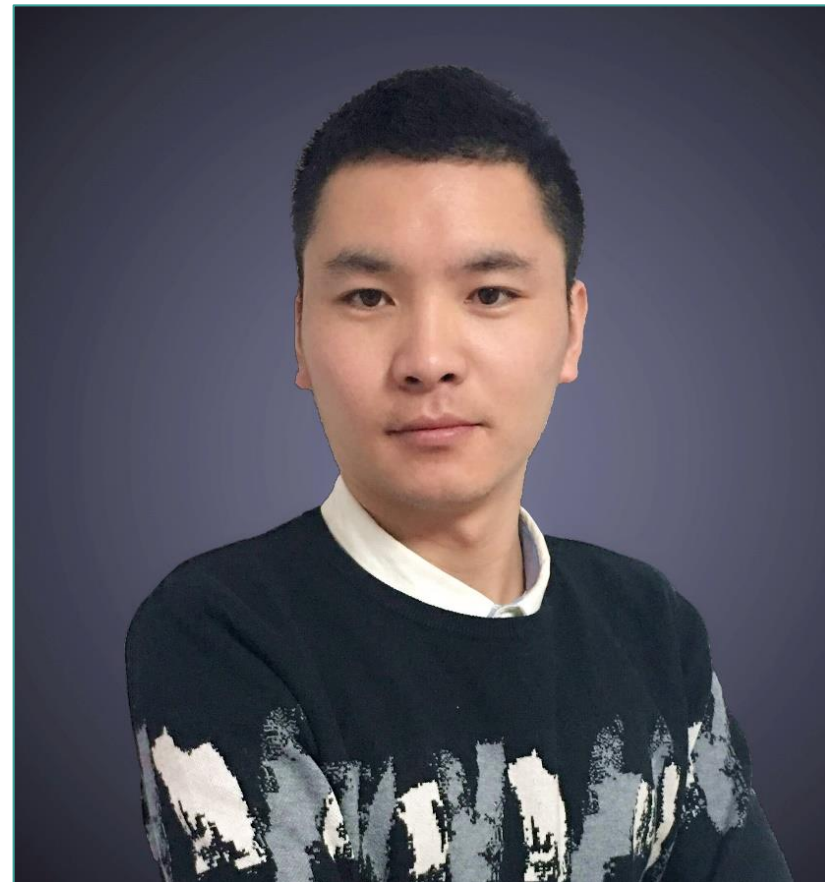


致敬未来的你！

Docker Swarm容器集群管理应用实战

个人介绍



讲师：李振良

资深运维工程师，51CTO知名博主。曾就职在IDC，大数据，金融行业，现任职360公司，经重重磨练，具备丰富的运维实战经验。

技术博客：<http://blog.51cto.com/lizhenliang>

DevOps技术栈

专注于分享DevOps工具链
及经验总结。



Docker/K8s技术学员群：[397834690](#)

Swarm容器集群管理

- Swarm介绍
- 集群部署
- 节点管理
- 服务管理
- 管理应用程序数据
- 集群服务发布
- 服务发现与负载均衡
- 高可用架构
- 配置文件管理
- 应用案例

Swarm容器集群管理

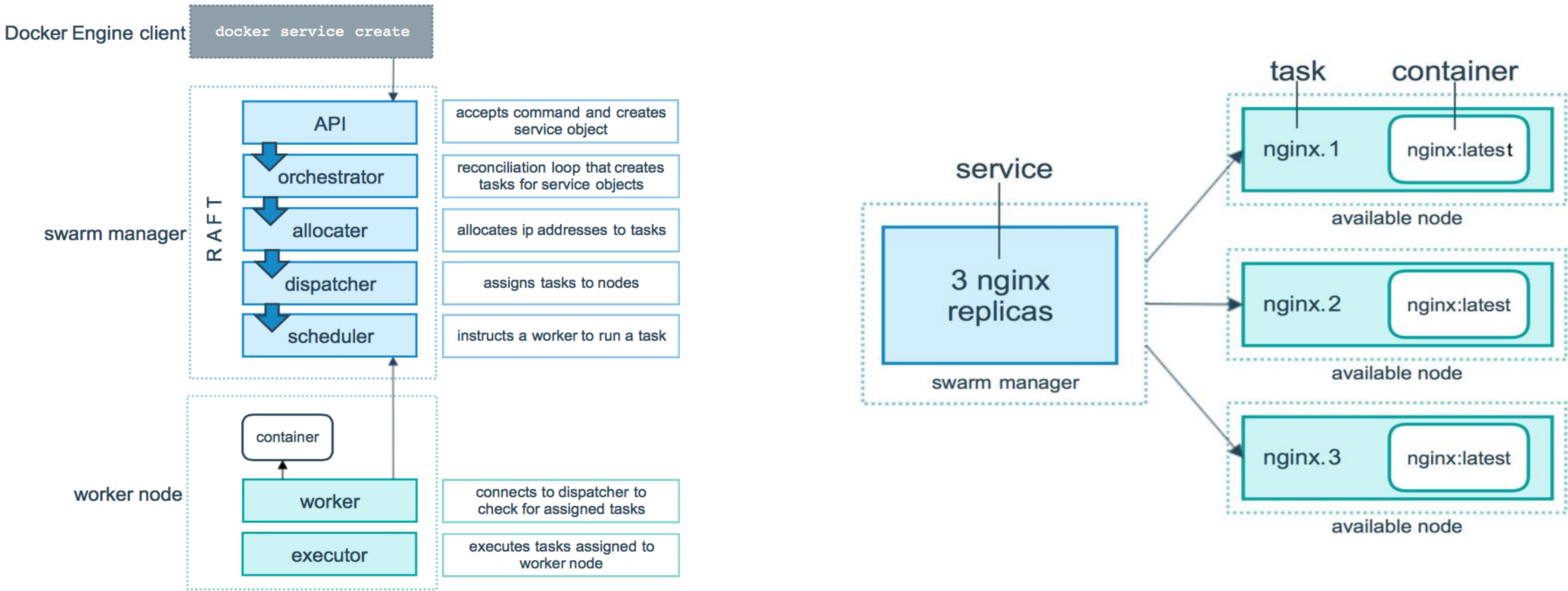
Swarm介绍

Swarm是Docker公司自研发的容器集群管理系统，Swarm在早期是作为一个独立服务存在，在Docker Engine v1.12中集成了Swarm的集群管理和编排功能。可以通过初始化Swarm或加入现有Swarm来启用Docker引擎的Swarm模式。

Docker Engine CLI和API包括了管理Swarm节点命令，比如添加、删除节点，以及在Swarm中部署和编排服务。

也增加了服务栈（Stack）、服务（Service）、任务（Task）概念。

Swarm介绍



Manager：接收客户端服务定义，将任务发送到worker节点；维护集群期望状态和集群管理功能及Leader选举。默认情况下manager节点也会运行任务，也可以配置只做管理任务。

Worker：接收并执行从管理节点分配的任务，并报告任务当前状态，以便管理节点维护每个服务期望状态。

Swarm容器集群管理

Swarm介绍

Swarm特点：

1. Docker Engine集成集群管理

使用Docker Engine CLI 创建一个Docker Engine的Swarm模式，在集群中部署应用程序服务。

2. 去中心化设计

Swarm角色分为Manager和Worker节点，Manager节点故障不影响应用使用。

3. 扩容缩容

可以声明每个服务运行的容器数量，通过添加或删除容器数自动调整期望的状态。

4. 期望状态协调

Swarm Manager节点不断监视集群状态，并调整当前状态与期望状态之间的差异。

5. 多主机网络

可以为服务指定overlay网络。当初始化或更新应用程序时，Swarm manager会自动为overlay网络上的容器分配IP地址。

6. 服务发现

Swarm manager节点为集群中的每个服务分配唯一的DNS记录和负载均衡VIP。可以通过Swarm内置的DNS服务器查询集群中每个运行的容器。

7. 负载均衡

实现服务副本负载均衡，提供入口访问。

8. 安全传输

Swarm中的每个节点使用TLS相互验证和加密，确保安全的其他节点通信。

9. 滚动更新

升级时，逐步将应用服务更新到节点，如果出现问题，可以将任务回滚到先前版本。

Swarm容器集群管理

集群部署

使用Swarm前提：

- ◆ Docker版本1.12+
- ◆ 集群节点之间保证TCP 2377（集群管理）、TCP/UDP 7946（容器网络发现）和UDP 4789（Overlay网络）端口通信

节点规划：

操作系统：CentOS7.4_x64

Manager	Worker
Manager	192.168.0.211
Worker01	192.168.0.212
Worker02	192.168.0.213

docker swarm COMMAND 管理Swarm

管理节点初始化Swarm：

```
docker swarm init --advertise-addr 192.168.0.211
```

工作节点加入Swarm：

```
docker swarm join --token SWMTKN-1-XXX 192.168.0.211:2377
```

Swarm容器集群管理

节点管理

docker node COMMAND 管理Swarm节点

Swarm容器集群管理

服务管理

创建服务

```
docker service create --replicas 1 --name hello busybox
```

显示服务详细信息

```
docker service inspect --pretty hello # 易于阅读显示
```

```
docker service inspect hello # json格式返回
```

扩展服务实例数

```
docker service scale hello=3
```

查看服务任务

```
docker service ls
```

```
docker service ps hello
```

```
docker service ps -f 'desired-state=running' hello
```

滚动更新服务

```
docker service create \
```

```
--replicas 3 \
```

```
--name redis \
```

```
--update-delay 10s \
```

```
redis:3.0.6
```

```
docker service update --image redis:3.0.7 redis
```

创建服务时设定更新策略

```
docker service create \
```

```
--name my_web \
```

```
--replicas 10 \
```

```
--update-delay 10s \
```

```
--update-parallelism 2 \
```

```
--update-failure-action continue \
```

```
nginx:1.12
```

创建服务时设定回滚策略

```
docker service create \
```

```
--name my_web \
```

```
--replicas 10 \
```

```
--rollback-parallelism 2 \
```

```
--rollback-monitor 20s \
```

```
--rollback-max-failure-ratio .2 \
```

```
nginx:1.12
```

服务更新

```
docker service update --image nginx:1.13 my_web
```

手动回滚

```
docker service update --rollback my_web
```

Swarm容器集群管理

管理应用程序数据 – 将宿主机数据挂载到容器

Volume

创建数据卷

```
docker service create \
  --mount type=volume,src=<VOLUME-NAME>,dst=<CONTAINER-PATH> \
  --name myservice \
  <IMAGE>
```

查看数据卷详细信息

```
docker volume inspect <VOLUME-NAME>
```

Bind Mounts

读写挂载

```
docker service create \
  --mount type=bind,src=<HOST-PATH>,dst=<CONTAINER-PATH> \
  --name myservice \
  <IMAGE>
```

只读挂载

```
docker service create \
  --mount type=bind,src=<HOST-PATH>,dst=<CONTAINER-PATH>,ro \
  --name myservice \
  <IMAGE>
```

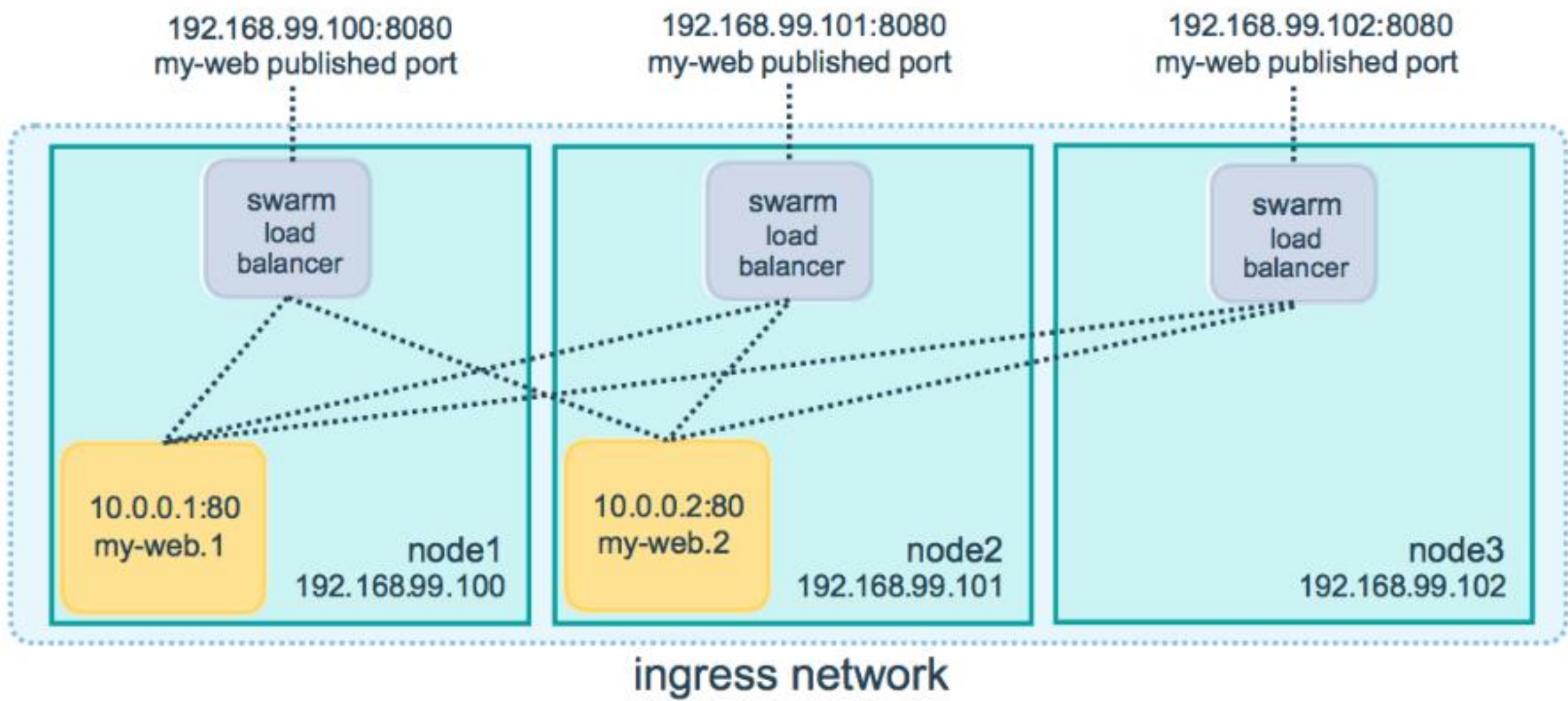

Swarm容器集群管理

管理应用程序数据 – NFS数据持久存储

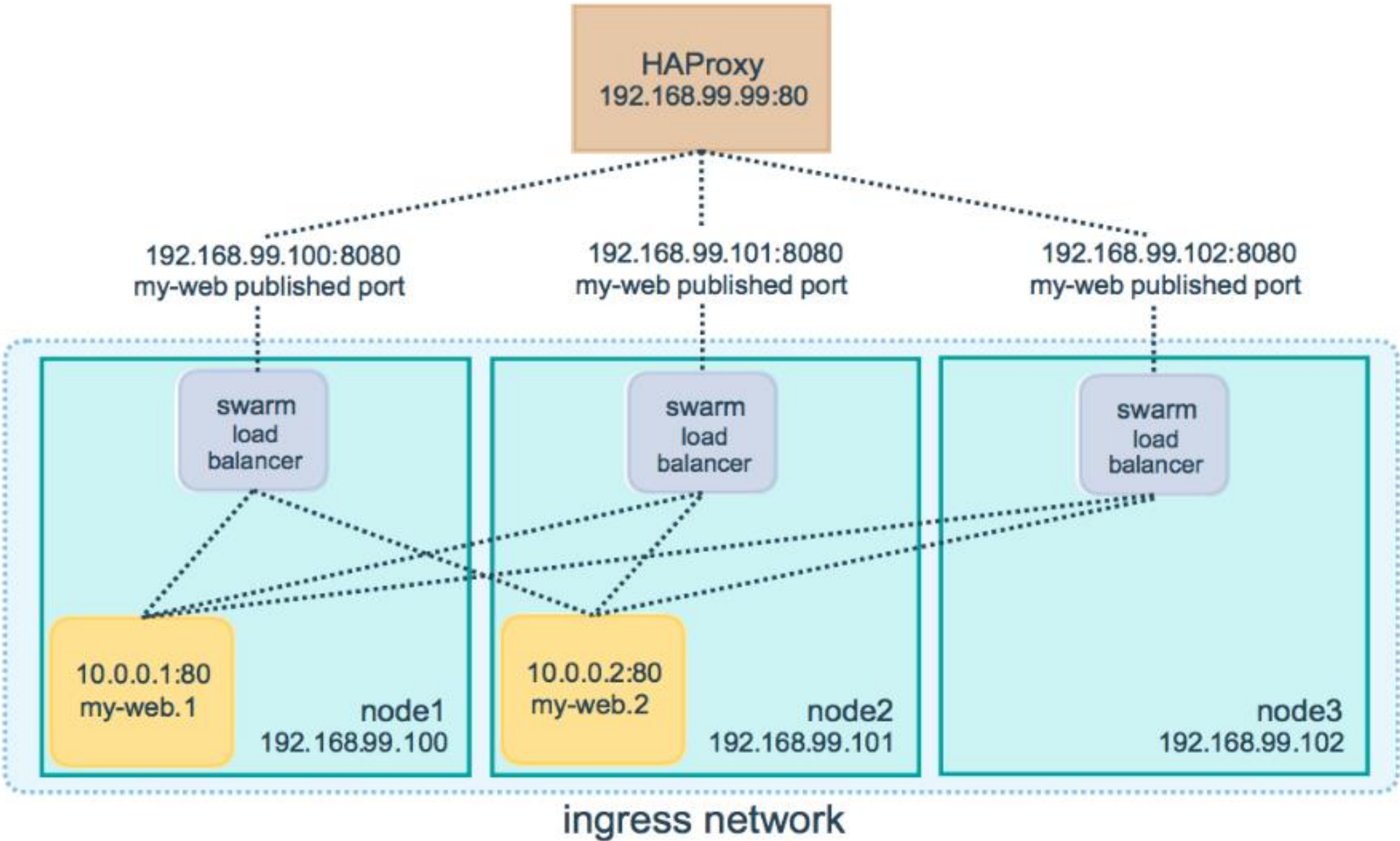
```
$ docker service create \
  --mount 'type=volume,src=<VOLUME-NAME>,dst=<CONTAINER-PATH>,volume-driver=local,volume-opt=type=nfs,volume-
opt=device=<nfs-server>:<nfs-path>,"volume-opt=o=addr=<nfs-address>,vers=4,soft,timeo=180,bg,tcp,rw"'
  --name myservice \
  <IMAGE>
```

Swarm容器集群管理

发布服务

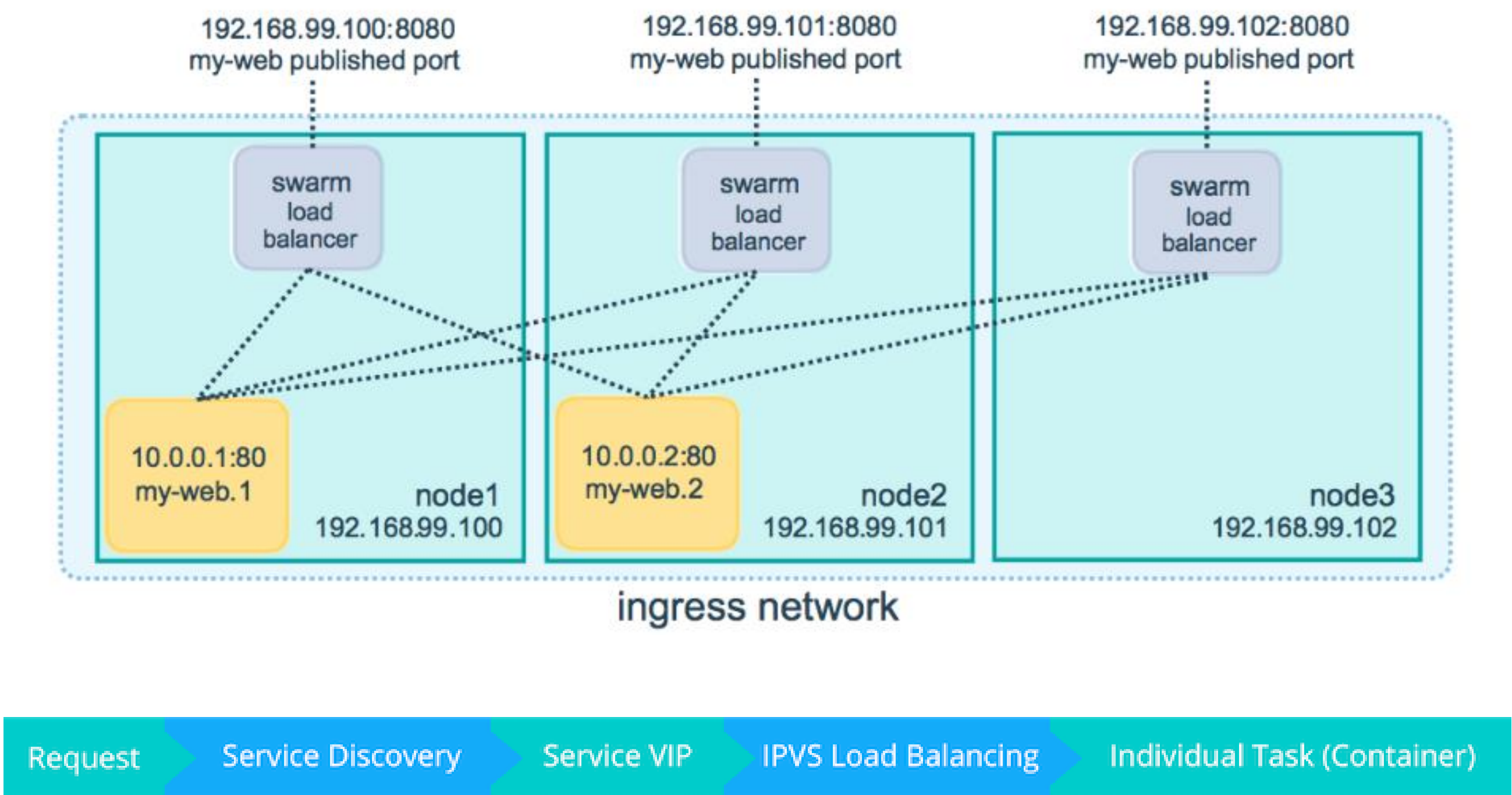


发布服务



Swarm容器集群管理

服务发现与负载均衡



服务发现：Swarm模式内置DNS组件，自动为每个服务分配DNS记录，然后服务的DNS名称在集群内的服务直接分发请求。

负载均衡：在Swarm集群中创建服务时，Ingress网络会自动为其分配一个虚拟IP(VIP)，在DNS解析时返回VIP，流入该VIP的流量将自动发送（IPVS）该服务的所以健康任务（容器）。

Swarm容器集群管理

服务发现与负载均衡

进容器查看DNS记录

```
nslookup hello
```

获取虚拟IP

```
docker service inspect -f '{{json .Endpoint.VirtualIPs}}' hello
```

设置DNS轮询模式

```
docker service create \
```

```
--replicas 3 \
```

```
--name my-web \
```

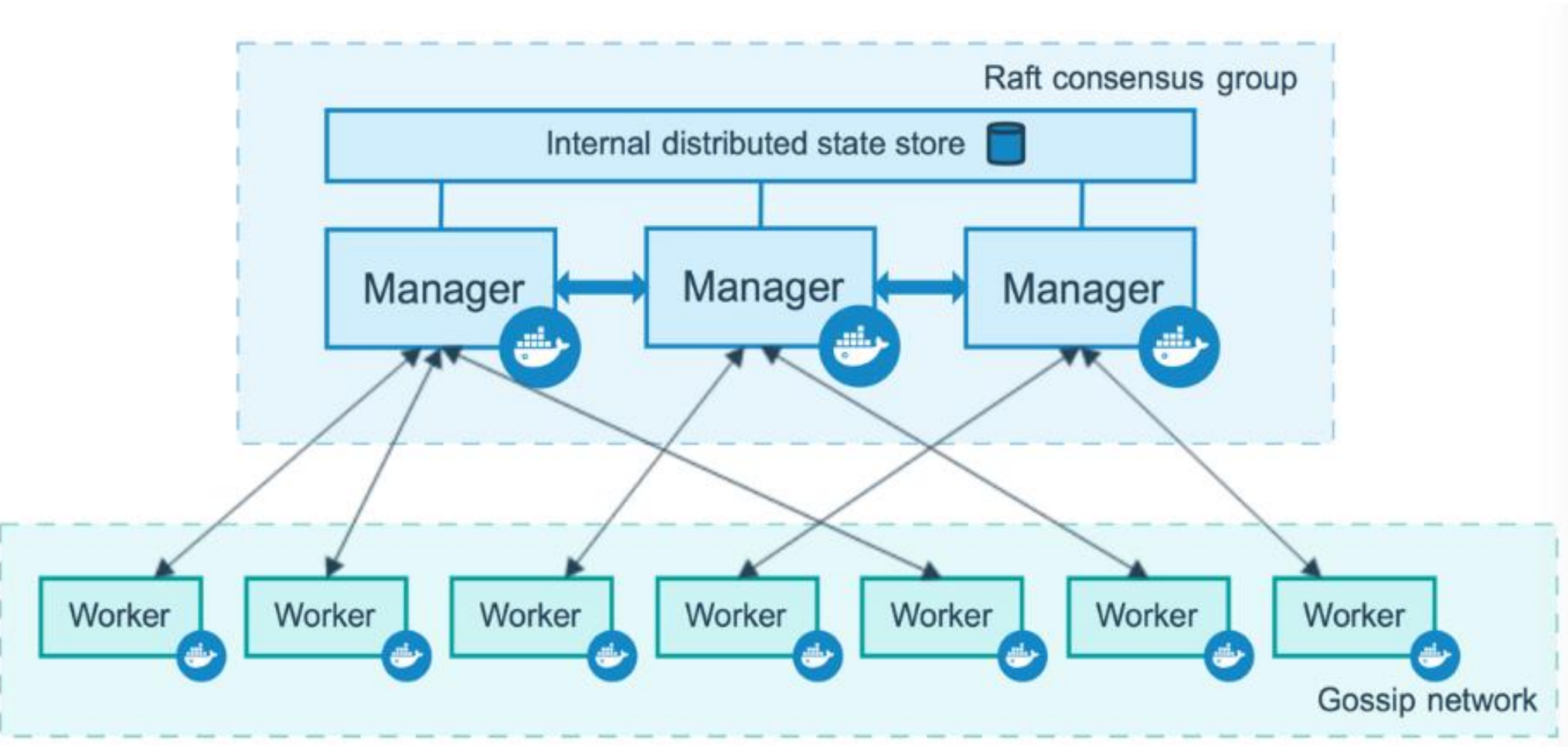
```
--network my-network \
```

```
--endpoint-mode dnsrr \
```

```
nginx
```

Swarm容器集群管理

高可用架构



Swarm Size	Majority	Fault Tolerance
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	3
9	5	4

Swarm容器集群管理

配置文件管理

1、生成一个基本的Nginx配置文件

```
# cat site.conf
server {
    listen      80;
    server_name localhost;
    location / {
        root    /usr/share/nginx/html;
        index   index.html index.htm;
    }
}
```

2、将site.conf保存到docker配置中

```
# docker config create site.conf site.conf
# docker config ls
```

3、创建一个Nginx并应用这个配置

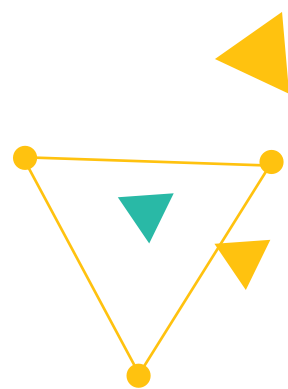
```
# docker service create \
    --name nginx \
    --config source=site.conf,target=/etc/nginx/conf.d/site.conf \
    --publish 8080:80 \
    nginx
```

应用案例 – 集群部署LNMP网站平台

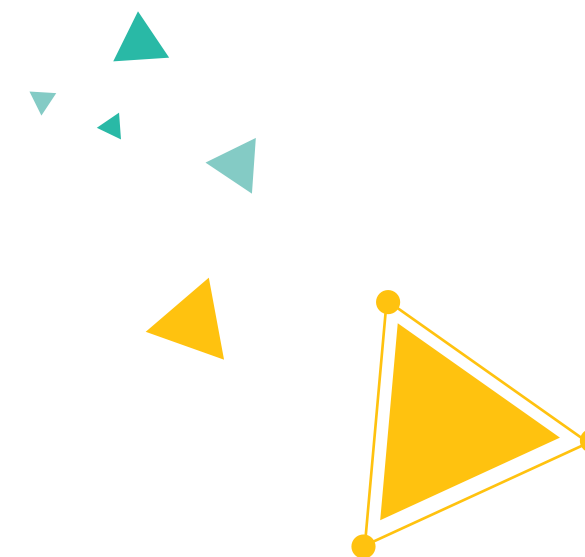
NOT SUPPORTED FOR DOCKER STACK DEPLOY

The following sub-options (supported for `docker compose up` and `docker compose run`) are *not supported* for `docker stack deploy` or the `deploy` key.

- `build`
- `cgroup_parent`
- `container_name`
- `devices`
- `dns`
- `dns_search`
- `tmpfs`
- `external_links`
- `links`
- `network_mode`
- `security_opt`
- `stop_signal`
- `sysctls`
- `usersns_mode`



谢谢



微信扫一扫



DevOps技术栈

专注于分享DevOps工具链
及经验总结。

Docker/K8s技术学员群: [397834690](#)