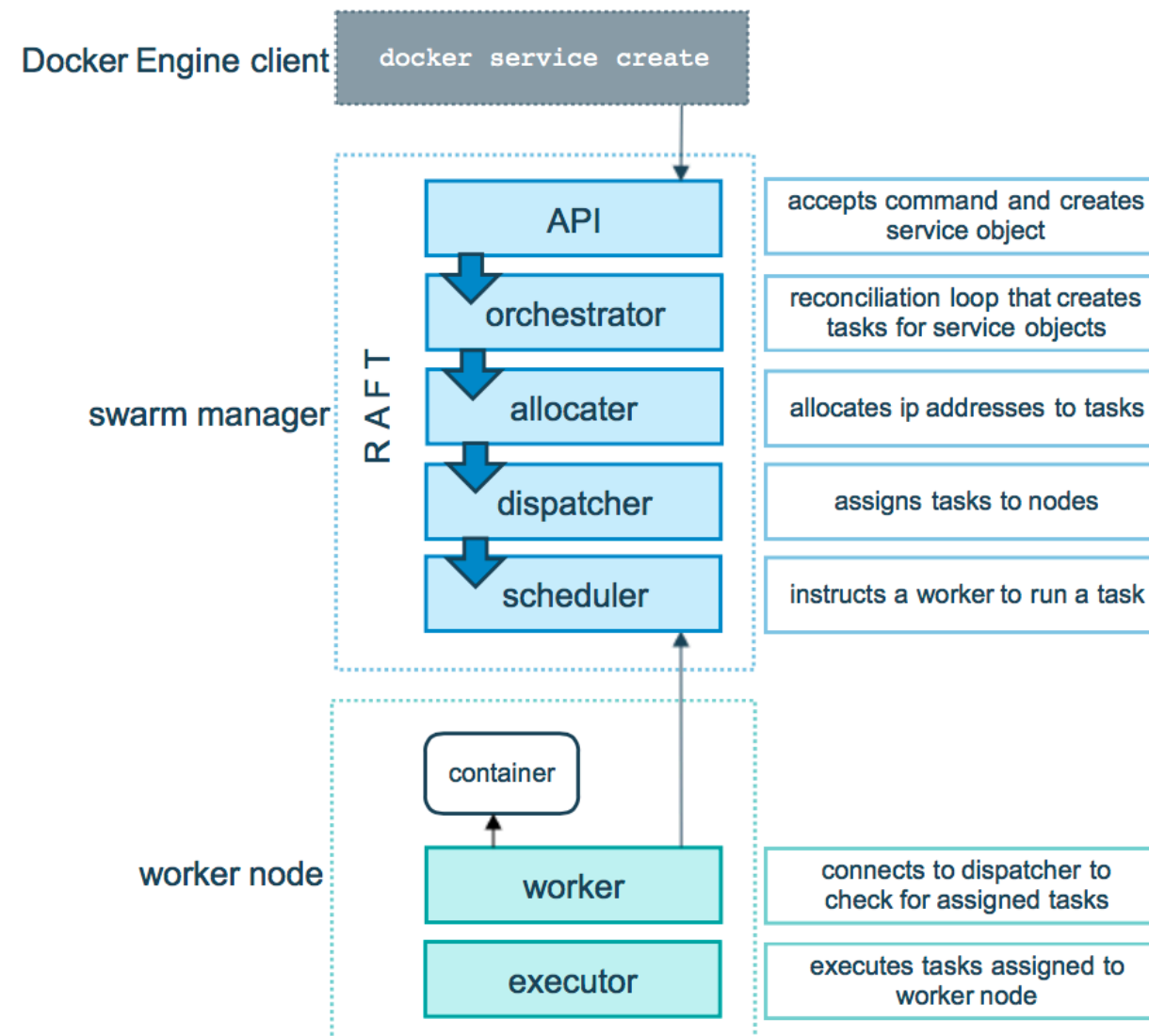
Swarm mode: 什么原理

- Swarm 集群的架构：
 - 物理架构：
 - 多个 Docker 服务器节点，join 构成
 - 2 种角色，manager、worker
 - 逻辑架构：
 - 分为 Service 和 Task，控制服务的部署



Swarm mode: 什么原理

- Swarm 集群的管理：
 - **Manager**: 接收管理命令 docker service or docker stack
 - **Worker**: 执行具体的 Task, 管理 Container 的生命周期



目录

- Docker swarm mode
 - 是什么?
 - 怎么用?
 - 核心概念
 - 搭建 Swarm 集群
 - 部署服务 Service
 - 服务升级 (滚动升级)
 - 什么原理?
- 参考资料

参考资料

- <https://docs.docker.com/engine/swarm/> Swarm mode Overview
- <https://docs.docker.com/swarm/> Docker Swarm
- <https://docs.docker.com/compose/swarm/> Use Compose with Swarm
- <https://docs.docker.com/engine/swarm/swarm-tutorial/> Getting started with swarm mode
- https://yeasy.gitbooks.io/docker_practice/content/swarm_mode/create.html 创建 Swarm 集群