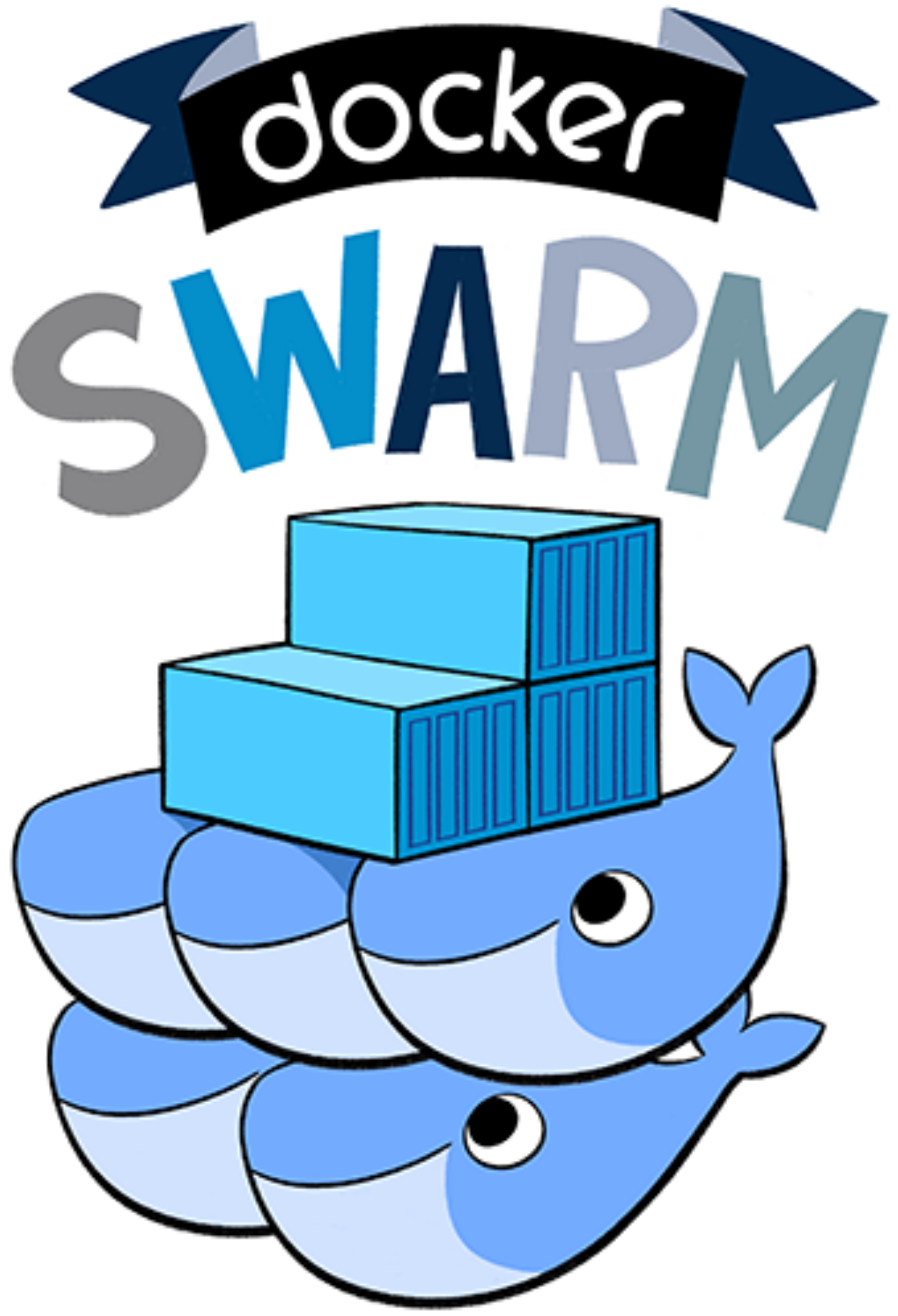
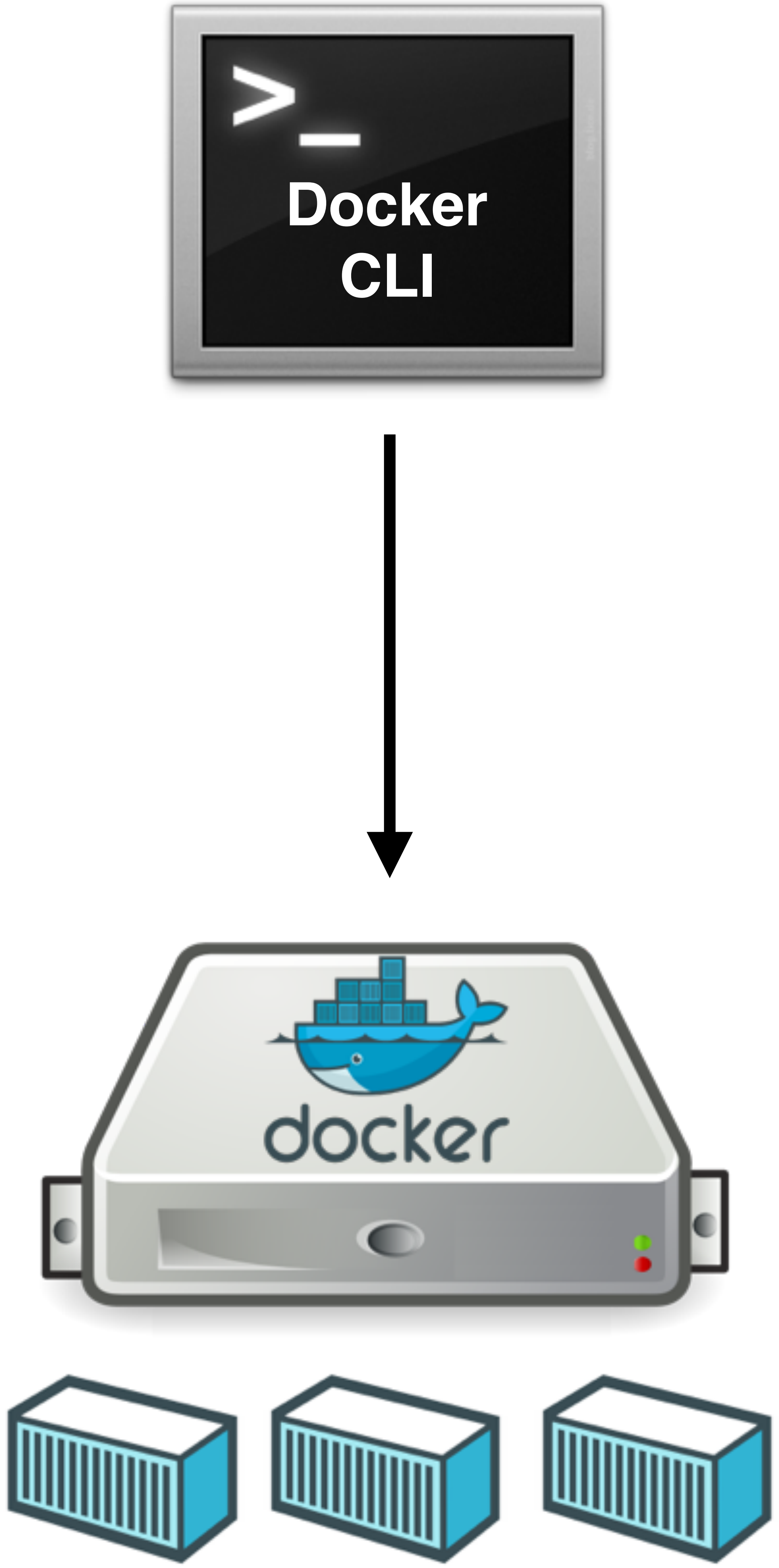
Swarm goes stable and v1.0

with a cute new logo



@abronan

Single docker engine



Multiple Docker Engines == Swarm

etcd

zookeeper

consul

libkv

Simple File

Docker Hub
Token service

Machine
discovery

CLI

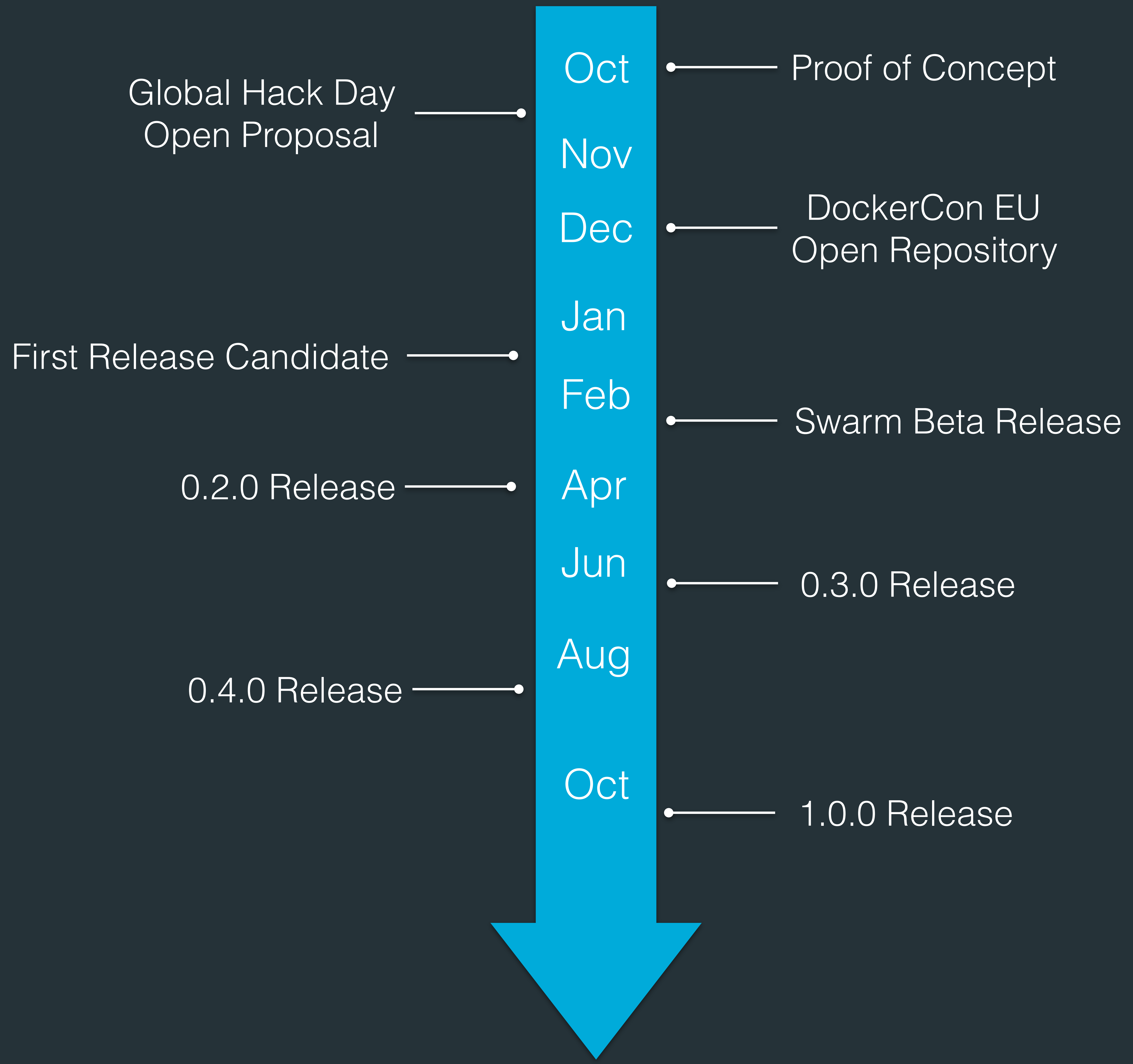
Docker
CLI

Managers

Agents

>_
docker
SWARM
docker
SWARM
docker
SWARM
docker
docker
docker
docker
docker

Timeline



Oct

Proof of Concept

Global Hack Day
Open Proposal

Nov

Dec

DockerCon EU
Open Repository

Jan

First Release Candidate

Feb

Swarm Beta Release

0.2.0 Release

Apr

Jun

0.3.0 Release

Aug

0.4.0 Release

Oct

1.0.0 Release

Swarm in General

- **Turns a set of Docker Engines into a single pool of resources**
- **Supports the Docker REST API** (99%)
- **Resource management** (CPU, Mem, Networking)
- **Advanced scheduling** with constraints and affinities
- Multiple **Discovery Backends** (hub, etcd, consul, zookeeper)
- **TLS**: Encryption & Authentication
- **Multi Tenancy** / Leader Election

New release highlights

- **Improved scheduler**
- **Multi-Host Networking**
  - network management, overlay networking (vxlan)
- **Volume management - using volume plugins**
  - stateful containers, distributed volumes (Ceph, GlusterFS, ZFS/Flocker)
- **Quality of life improvements**
  - build-time constraints (ARG), unless-stopped restart policy, fixed concurrent pull
- **Better integration with Compose**
- **More tests / code cleanup / bug fixes**

Production ready

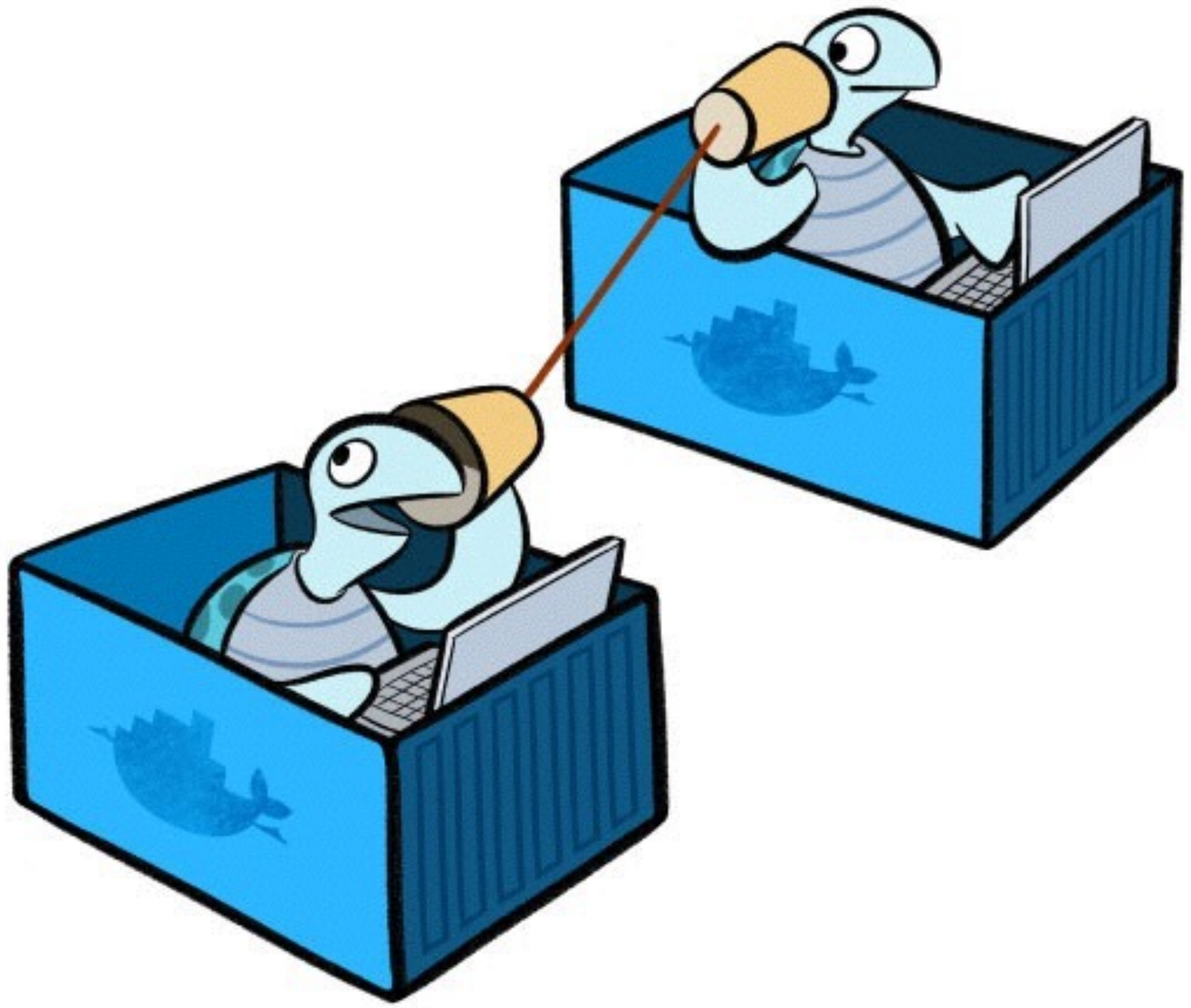
- **Stable, can be used in production environment**

- **Stress tested on 1000 EC2 nodes**

- **Still lots of improvements to come, on the roadmap for 1.1:**
  - container rebalancing
  - global scheduling

- **Rationale: Keep it simple to use/operate. Hear back from users and only include what is wanted/needed by the community**
  - example: networking, build time constraints, etc.

Multi-Host Networking



Multi-Host networking

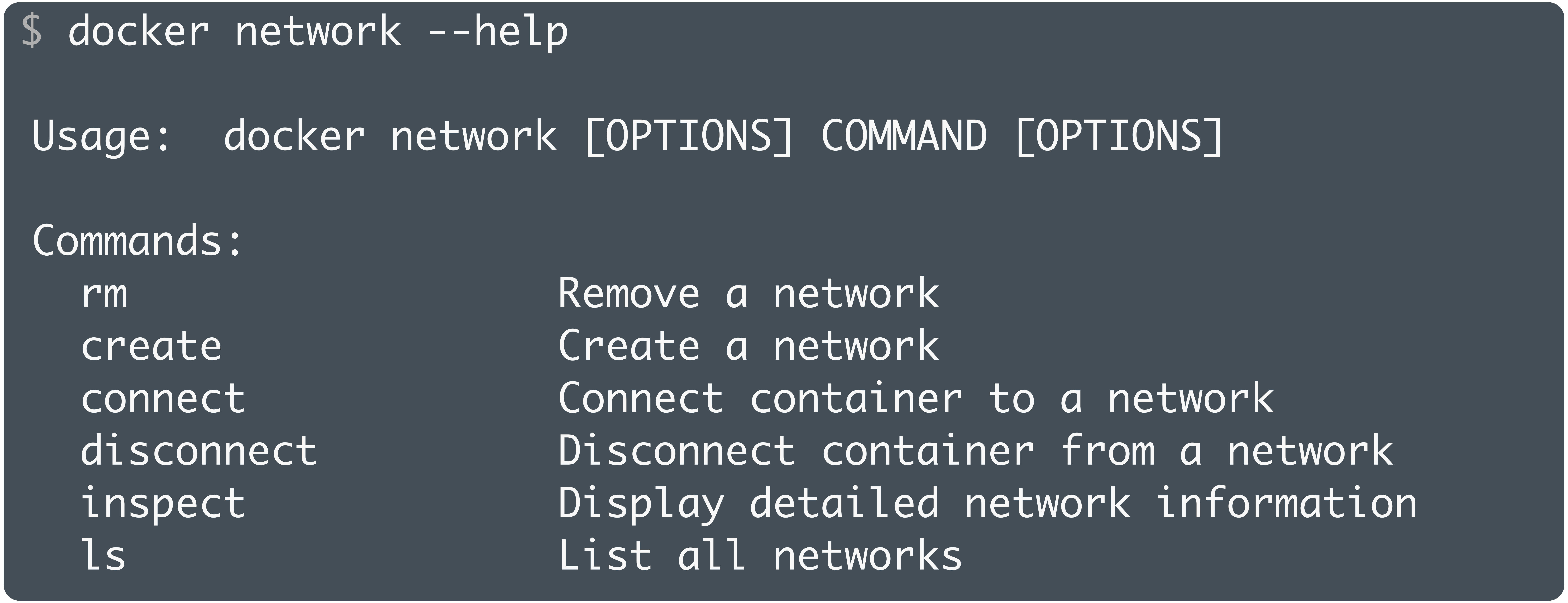
- Announced as part of the experimental release in DockerCon SF June

- Now in docker **stable**!

- Allows you to create overlay networks between containers using the **vxlan** driver

- Each container connected to the same overlay network are able to see and discover each other

A wild docker network appears



```
$ docker network --help

Usage:  docker network [OPTIONS] COMMAND [OPTIONS]

Commands:
  rm                    Remove a network
  create                Create a network
  connect               Connect container to a network
  disconnect            Disconnect container from a network
  inspect               Display detailed network information
  ls                    List all networks
```

Swarm / Networking

CLI

Managers

Agents

Docker CLI

Machine discovery

etcd

consul

zookeeper

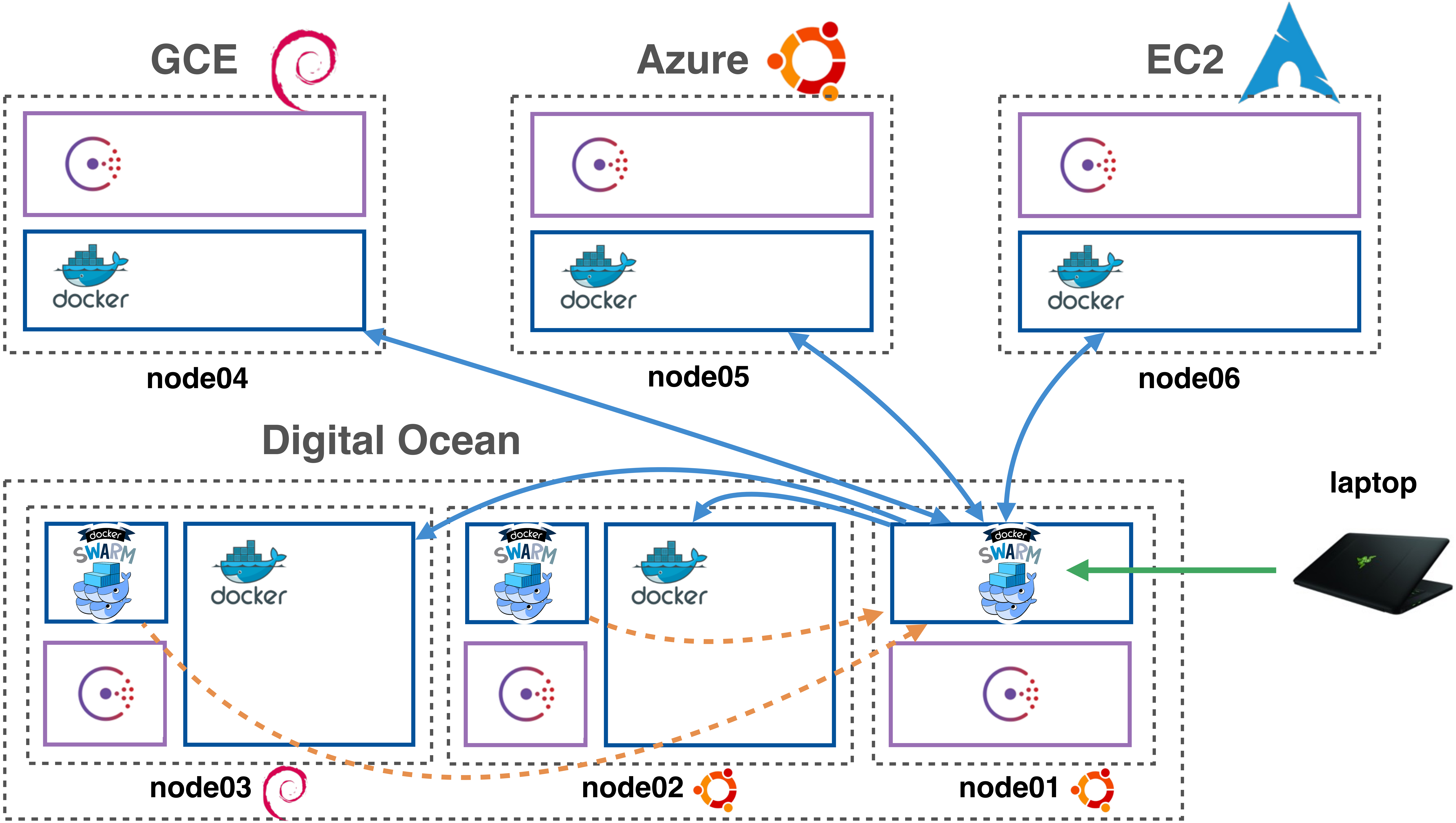
libkv

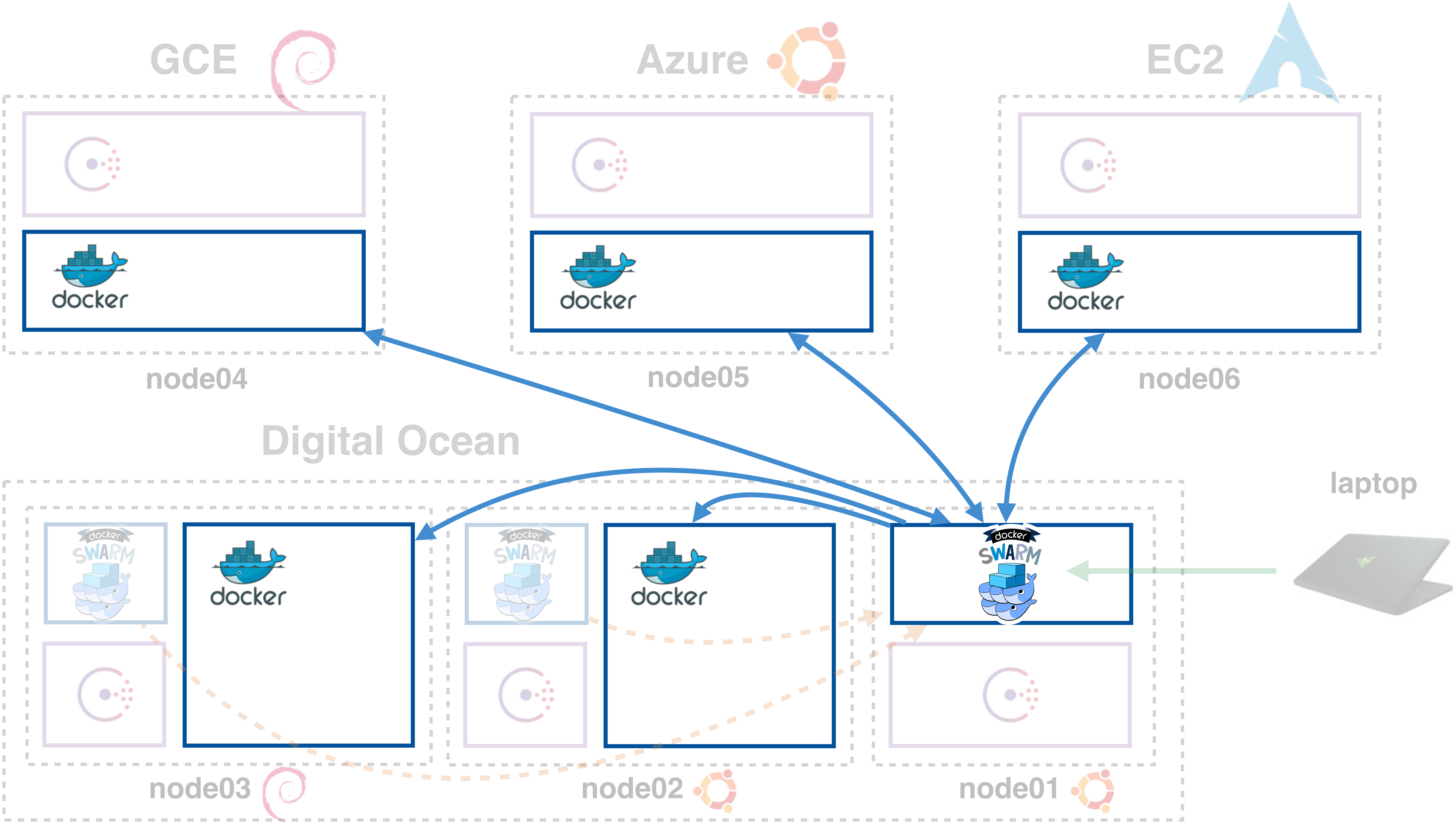
Simple File

Docker Hub
Token service

>_
docker
SWARM
docker

Overlay networking *(vxlan)*


GCE
Azure
EC2
docker
docker
docker
node04
node05
node06
Digital Ocean
laptop
docker
SWARM
docker
docker
SWARM
docker
docker
SWARM
node03
node02
node01



GCE

Azure

EC2

node04

node05

node06

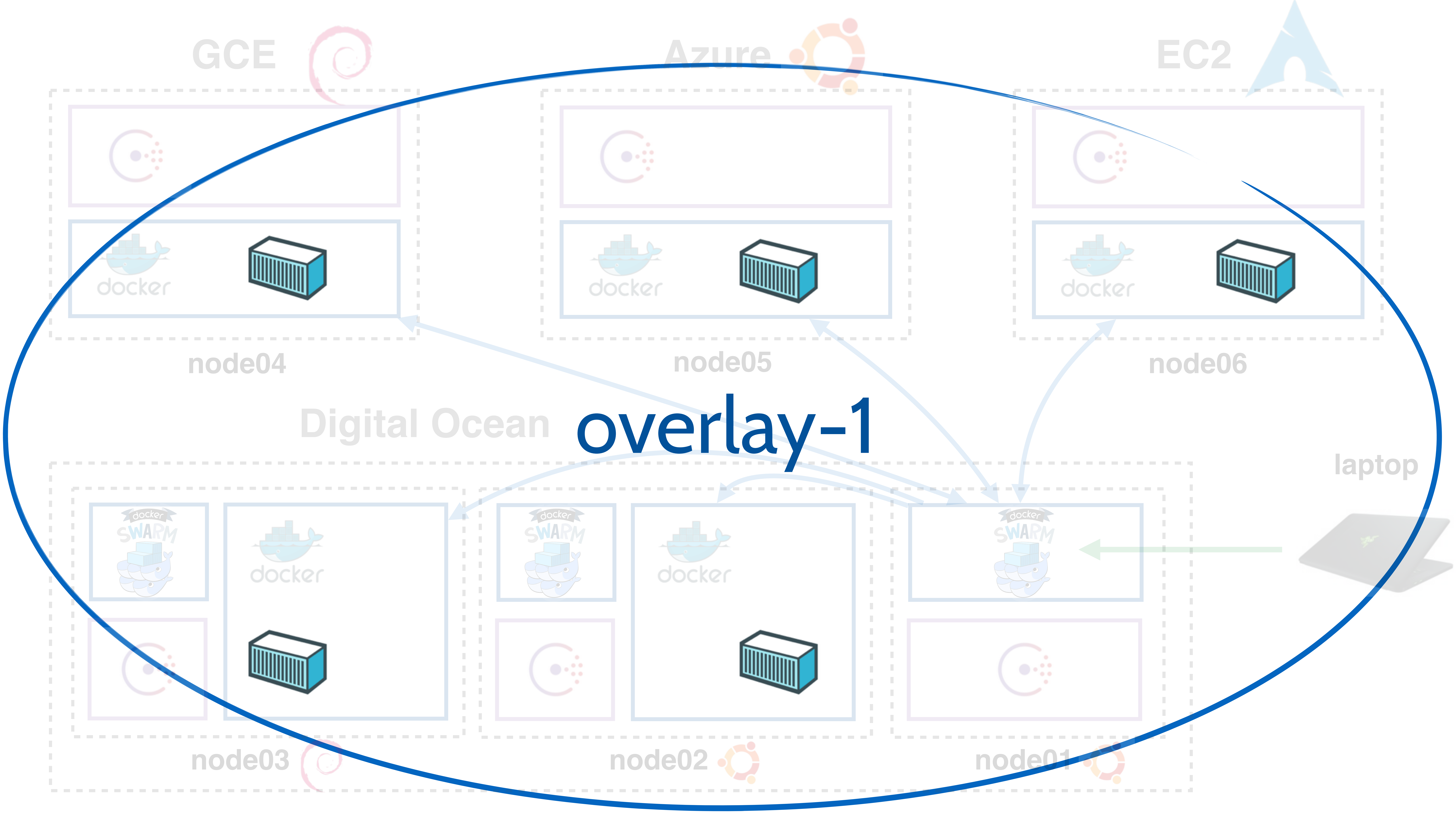
Digital Ocean

laptop

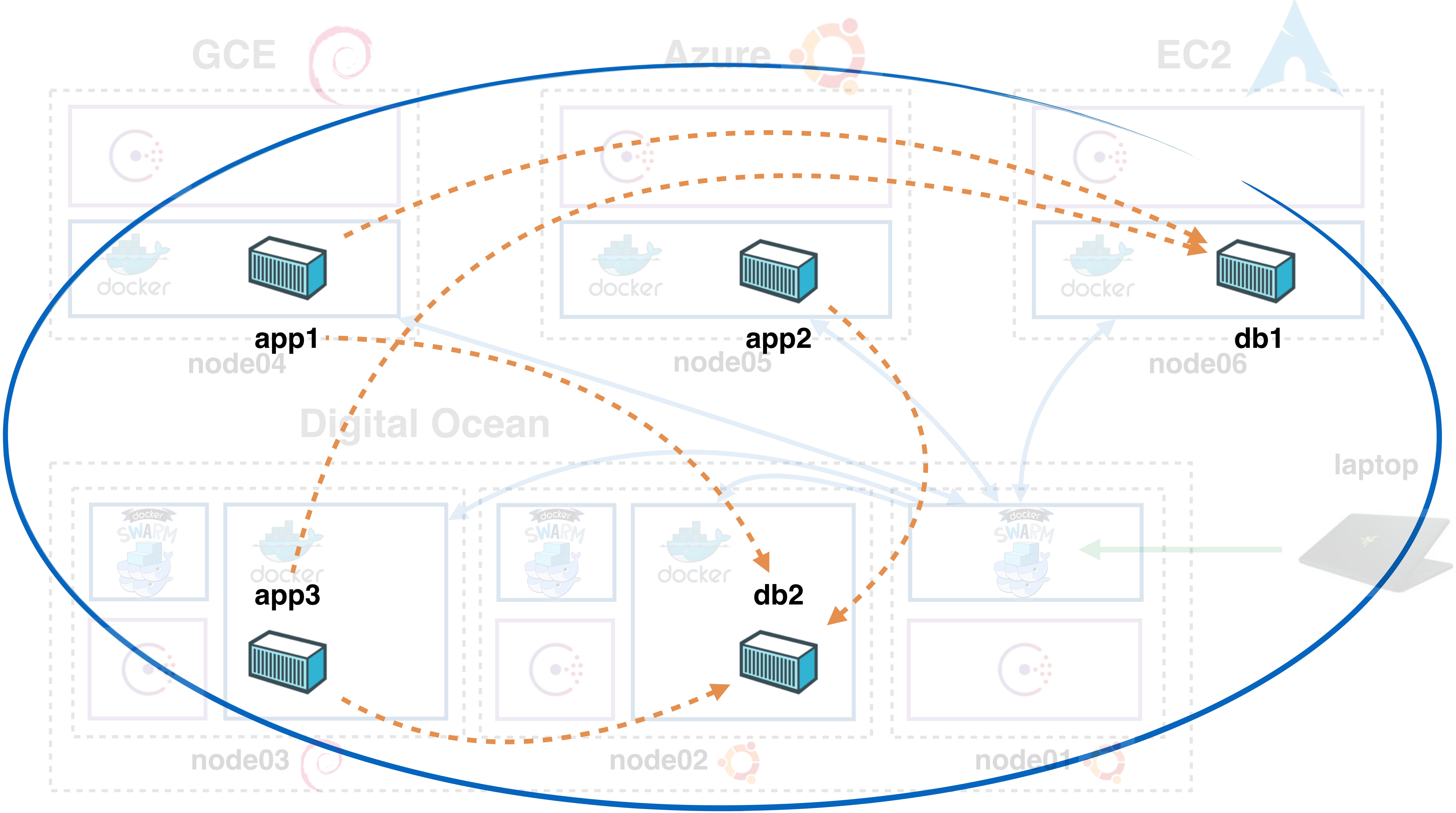
node03

node02

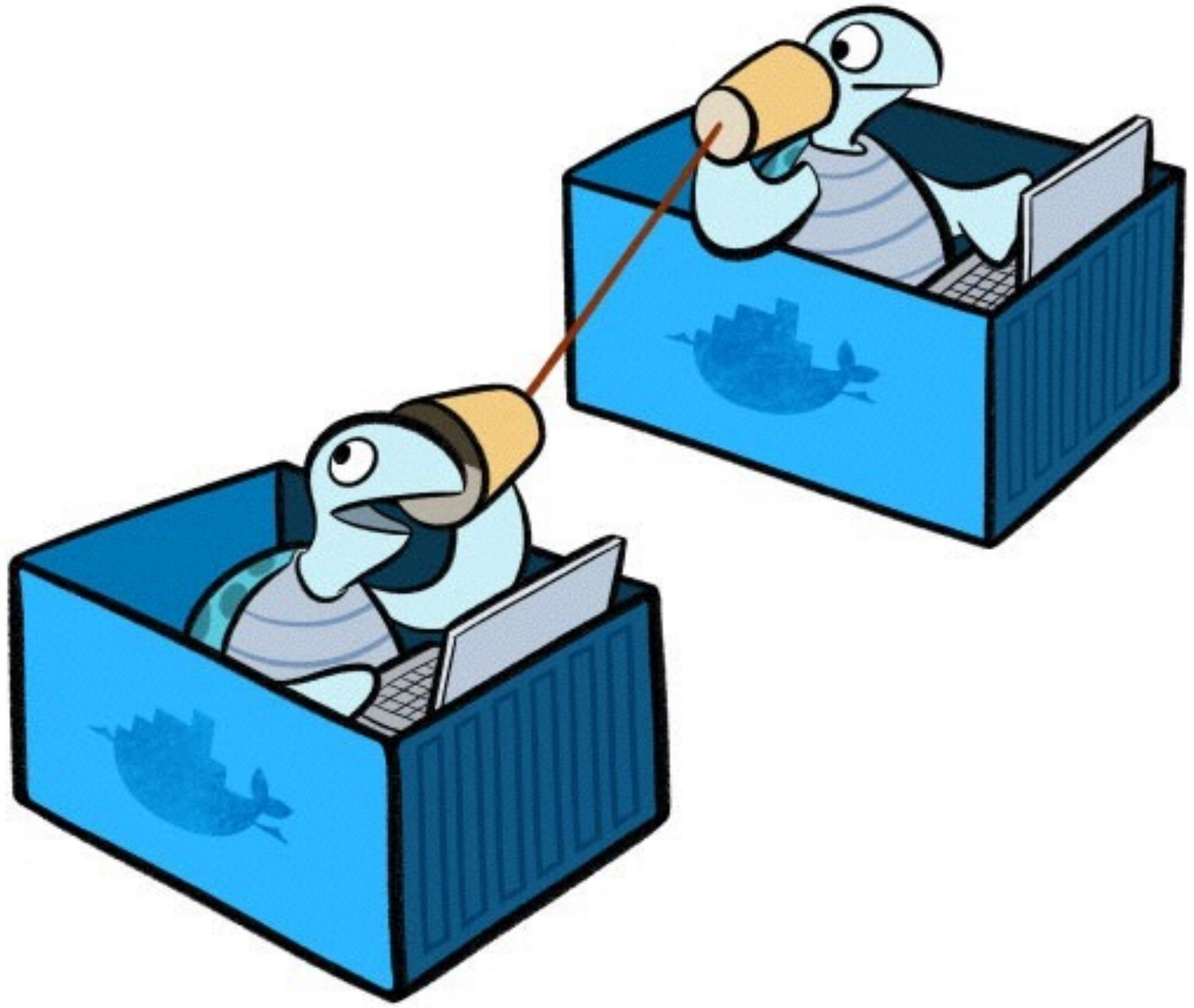
node01



GCE
Azure
EC2
docker
node04
node05
node06
Digital Ocean
overlay-1
laptop
docker SWARM
node03
node02
node01


GCE
Azure
EC2
docker
app1
node04
docker
app2
node05
docker
db1
node06
Digital Ocean
laptop
docker SWARM
docker
app3
docker SWARM
docker
db2
docker SWARM
node03
node02
node01

Demo time!



Volume management

```
$ docker volume --help

Usage:  docker volume [OPTIONS] [COMMAND]

Manage Docker volumes

Commands:
  create               Create a volume
  inspect              Return low-level information on a volume
  ls                   List volumes
  rm                   Remove a volume
```

Volume management - example

Create a volume

```
$ docker volume create --name=data
data
```

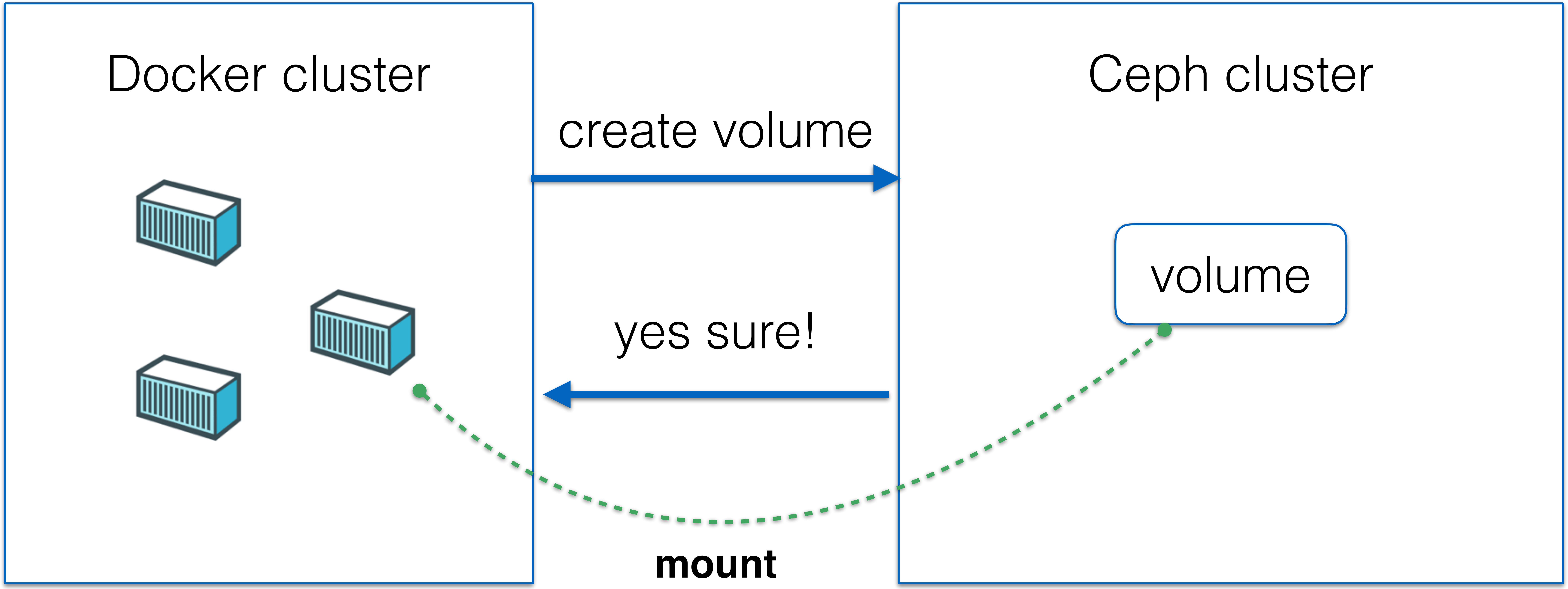
Run a container with the volume

```
$ docker run -ti -v data:/data alpine /bin/sh -c “echo hello > /data/world”
```

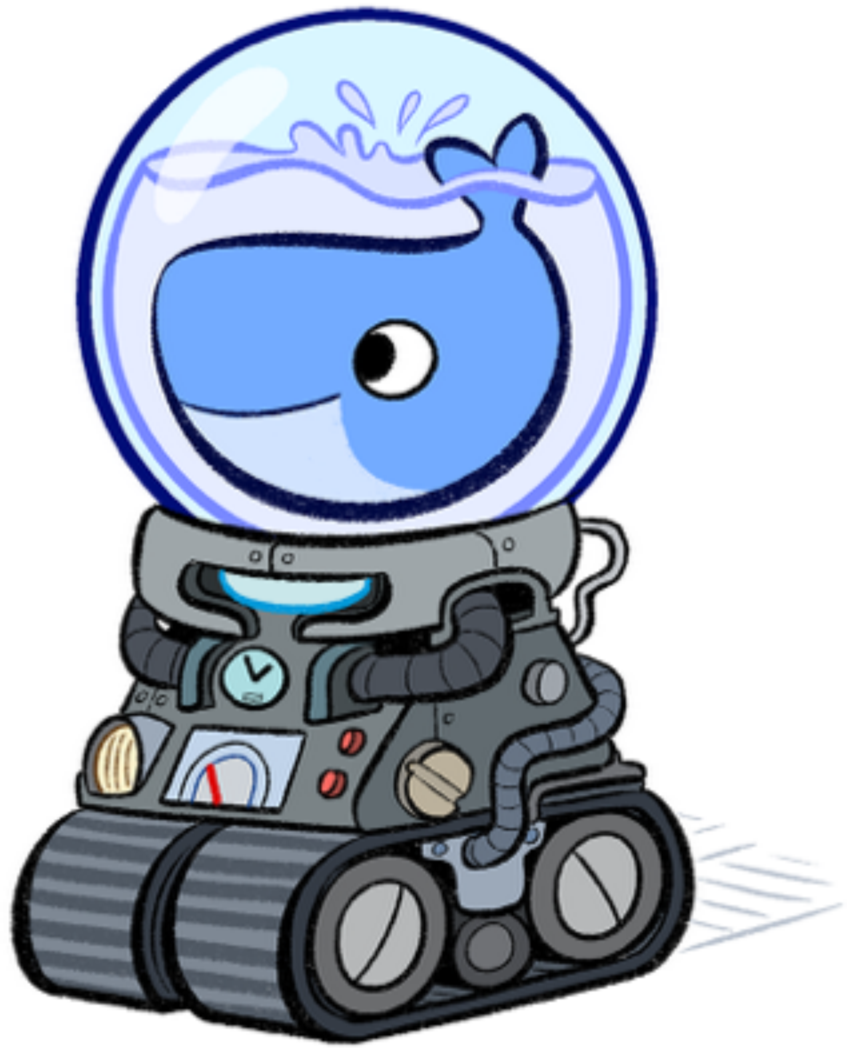
Running another container with that volume

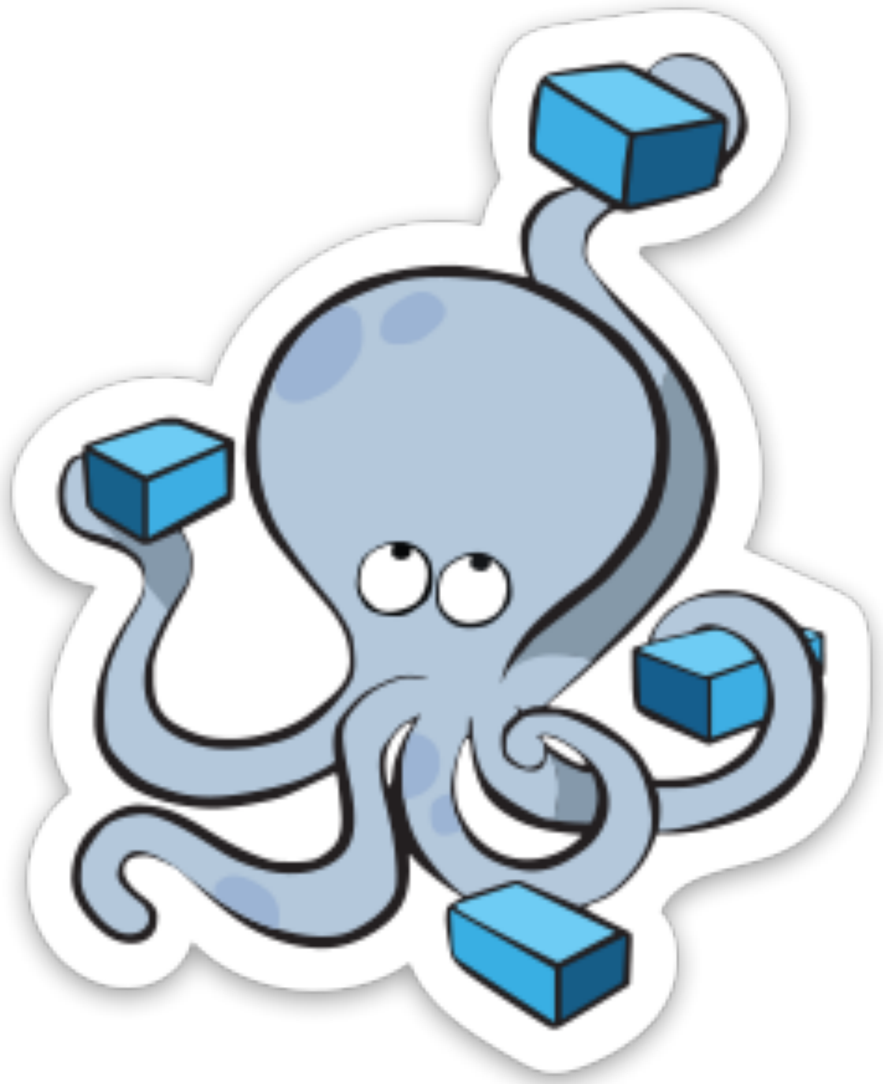
```
$ docker run -ti -v data:/data alpine cat /data/world
hello
```

Volume Plugins

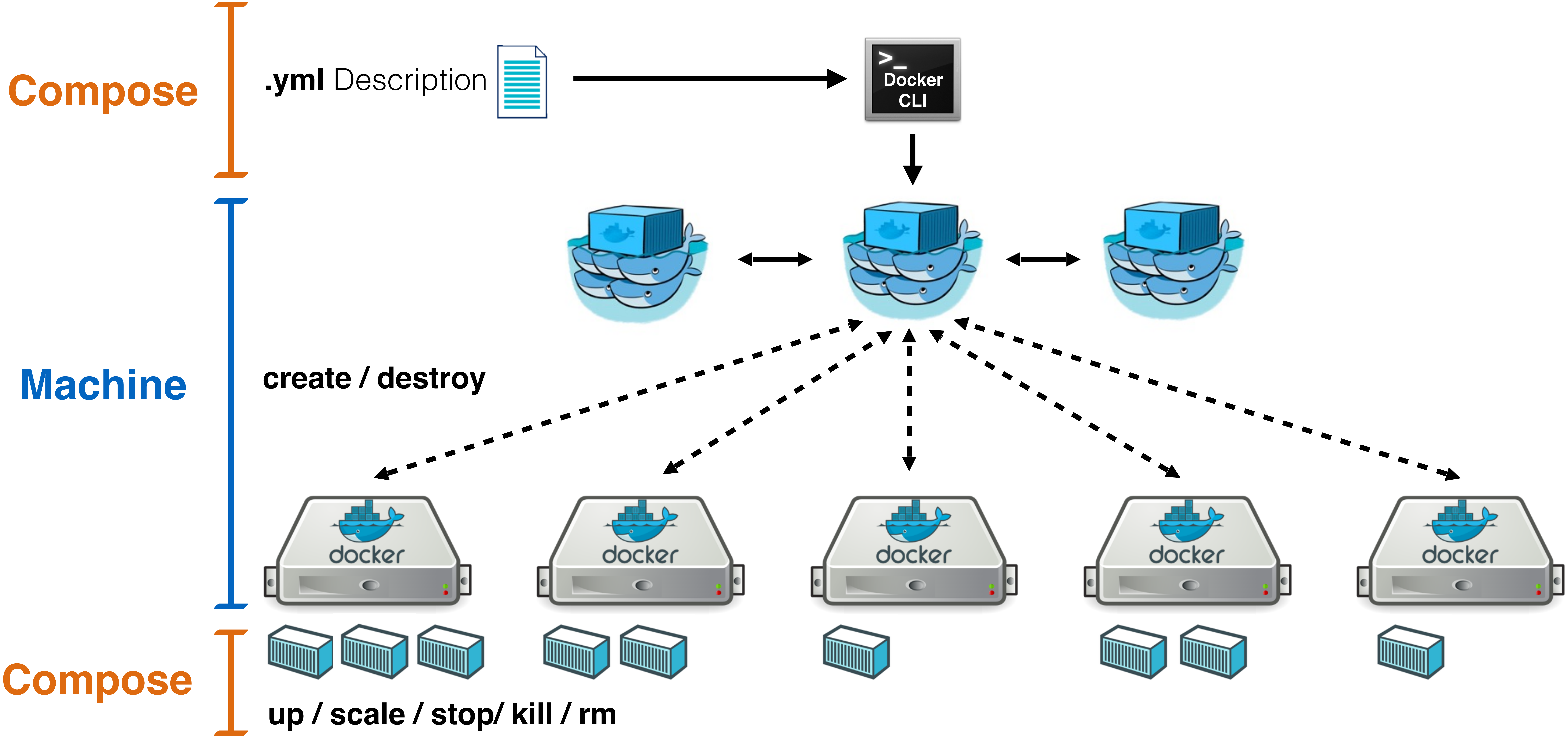


Better integration with Machine and Compose





Swarm + Machine + Compose



Compose

.yml Description

Machine

create / destroy

Compose

up / scale / stop/ kill / rm

Thank You. Questions?

http://github.com/docker/swarm

@abronan