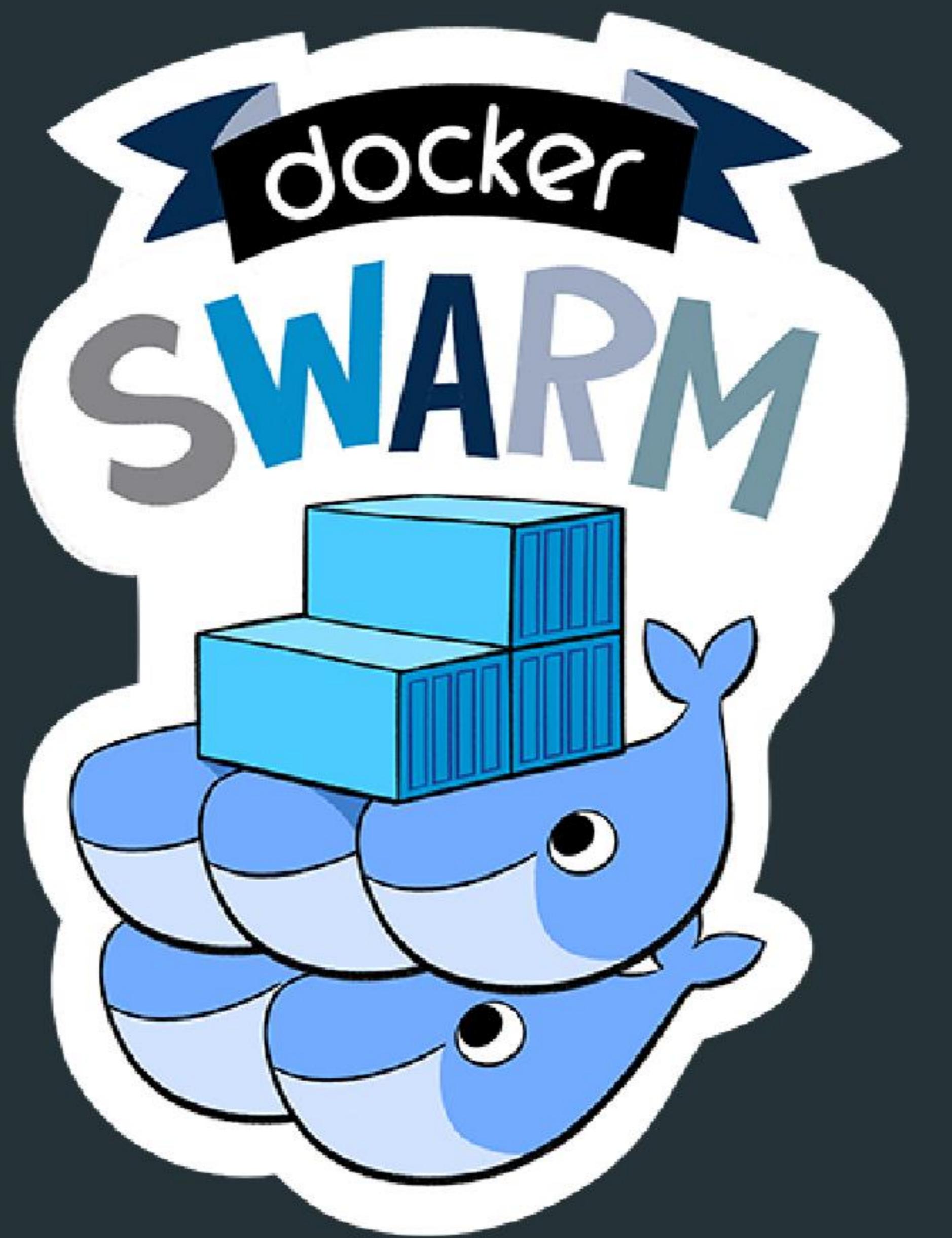


Docker Swarm

the Docker-native clustering system



What's new in Swarm 1.1

@线超博

自我介绍

个人简介：

- 线超博
- 华为IT云计算架构与设计部 高级工程师
- Docker社区Swarm项目Maintainer

联系方式：

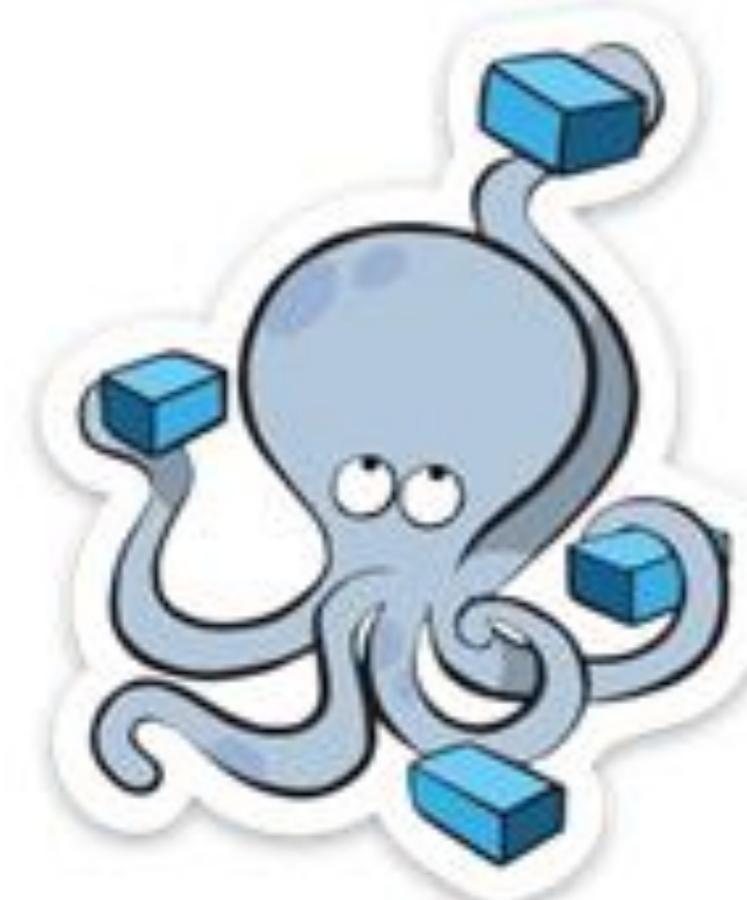
- Email : xianchaobo@gmail.com
- 微信 : 342288824
- 微博 : @线超博



Docker社区



Toolbox



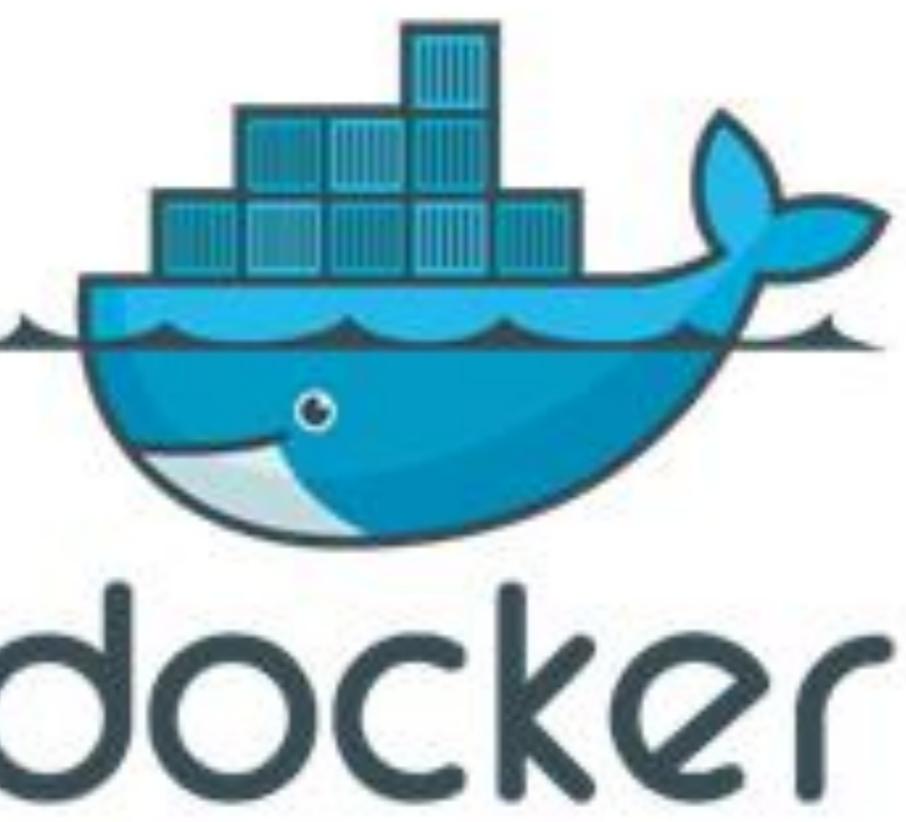
Compose



Swarm



Machine



docker



Libcontainer



Libnetwork

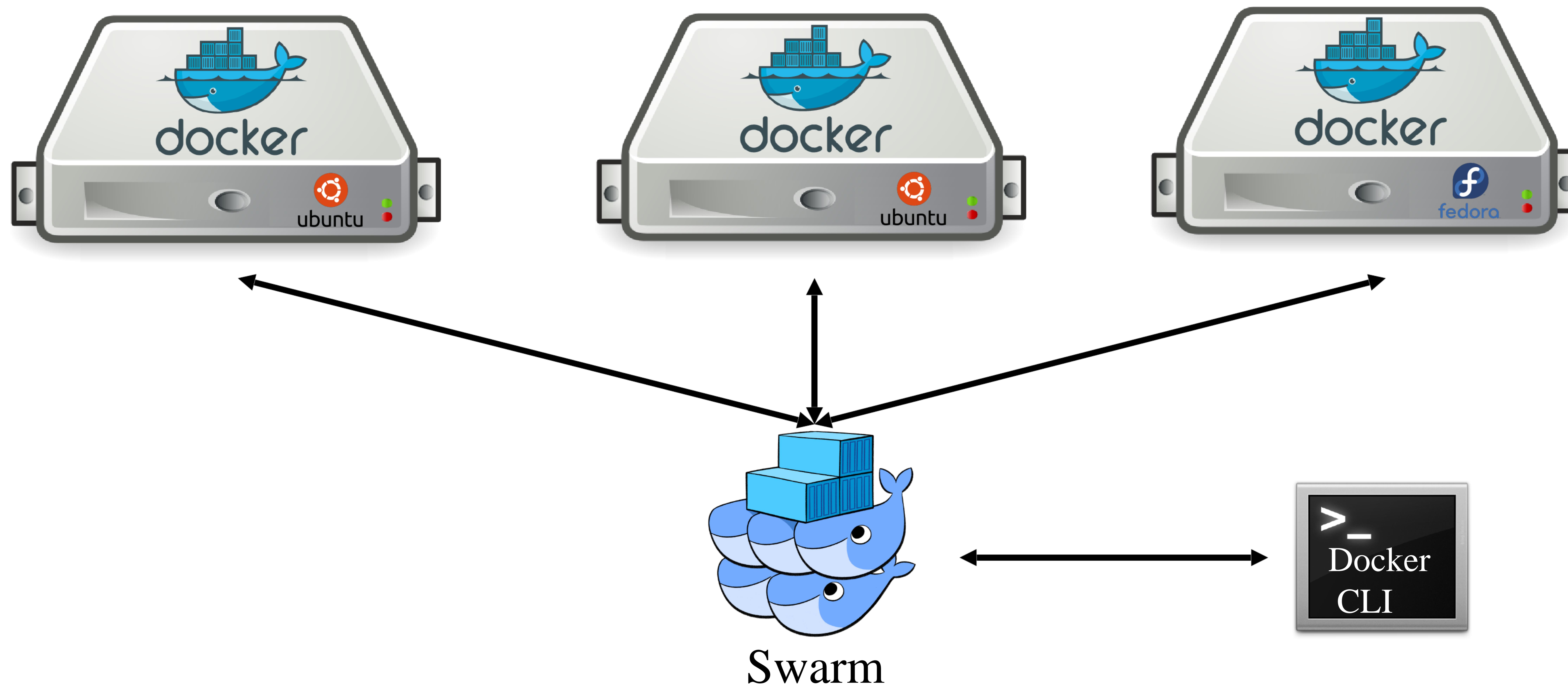


Notary



Registry

With Docker Swarm




Swarm特点

- 对外以Docker API接口呈现
- 轻量，易上手易部署
- Batteries included but swappable
- 对Docker命令参数支持比较完善



Swarm框架结构



Setup: With swarm agent



Setup using the hosted discovery service

/!\ Not to be used in production, for testing only /!\\

- Create a cluster:

```
$ swarm create
```

- Add nodes to a cluster:

```
$ swarm join --advertise=<engine_ip>:<engine_port> token://<token>
```

- Start Swarm

```
$ swarm manage <...> token://<token>
```



Setup using your own KV store

- Add nodes to a cluster:

```
$ swarm join --advertise=<engine_ip>:<engine_port> \  
  consul://<ip_consul>:<port_consul>
```

- Start Swarm

```
$ swarm manage <...> consul://<ip_consul>:<port_consul>
```

You can also use etcd or zookeeper



Setup using a file (static list of nodes)

- Add nodes to the file:

```
$ echo 10.0.0.1:2375 > my_cluster  
$ echo 10.0.0.2:2375 >> my_cluster  
$ echo 10.0.0.3:2375 >> my_cluster
```

- Start Swarm

```
$ swarm manage <...> file://my_cluster
```



Setup: Without swam agent



Setup using your own KV store

- Configure networking on engine

```
$ docker daemon --cluster-advertise=<engine_ip>:<engine_port> \  
--cluster-store=consul://<ip_consul>:<port_consul>
```

- Start Swarm

```
$ swarm manage <...> --discovery-opt kv.path=docker/docker \  
consul://<ip_consul>:<port_consul>
```

You can also use etcd or zookeeper



Docker Swarm internals



Resource Management

- Memory

```
$ docker run -m 1g ...
```

- CPU

```
$ docker run -c 1 ...
```

- Ports

```
$ docker run -p 80:80 ...
```



Swarm Scheduler

2 steps:

- 1- Apply filters to exclude nodes
 - constraints
 - affinity
 - dependency
 - health
 - ports
- 2- Use a strategy to rank and pick the best node
 - binpack
 - spread
 - random



Constraints

- Standard constraints induced from docker info

```
docker run -e "constraint:operatingsystem==*fedora*" ...
```

```
docker run -e "constraint:storagedriver==*aufs*" ...
```

- Custom constraints with host labels

```
docker daemon --label "region=china"
```

```
docker run -e "constraint:region==china" ...
```

- Pin a container to a specific host

```
docker run -e "constraint:node==node-2" ...
```



Affinities

- Containers affinities

```
docker run --name web nginx
```

```
docker run -e "affinity:container==web" logger
```

- Containers Anti-affinities

```
docker run --name redis-master redis
```

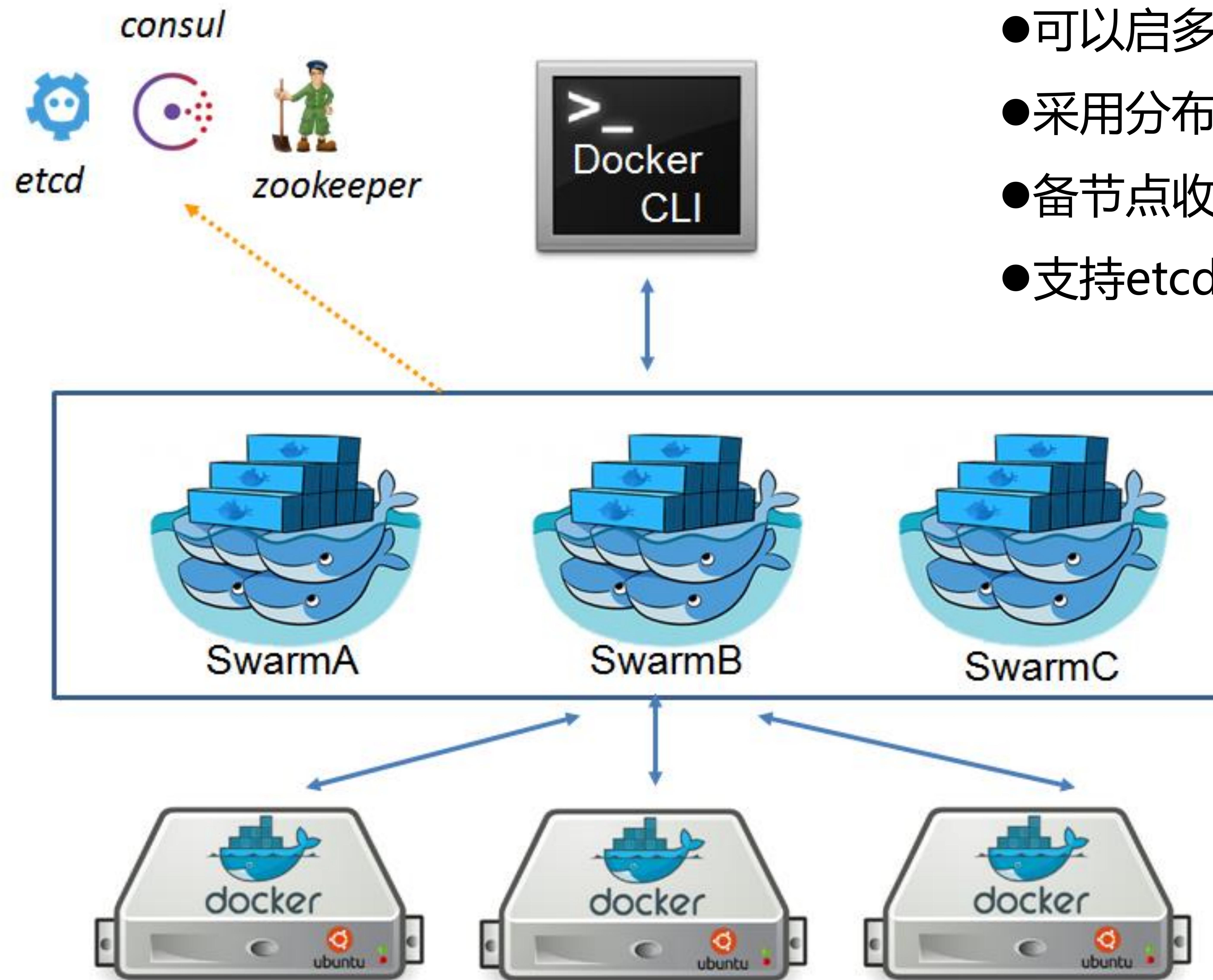
```
docker run --name redis-slave -e "affinity:container!=redis*" ...
```

- Images affinities

```
docker run -e "affinity:image==redis" redis
```



High Available Scheduler



- 可以启多个Manager实例，主备方式实现HA
- 采用分布式锁实现选主过程
- 备节点收到消息，会转发给主节点
- 支持etcd、zookeeper、consul

New in Swarm 1.1

- Improved node management
- Rescheduling (EXPERIMENTAL)
- New events



Improved node management: docker info

Nodes: 3

(unknown): 10.0.0.9:2375

...

Status: Pending

node-1: 10.0.0.1:2375

Status: Unhealthy

...

Error: Cannot connect to the docker engine endpoint

UpdatedAt: 2016-02-09T19:52:56Z

node-2: 10.0.0.0:2375

Status: Healthy

...

Labels: kernelversion=4.2.0-23-generic, operatingsystem=Ubuntu 14.04.3 LTS, ...

Error: (none)

UpdatedAt: 2016-02-09T19:52:56Z




Rescheduling


- Experimental feature
swarm --experimental manage ...
- On node failure
docker run -e “reschedule:on-node-failure” ...




On-failure rescheduling



On-failure rescheduling



On-failure rescheduling



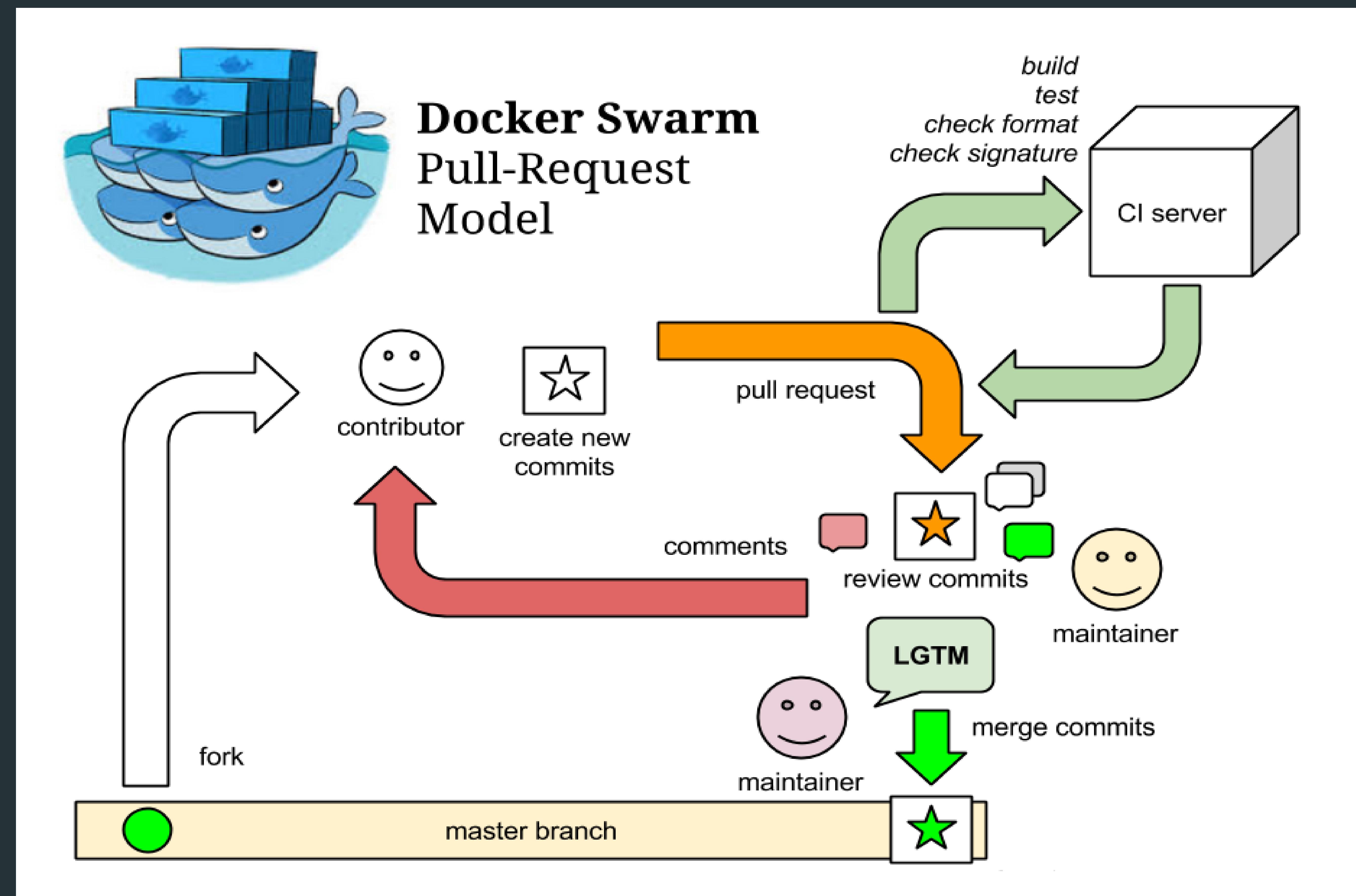
New events

```
$ docker events
...
2016-02-09T12:02:01 container create XXX (com.docker.swarm.id=YYY, image=busybox, node.addr=10.0.0.1:2375, node.name=node-1)
...
2016-02-09T12:02:01 network connect ZZZ (node.name=node-1, type=bridge, container=XXX, name=bridge, node.addr=10.0.0.1:2375,)
...
2016-02-09T12:03:10 swarm engine_connect (node.name=node-2, node.addr=10.0.0.2:2375)
...
```



Swarm 社区


- Github地址：<http://github.com/docker/swarm>
- IRC：[#docker-swarm on freenode](#)
- 版本周期：跟着docker一起发布新版本，采用迭代开发，每两个星期一个迭代

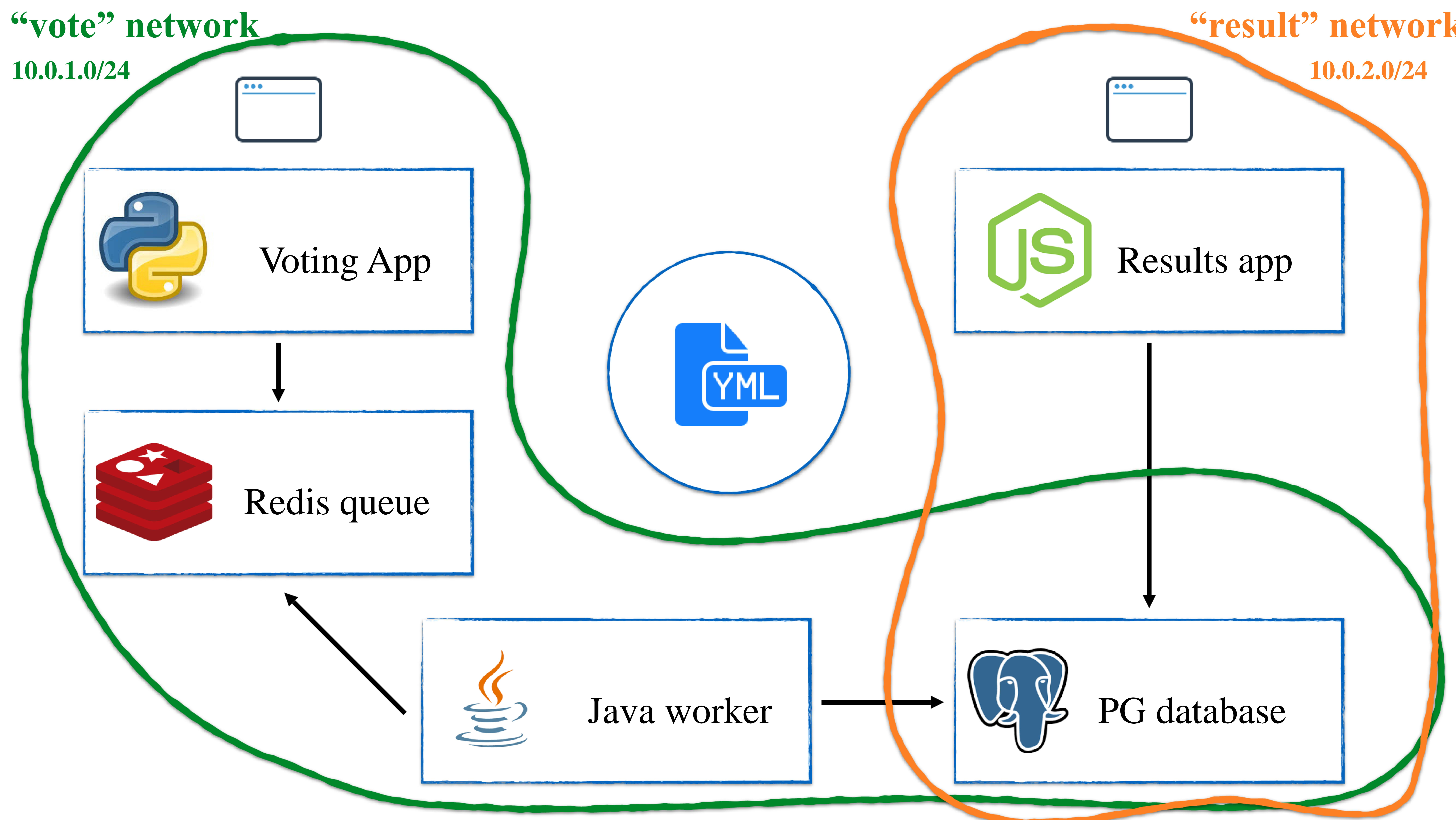


Demo

1. swarm环境搭建演示
2. 容器HA演示
3. compose + swarm + overlay演示







Thank You. Questions?

<http://github.com/docker/swarm>

#docker-swarm on freenode

