

# Rolling Update 还看Docker原生支持

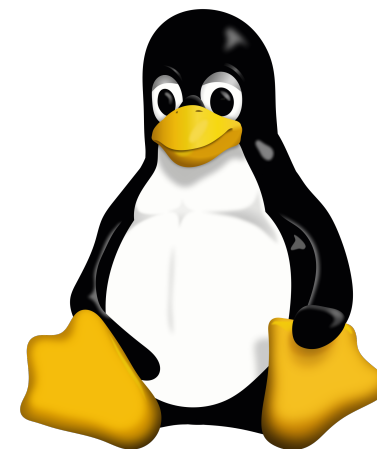
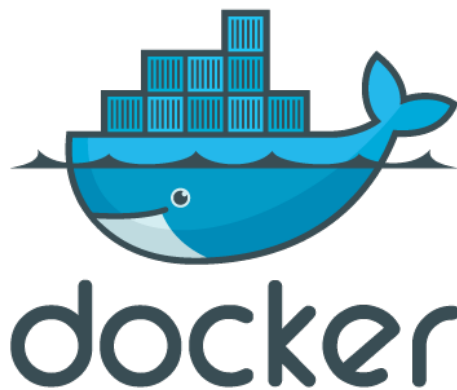
Allen Sun

# About me

- DaoCloud
- Docker
- Linux
- Open Source
- GitHub: allencloud



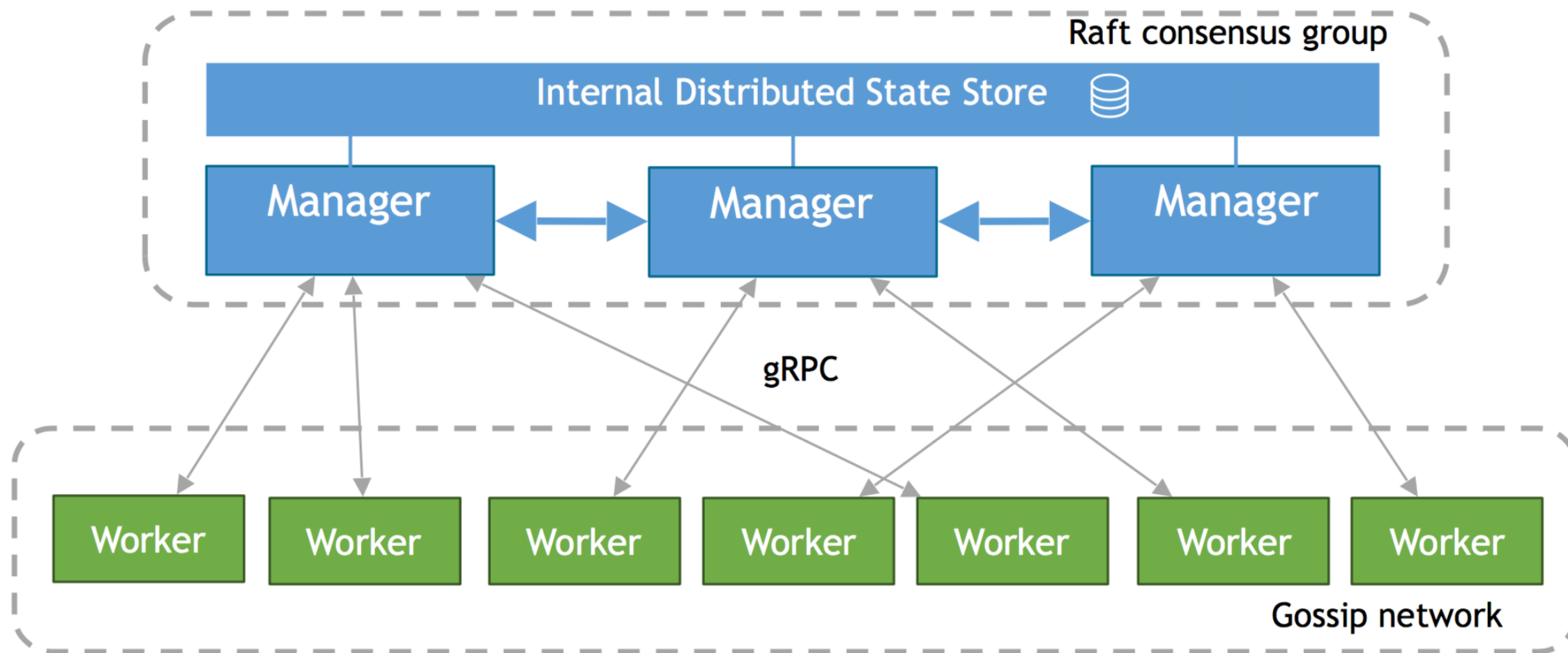
DaoCloud



# Agenda

- Docker Swarm Mode
- Rolling Update in Swarm
- Demo

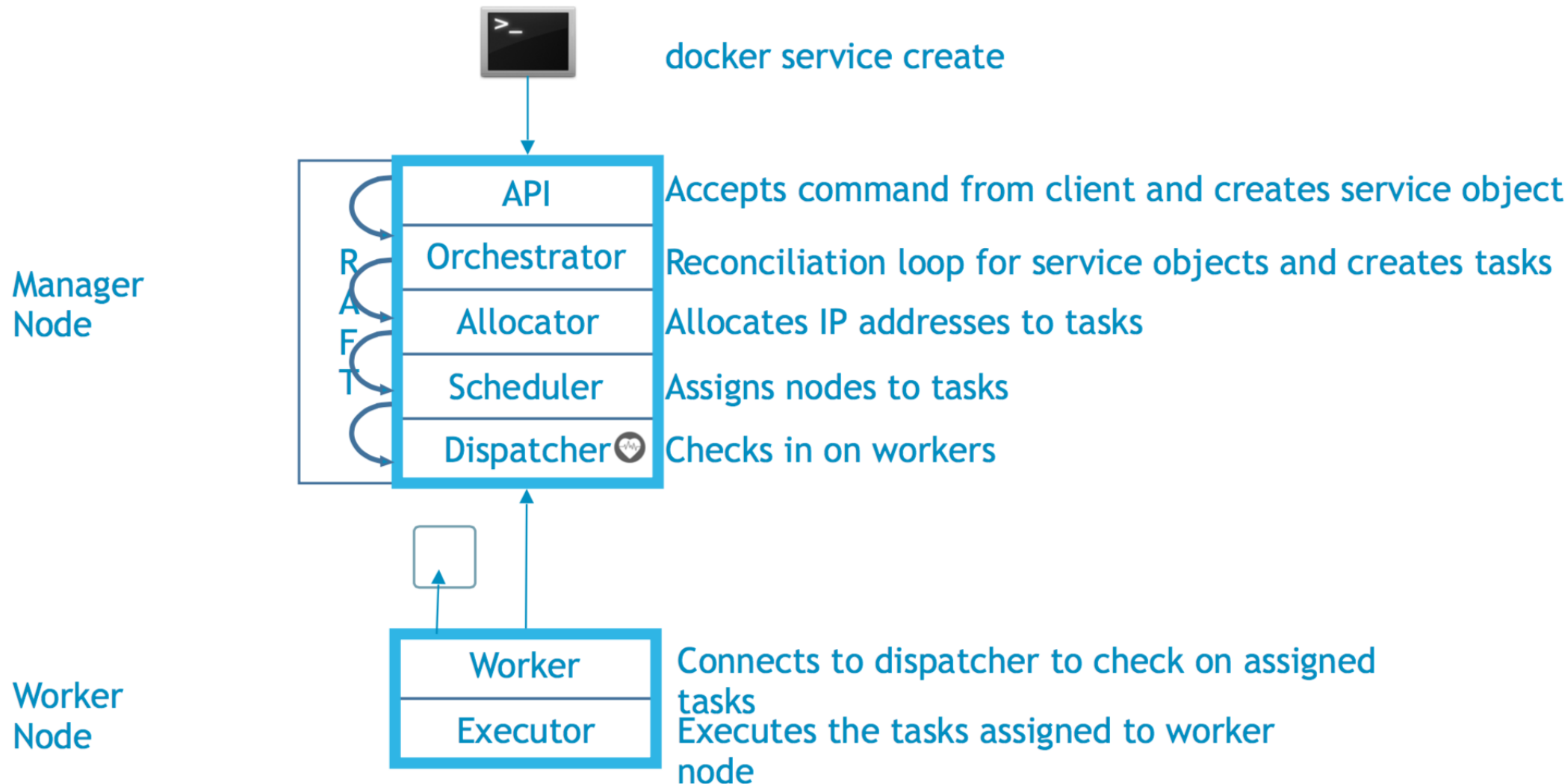
# SWARM MODE的架构



# SWARM MODE 服务

- 服务（Service）作为集群的操作对象，服务由任务（task）来实现，容器作为实现任务的一个执行方式
- 服务可以指定任务数量，也可以是全局任务（每个节点运行一个）
- 调度器管理任务的目标状态（desired state），分配资源给任务，选择节点来执行任务
- 节点支持将任务的状态推动到目标状态，反馈状态给管理节点
- 支持服务配置更新，滚动更新，回滚
- 内置overlay网络，DNS服务发现，负载均衡

# SWARM功能模块



# ROLLING UPDATE

- 迭代（更新一个新版本）
- 业务连续性
- 风险的规避方式



DaoCloud

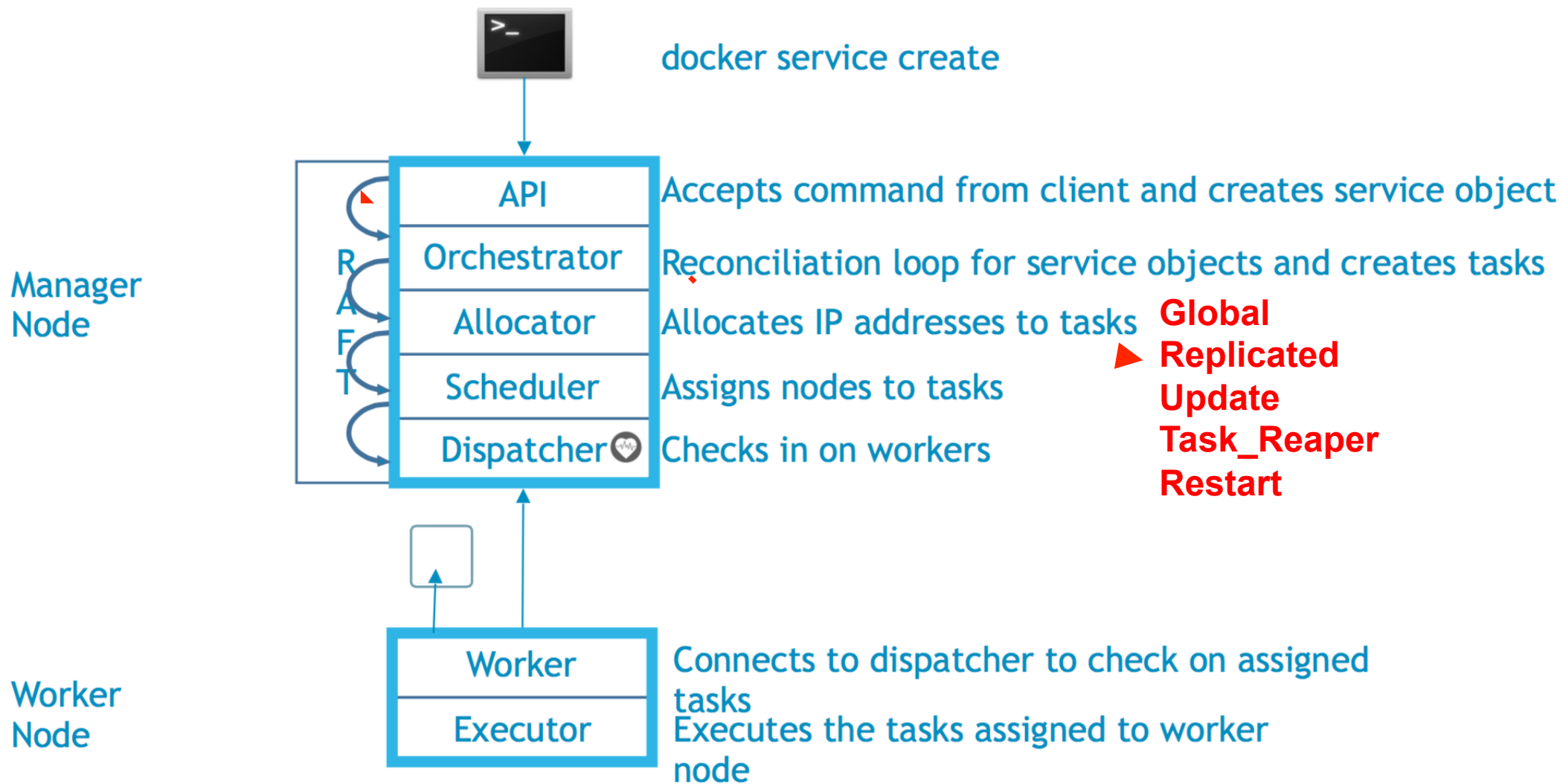
# ROLLING UPDATE & ORCHESTRATION

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

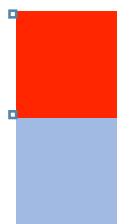
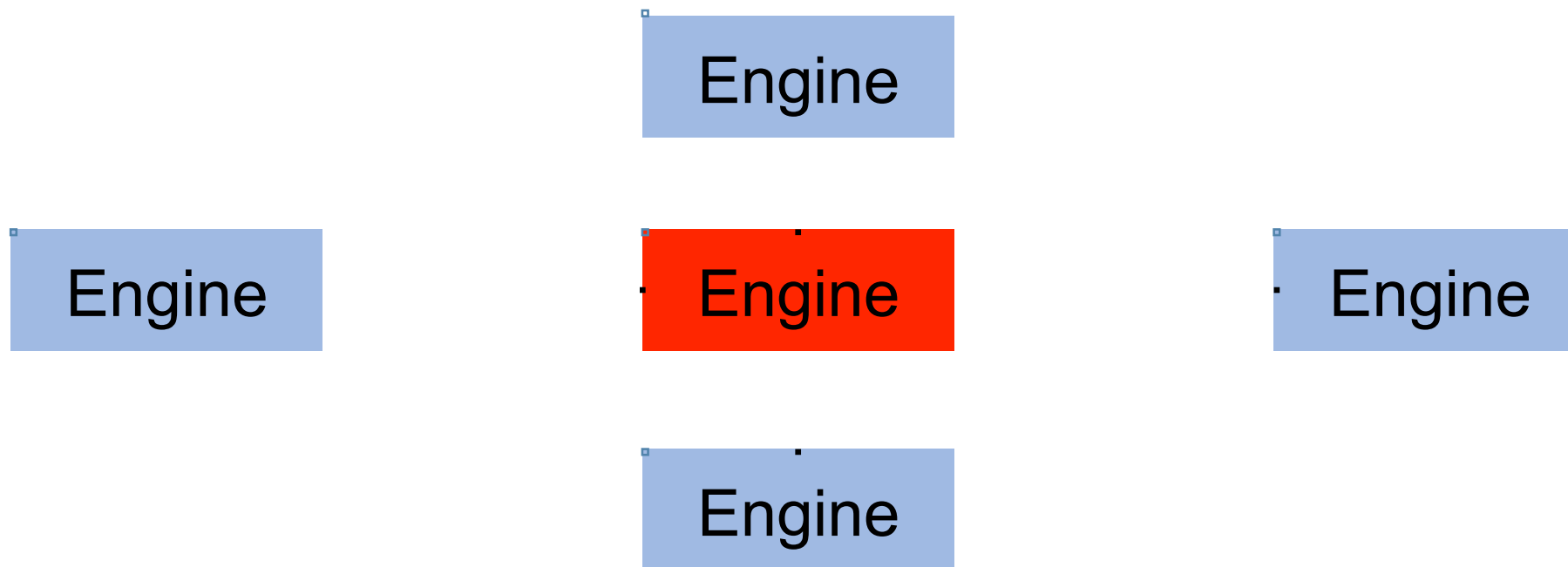
- ① 停止相应的任务
- ② 更新停止任务的信息
- ③ 为更新后的任务启动新容器
- ④ 若容器启动成功，进行下一次更新
- ⑤ 若容器启动失败，中止更新



# SWARM功能模块



# SWARM MODE



```
$ docker swarm init
```

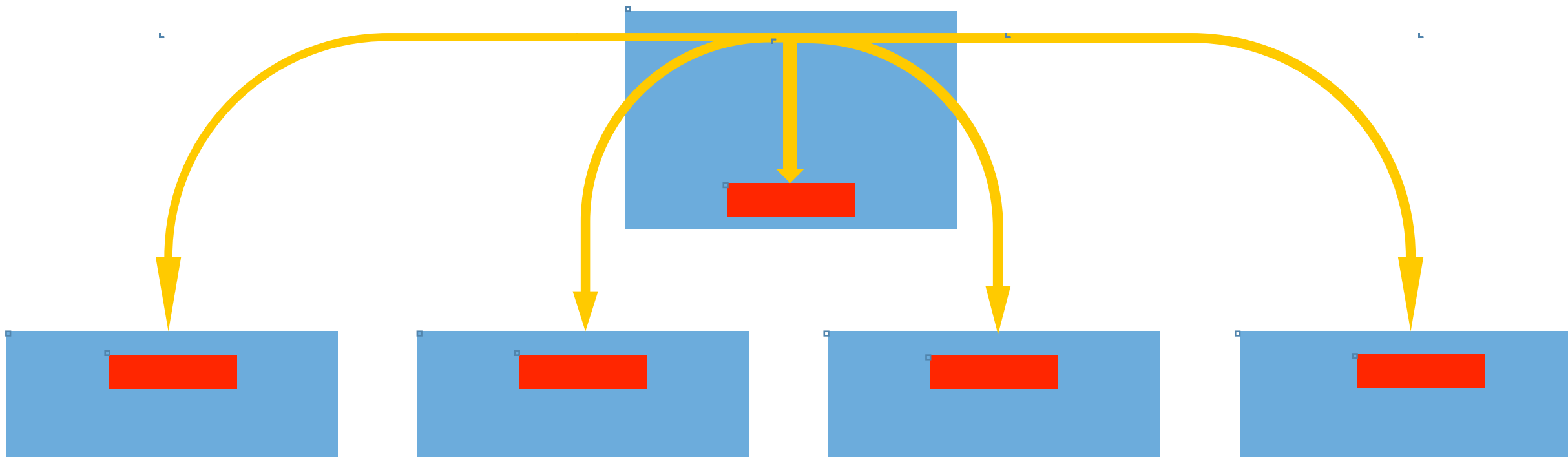
```
$ docker swarm join <manager ip>:2377
```



```
docker service create --name allen --replicas=5 -p 30001:80 alpine sleep 12345
```

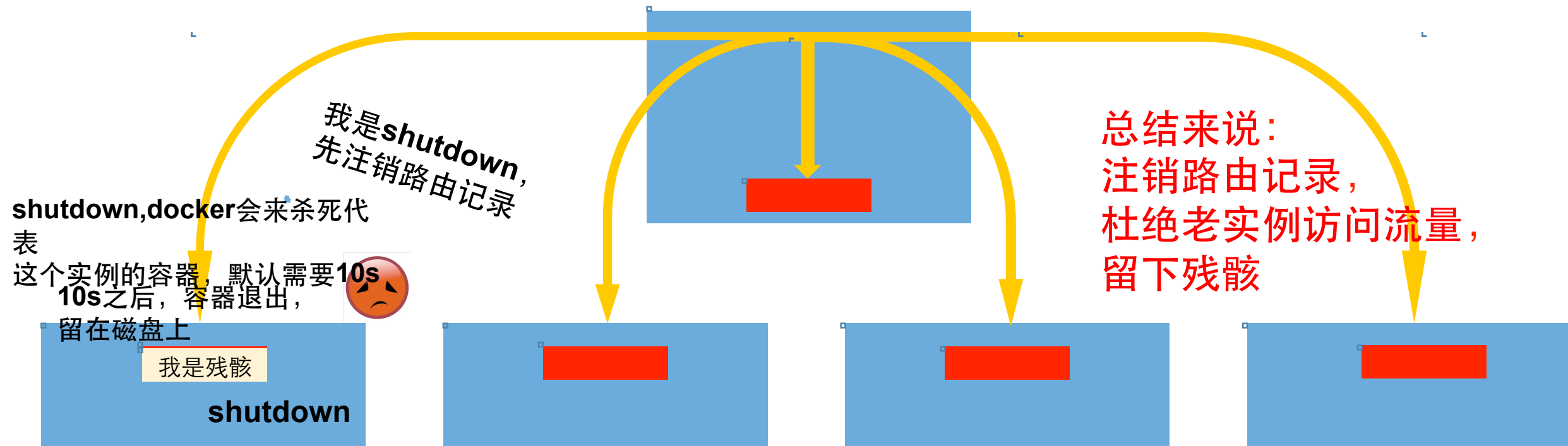
1. 创建指定数量的服务实例，这里是5

2. 支撑网络功能，每个服务实例完成服务注册，确保可以实现依据LVS完成的负载均衡



用户需要在现有系统中，使用新发布的alpine:v2.0版本来滚动升级线上的应用

```
docker service update --update-delay=20s --update-parallelism=1 --image alpine:v2.0 allen
```

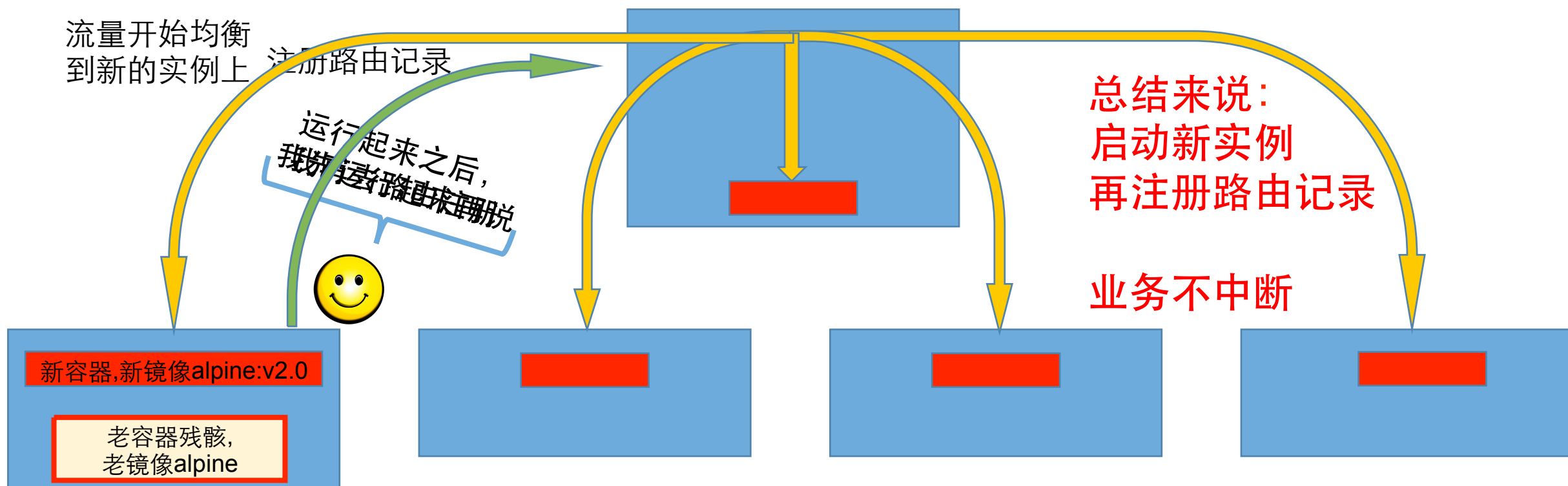


第一步:

随机找到一个运行中的实例（原因是更新并发数update-parallelism为1，所以只找一个），对其的状态设置为shutdown

用户需要在现有系统中，使用新发布的alpine:v2.0版本来滚动升级线上的应用

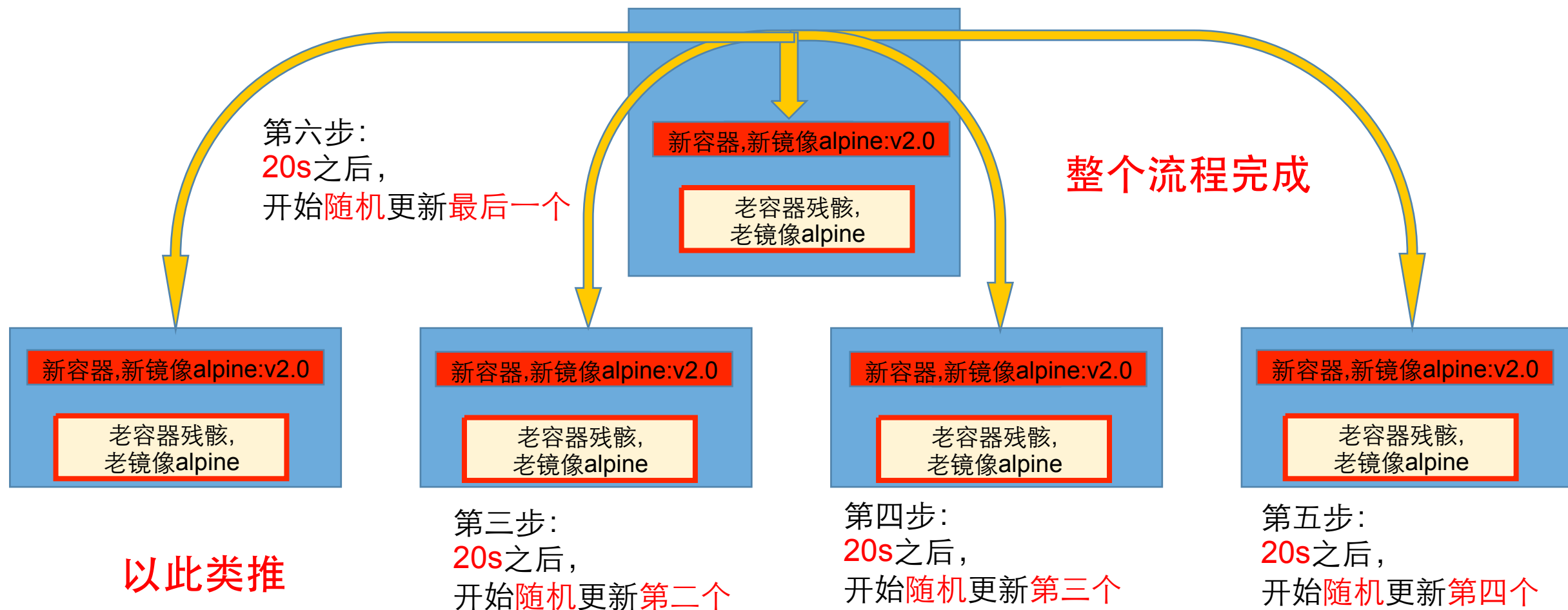
```
docker service update --update-delay=20s --update-parallelism=1 --image alpine:v2.0 allen
```



第二步:  
在shutdown的实例基础上,  
创建一个新的task

用户需要在现有系统中，使用新发布的alpine:v2.0版本来滚动升级线上的应用

docker service update ~~--update-delay=20s~~ --update-parallelism=1 --image alpine:v2.0 allen



# DEMO

# Q&A