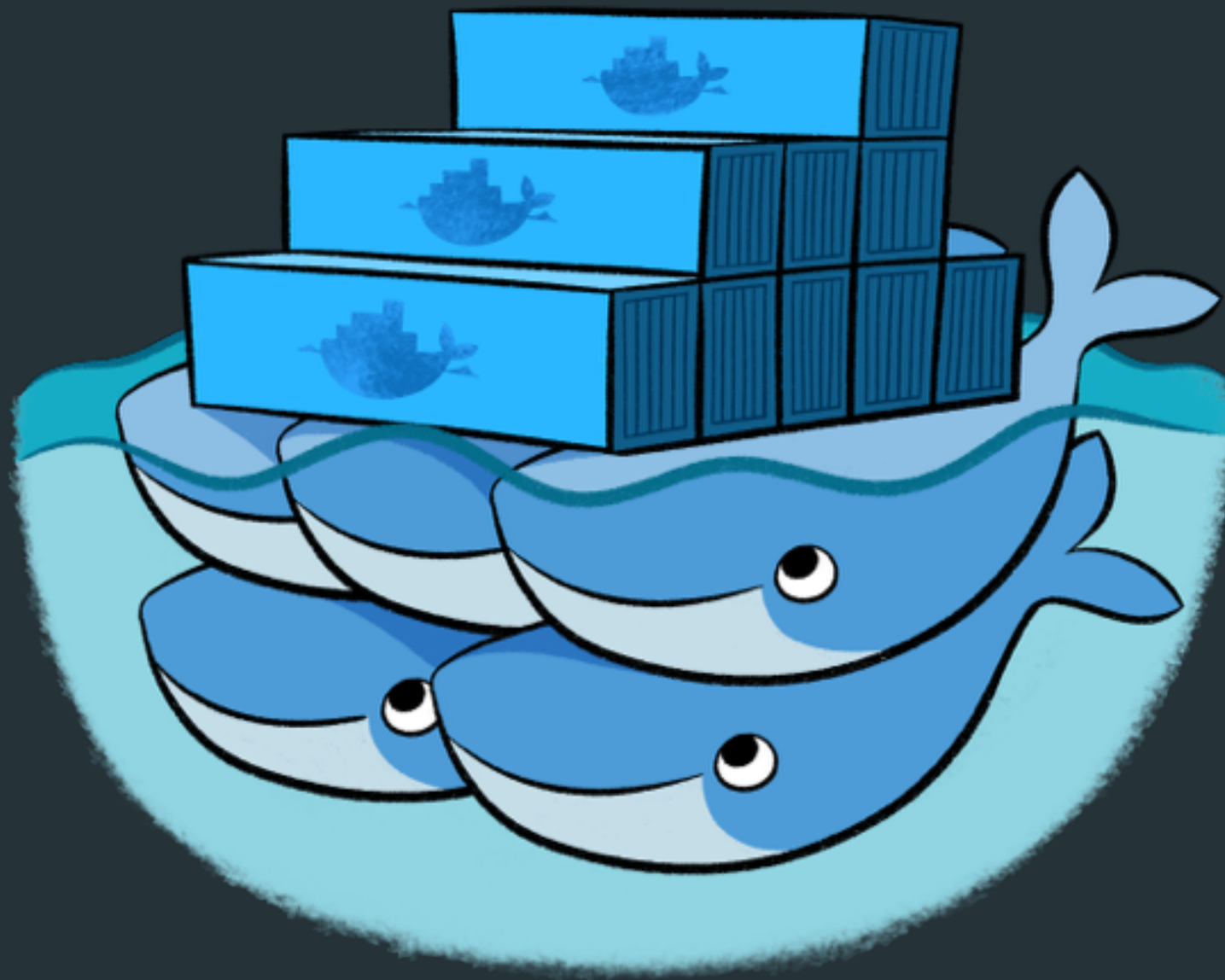


Introduction to Swarm

a Docker-native clustering system



@aluzzardi - @vieux - @abronan

Alexandre Beslic
@abronan



Introduction

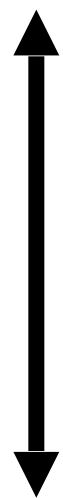
Demo

Upcoming features

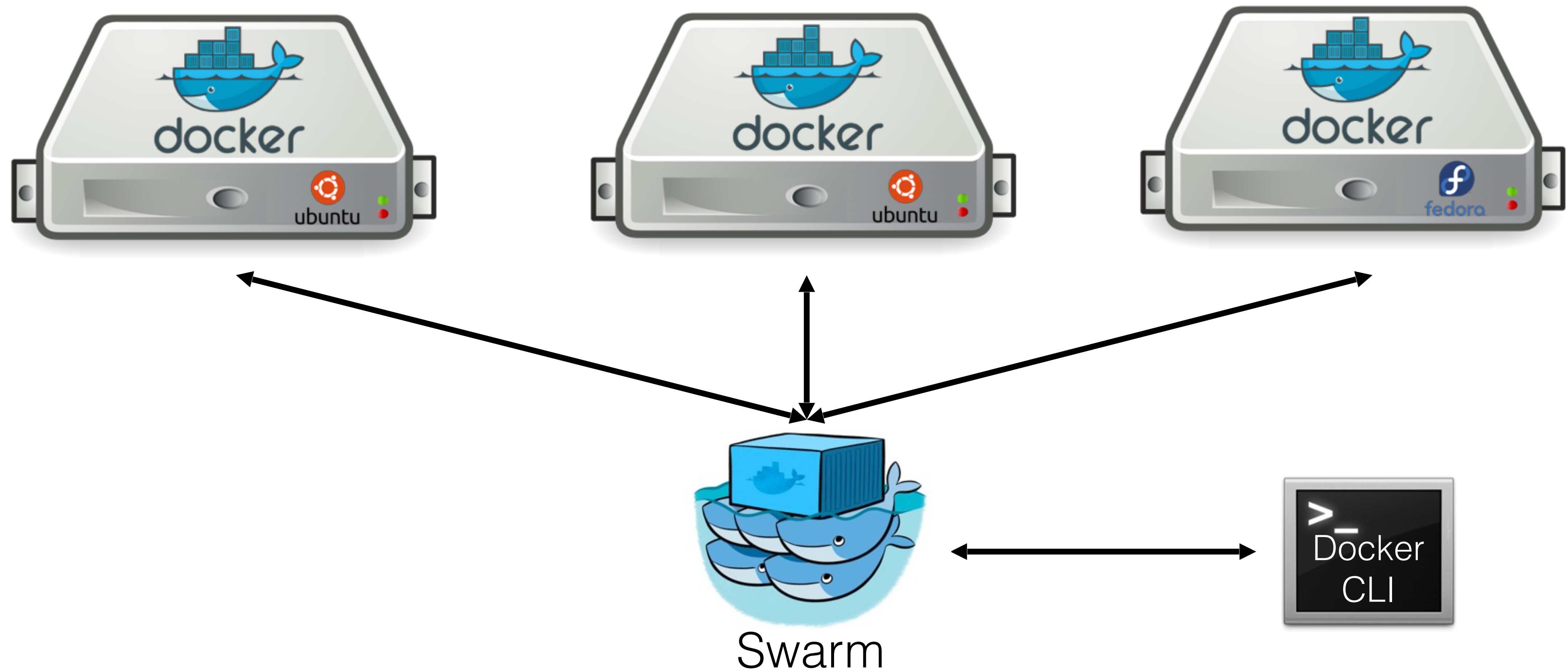
Q&A



Today



With Docker Swarm

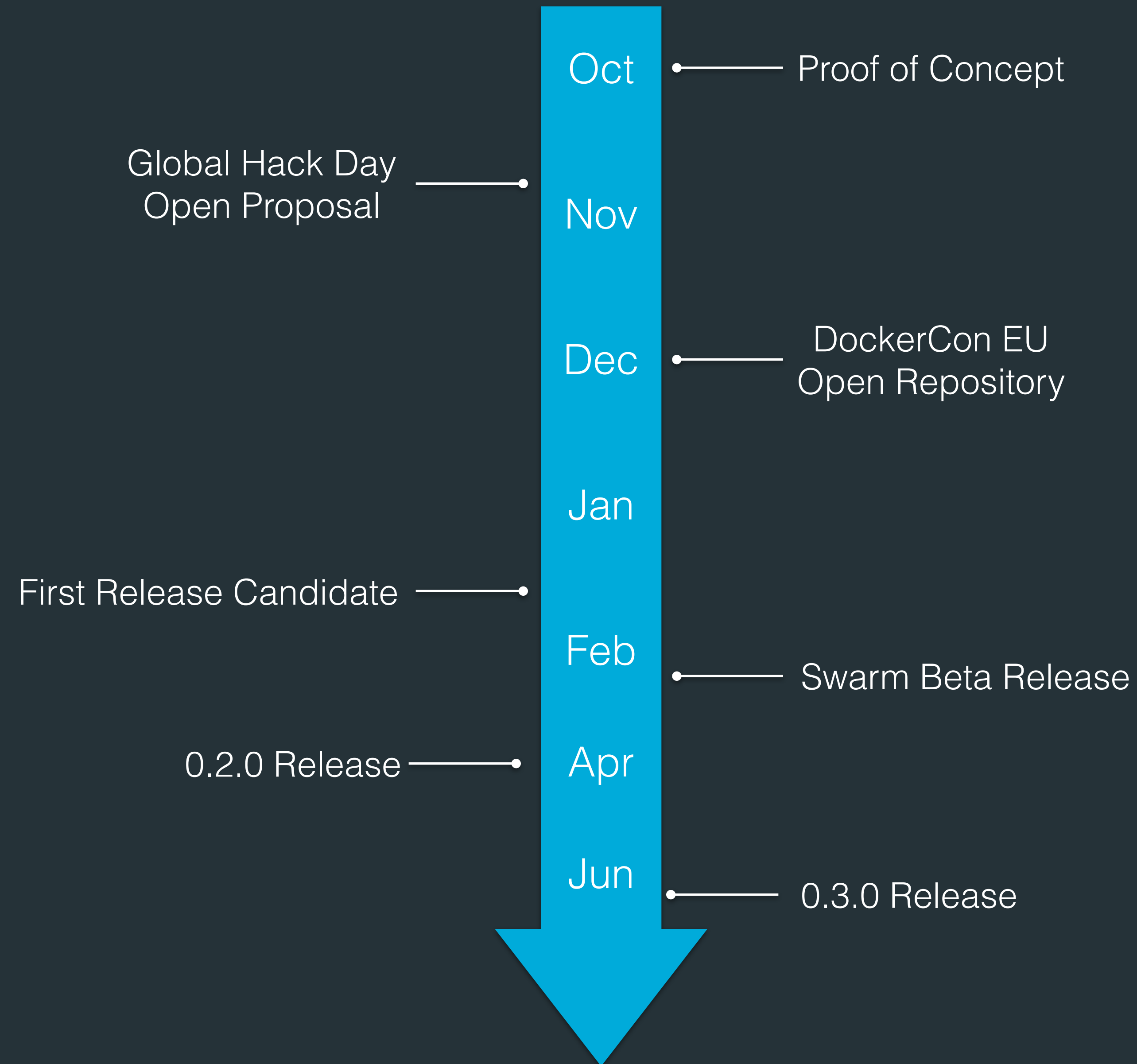


Swarm in a nutshell

- Exposes several Docker Engines as a single virtual Engine
- Serves the standard Docker API
- Extremely easy to get started
- Batteries included but swappable



Timeline



Swarm 0.2.0

- **Docker REST API** (>85%)
- **Resource management** (CPU, Mem, Networking)
- **Advanced scheduling** with constraints and affinities
- Multiple **Discovery Backends** (hub, etcd, consul, zookeeper)
- **TLS**: Encryption & Authentication



Setup using the hosted discovery service

- Create a cluster:
\$ swarm create
- Add nodes to a cluster:
\$ swarm join --add=<node_ip> token://<token>
- Start Swarm
\$ swarm manage --addr=<swarm_ip> token://<token>

Or you can use your own etcd, zookeeper or consul

Contributions are welcome ☺



Swarm Scheduler

2 steps:

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

Contributions are welcome ☺



Resource Management

- Memory
\$ **docker run -m 1g ...**
- CPU
\$ **docker run -c 1 ...**
- Ports
\$ **docker run -p 80:80 ...**
- More to come, ex: network interfaces



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 -d --label "region==us-east"
docker run -e "constraint:region==us-east" ...
- Pin a container to a specific host
docker run -e "constraint:node==ubuntu-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



Soft Affinities

- Containers soft affinities
`docker run -d --name redis5 -e affinity:container!=~redis* redis`



Swarm Beta: Integrations

- Fully integrated with **Machine**
`$ machine create -d azure --swarm --swarm-discovery token://<token> ...`
- Partially integrated with **Compose**
`$ DOCKER_HOST=<swarm_addr> compose up`
- **Mesos** & **DCOS** integration has started in collaboration with **Mesosphere**.
`$ swarm manage -c mesos zk://<zookeeper_addr>/swarm`

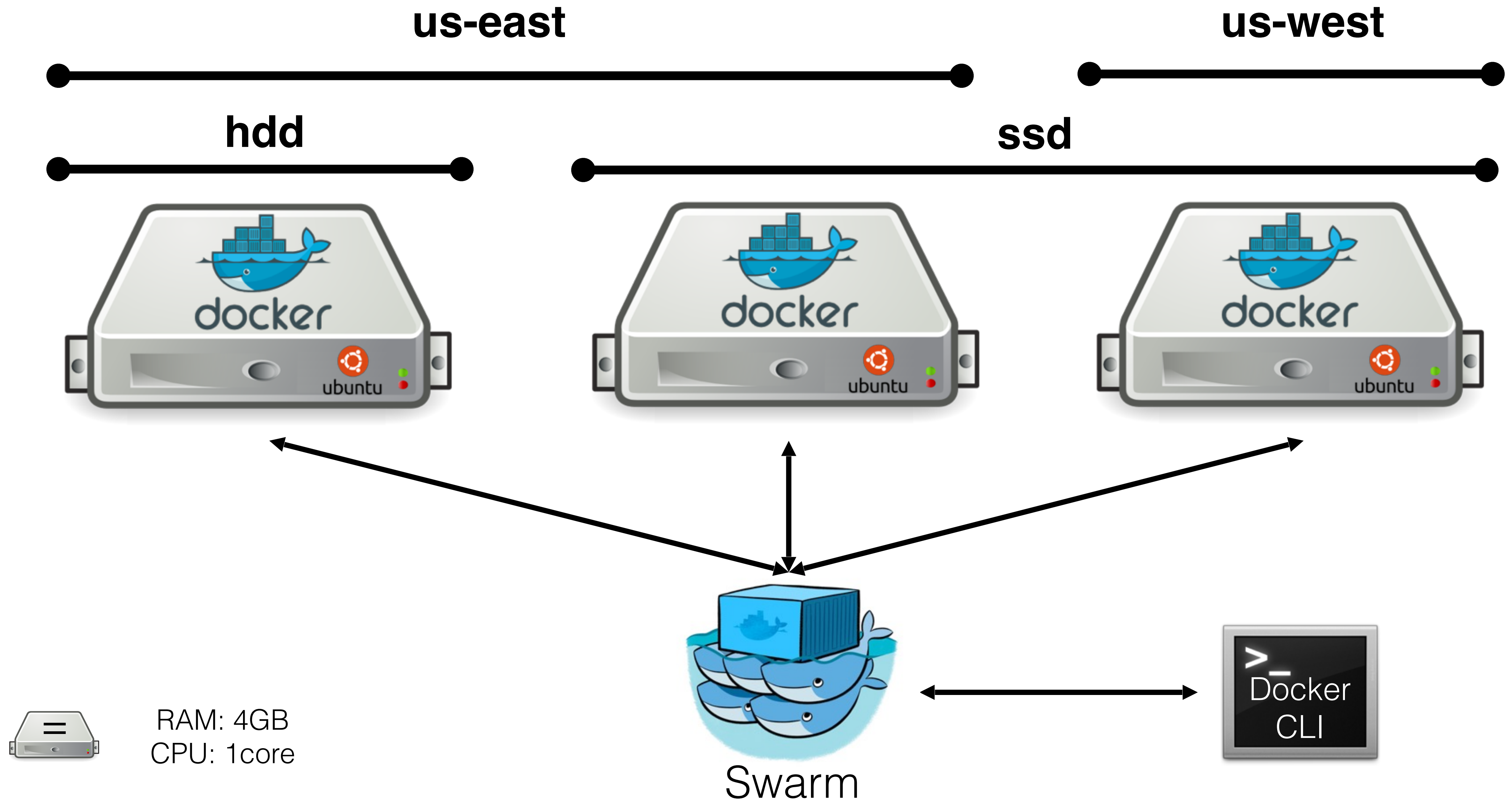


Demo

Scheduling containers on Swarm



Demo



Swarm Beta: Upcoming features

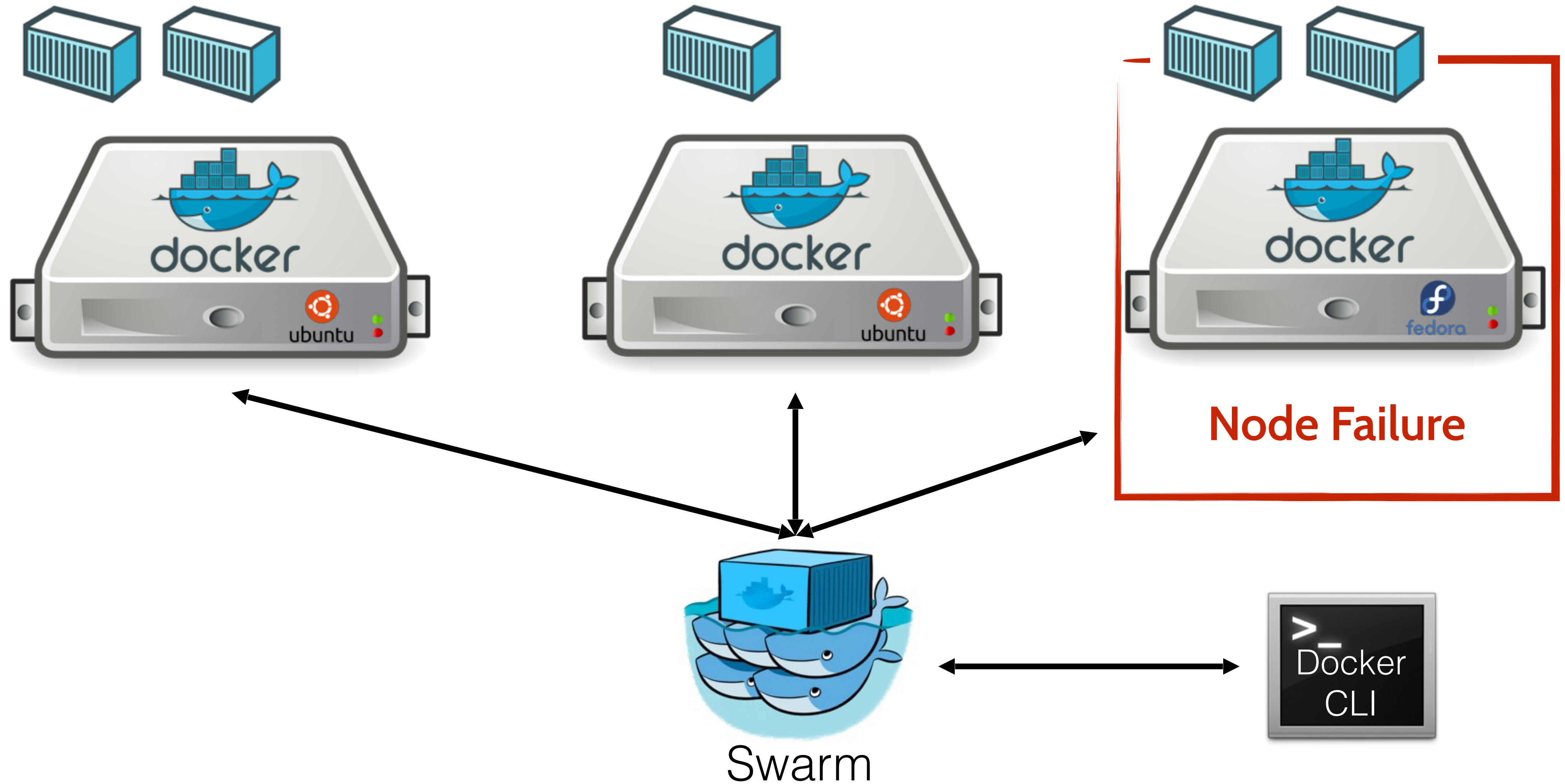
- Support for K/V backends (consul, etcd, zookeeper)
- High Availability with replicated state between multiple Managers
- Attempt at Re-scheduling on Node failure
- Networking (libnetwork integration into docker)



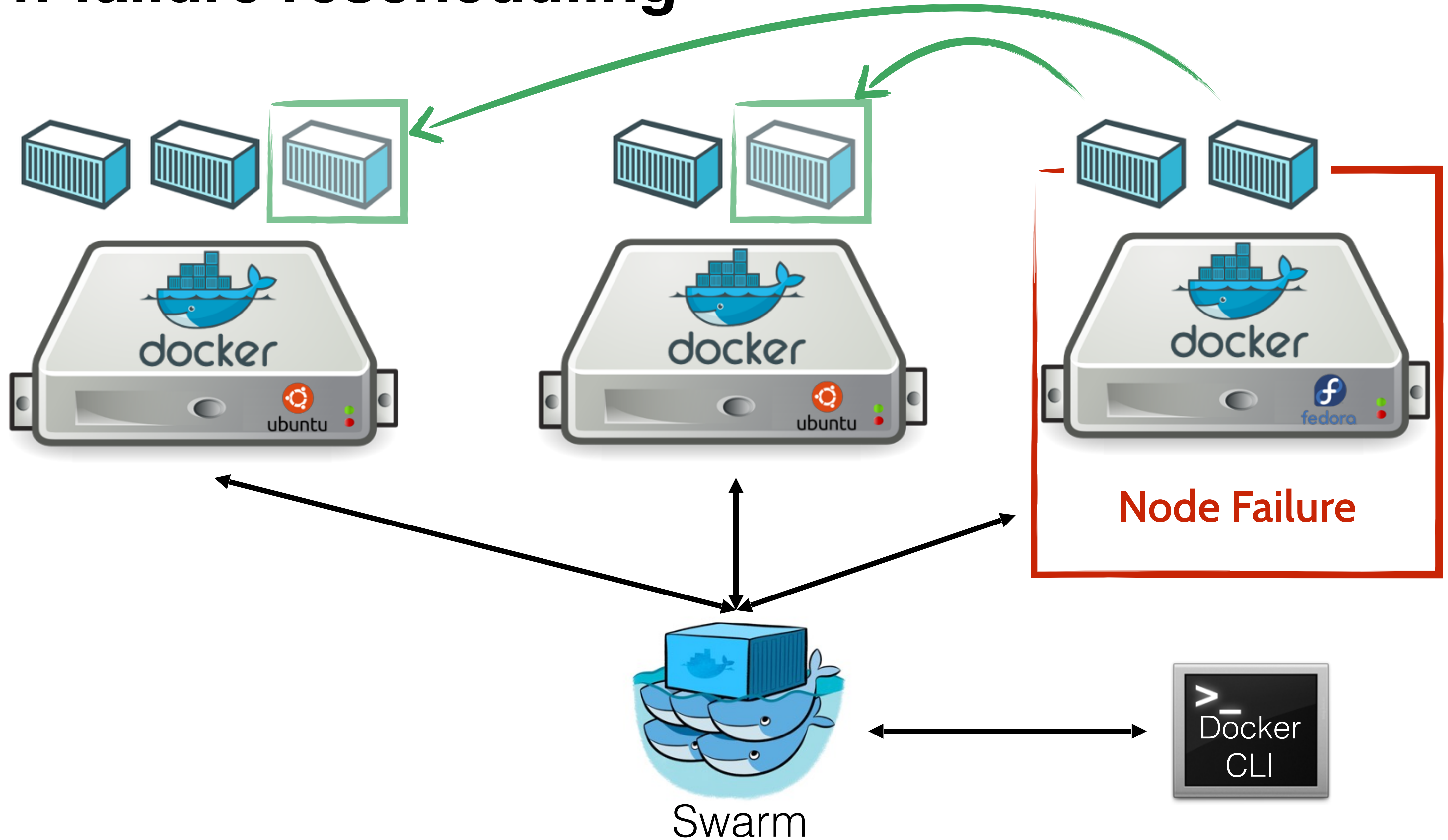
On-failure rescheduling



On-failure rescheduling

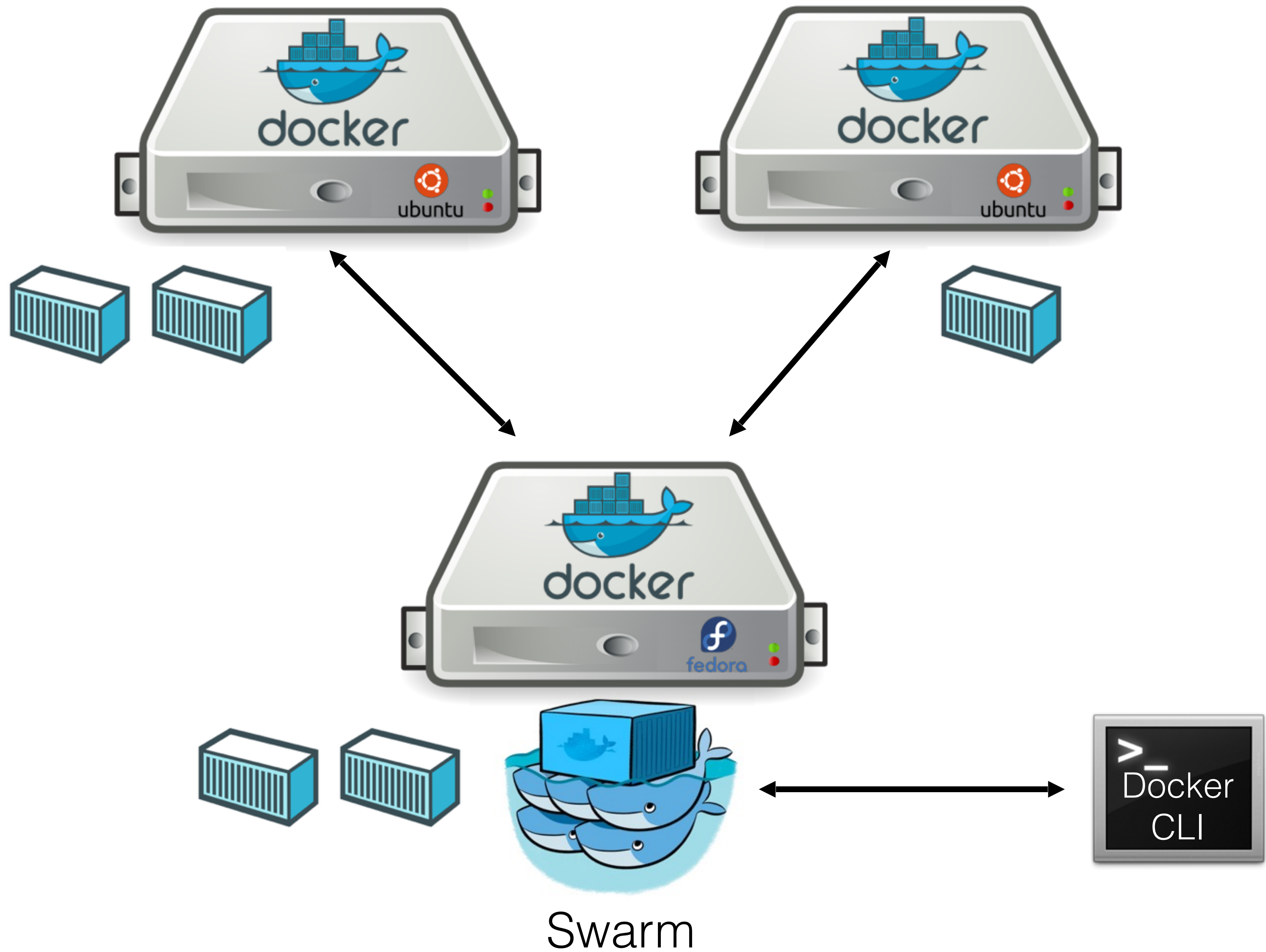


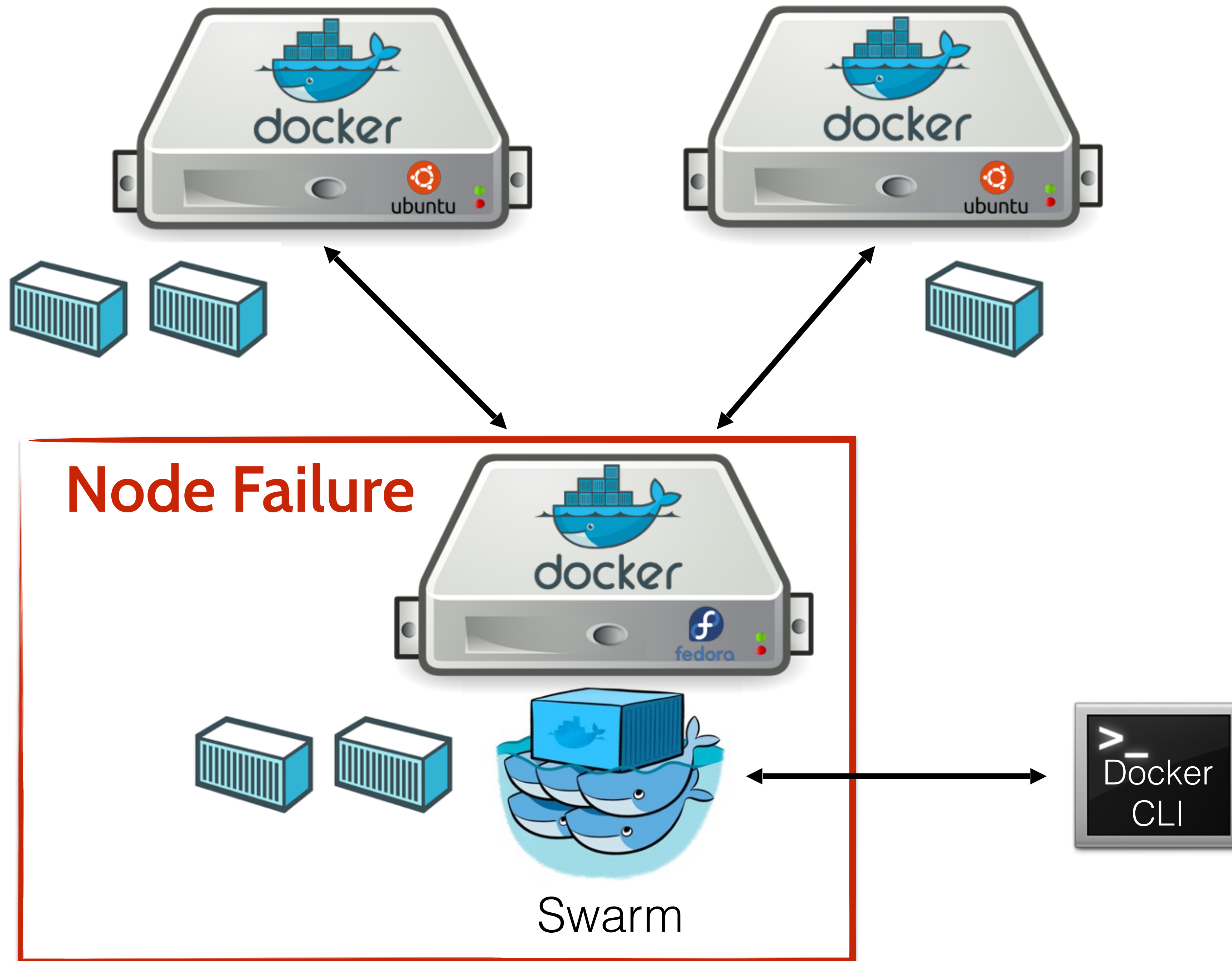
On-failure rescheduling

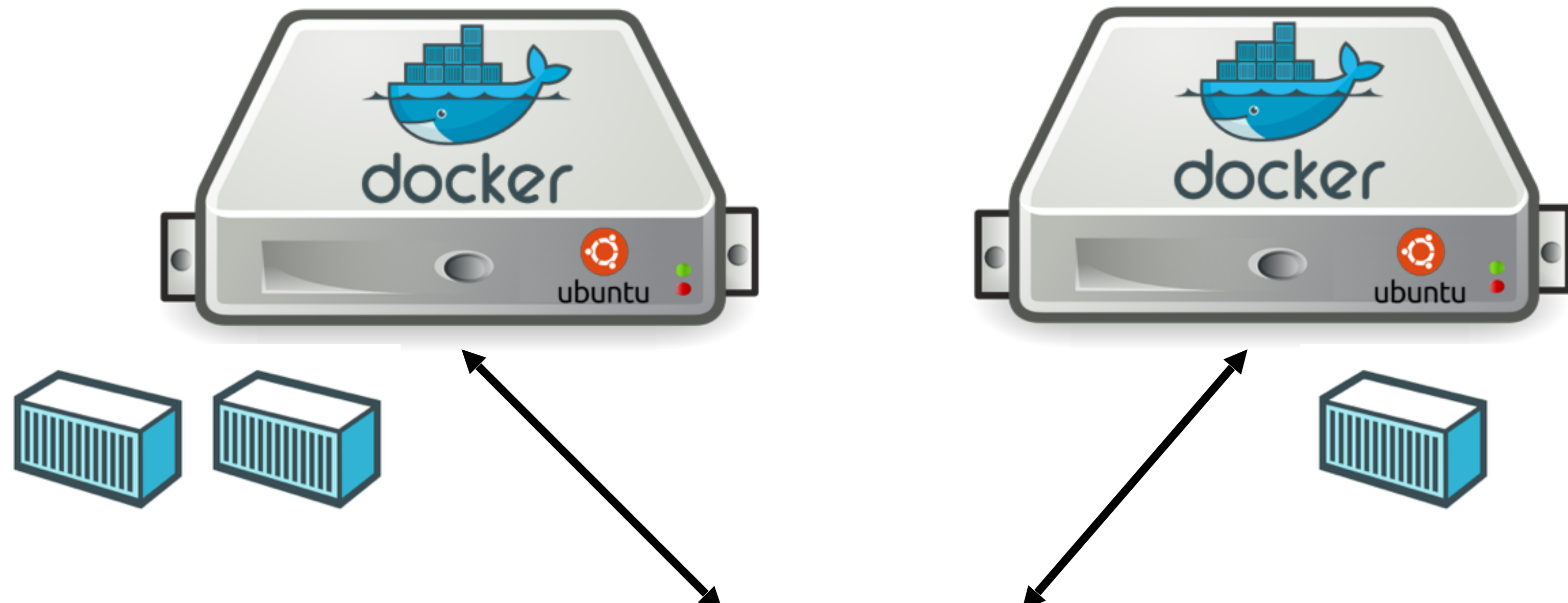


High Availability, State replication





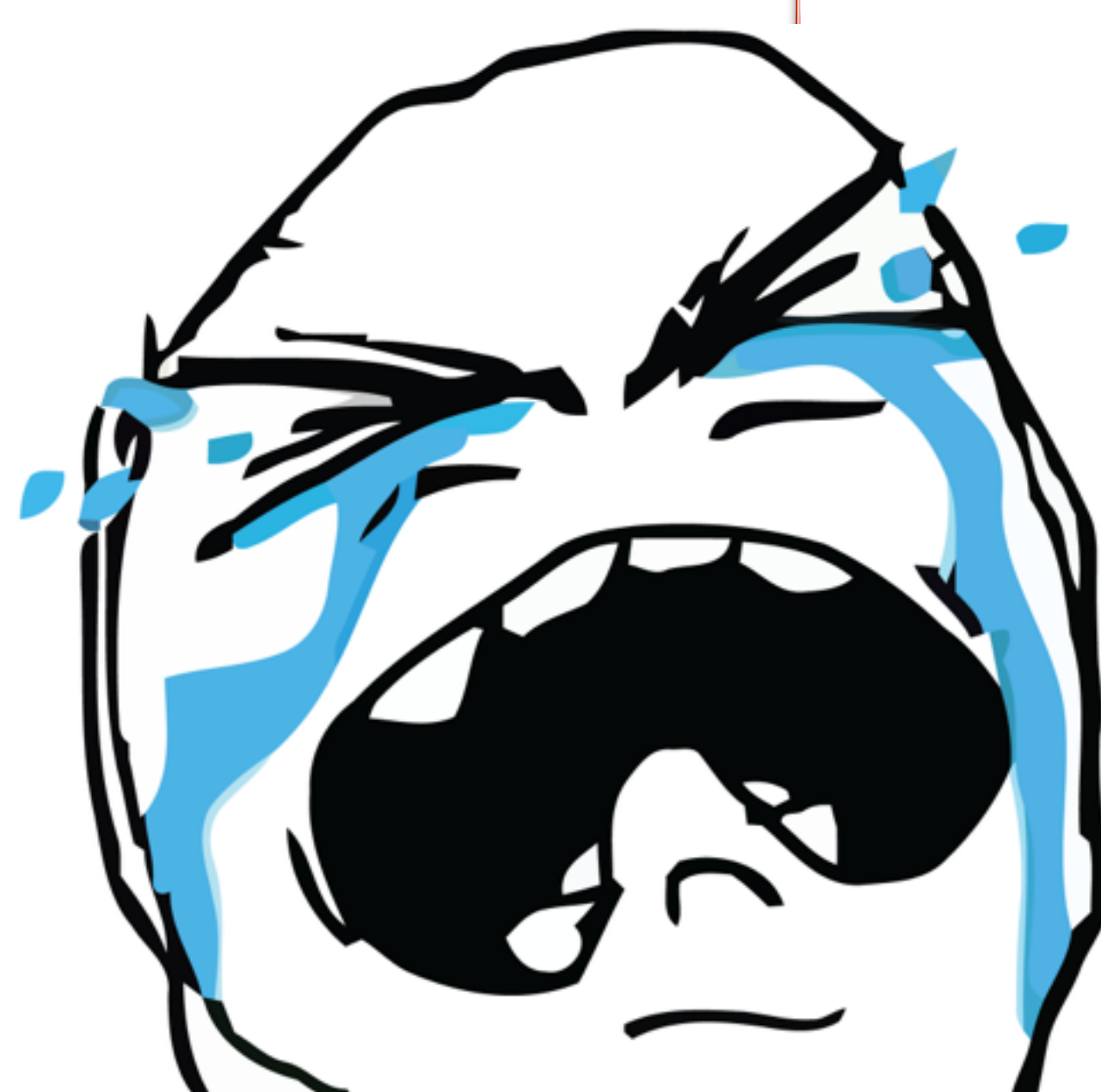




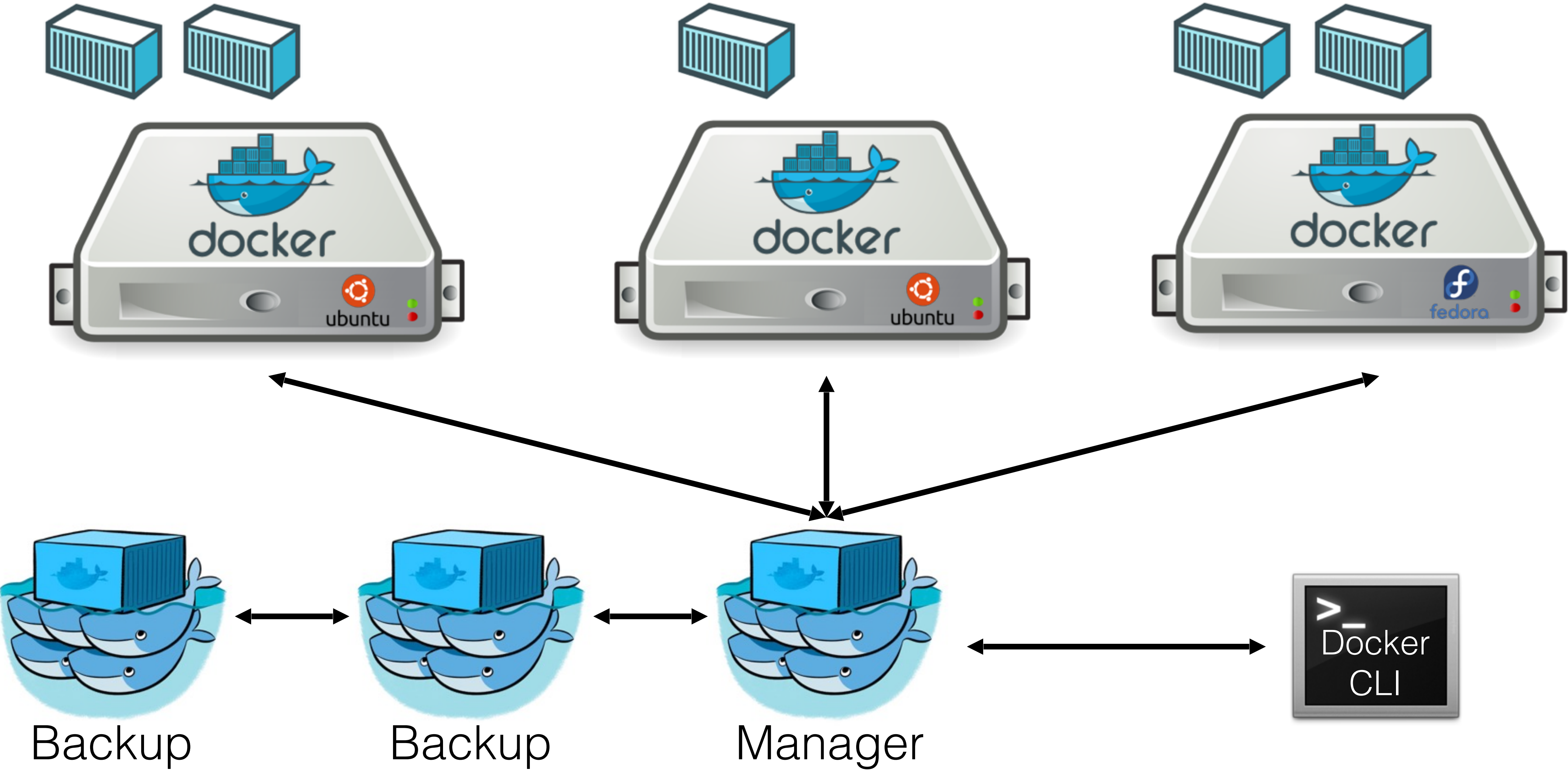
Node Failure



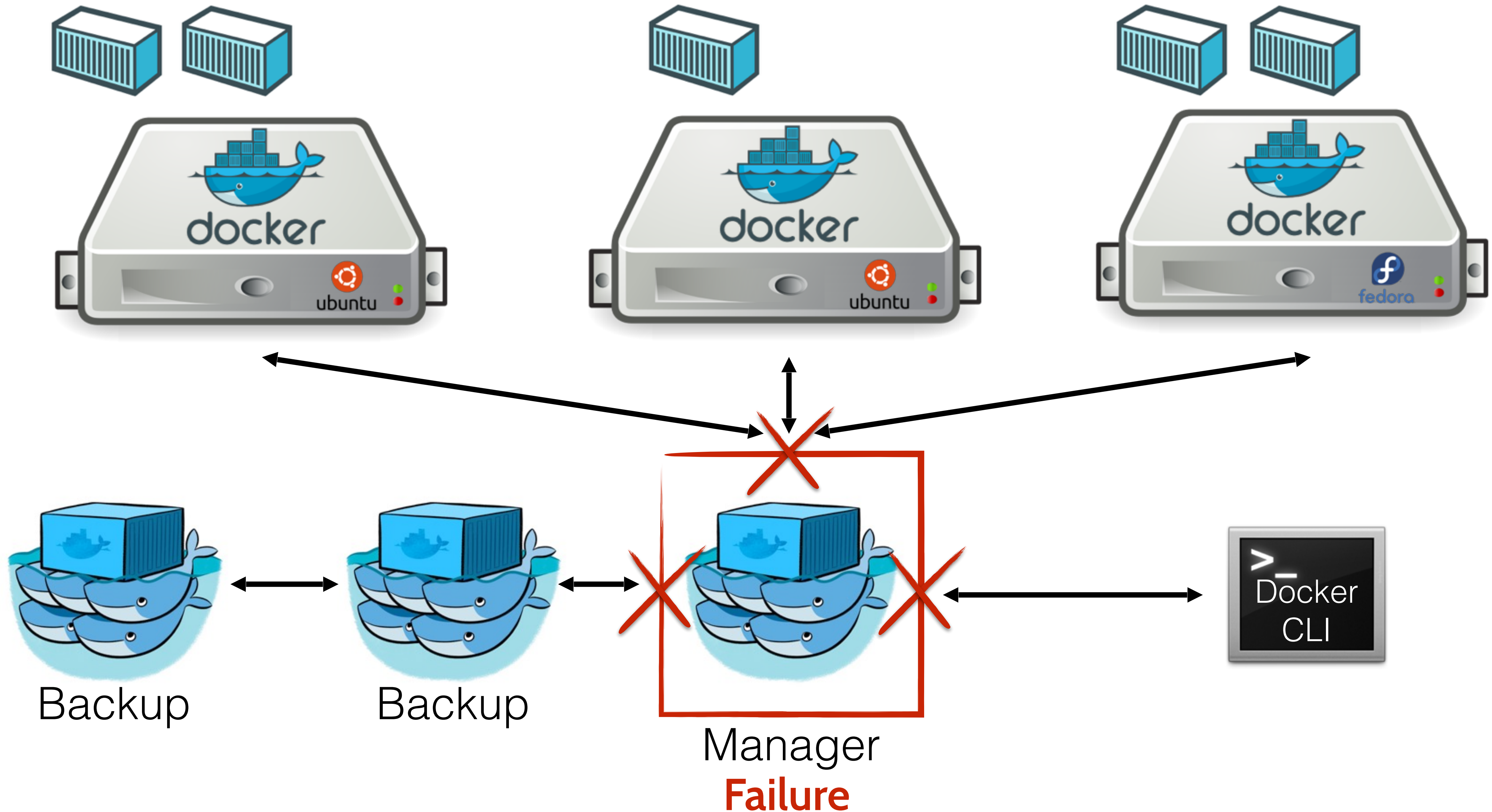
Swarm



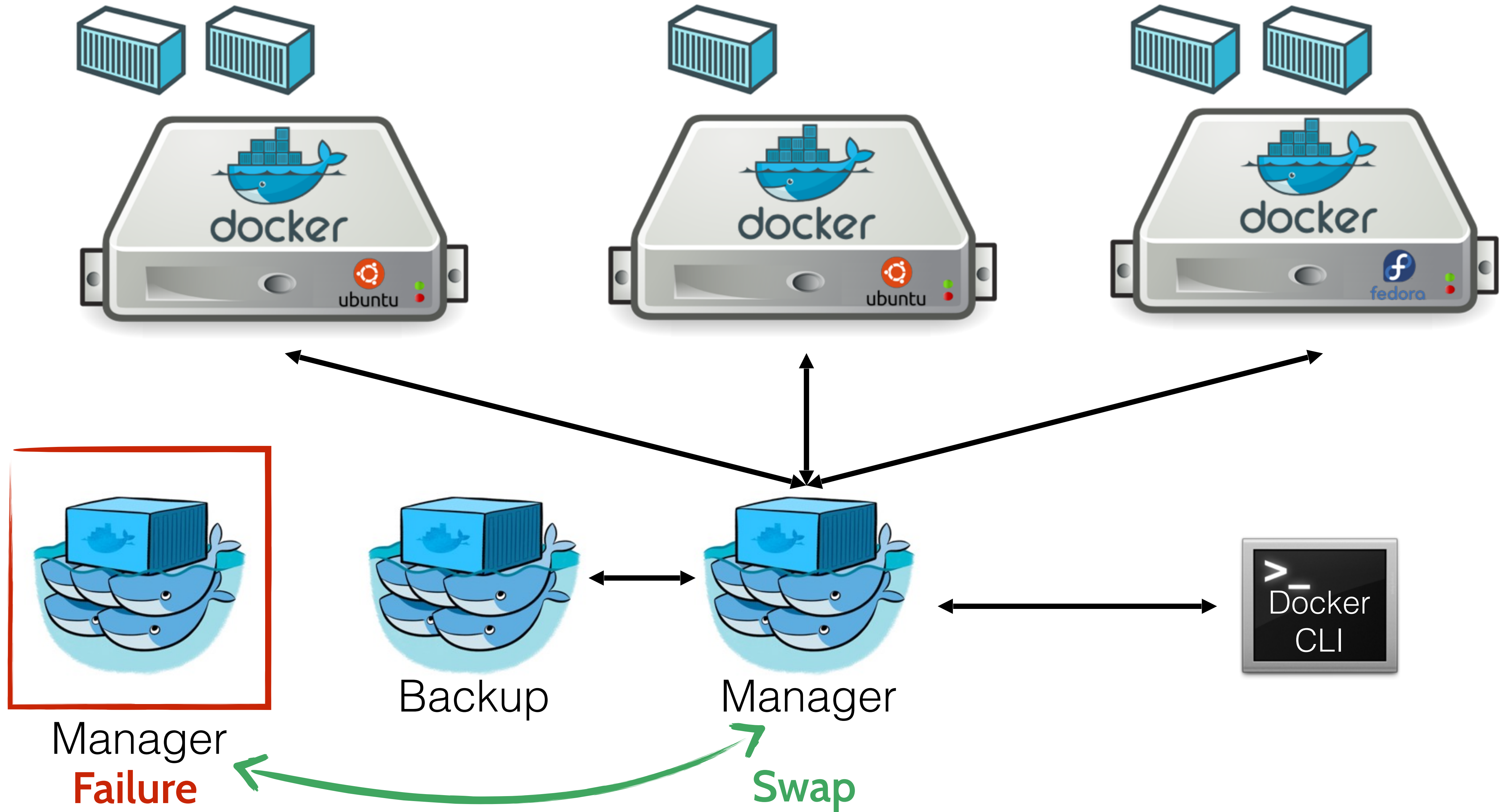
Manager backup instances



Manager backup instances



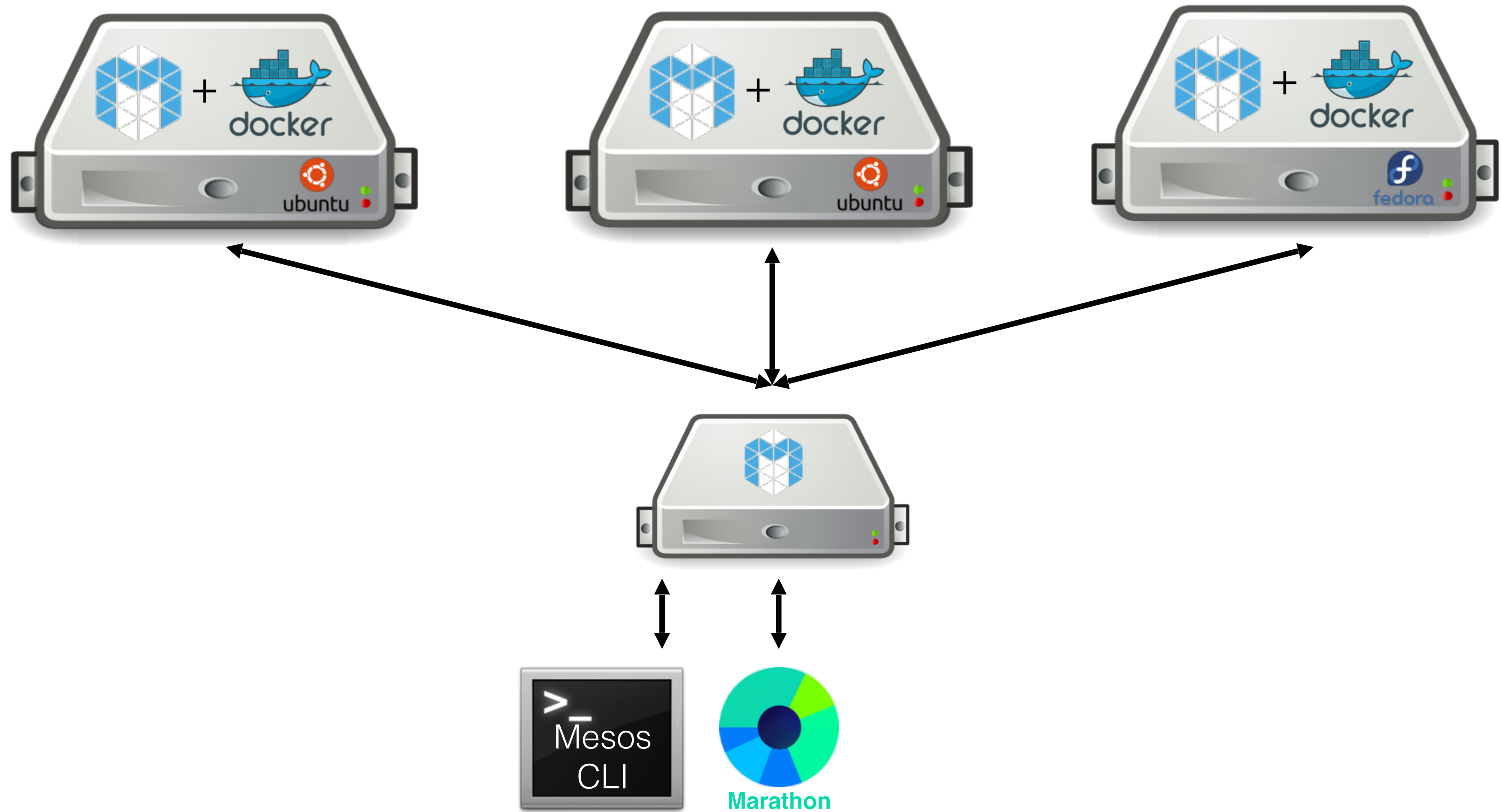
Manager backup instances



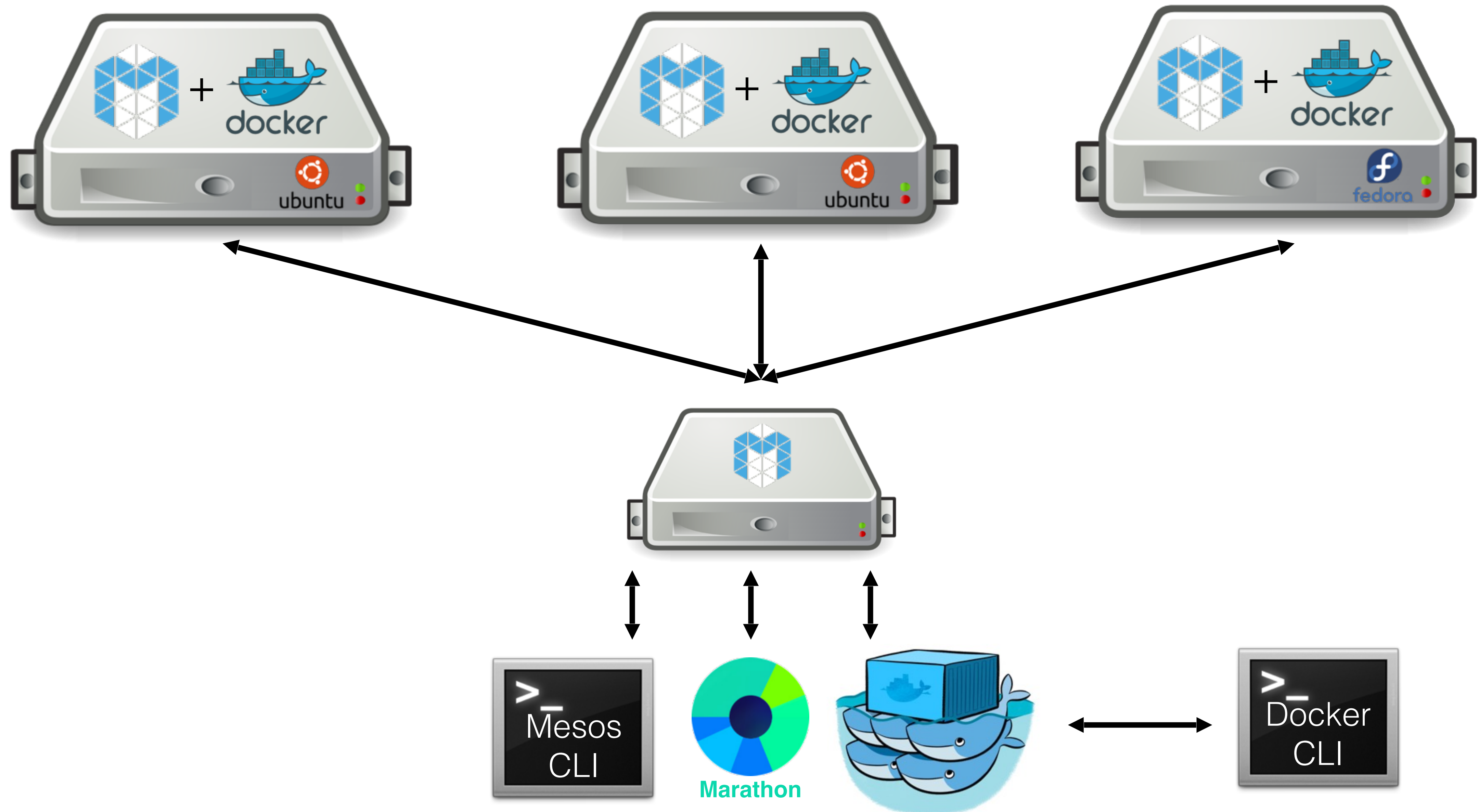
Mesos Integration



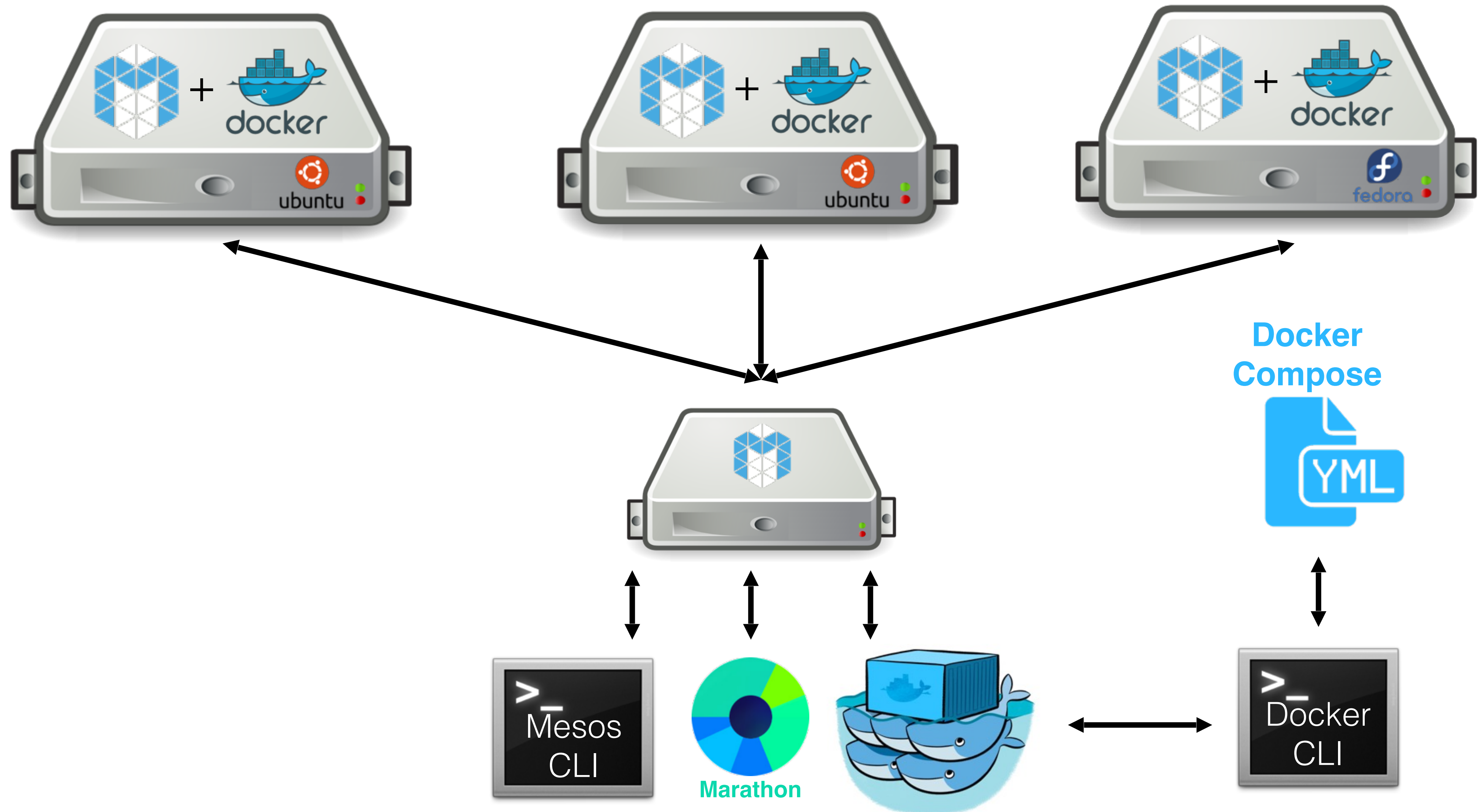
Mesos cluster



Mesos cluster + Docker Swarm



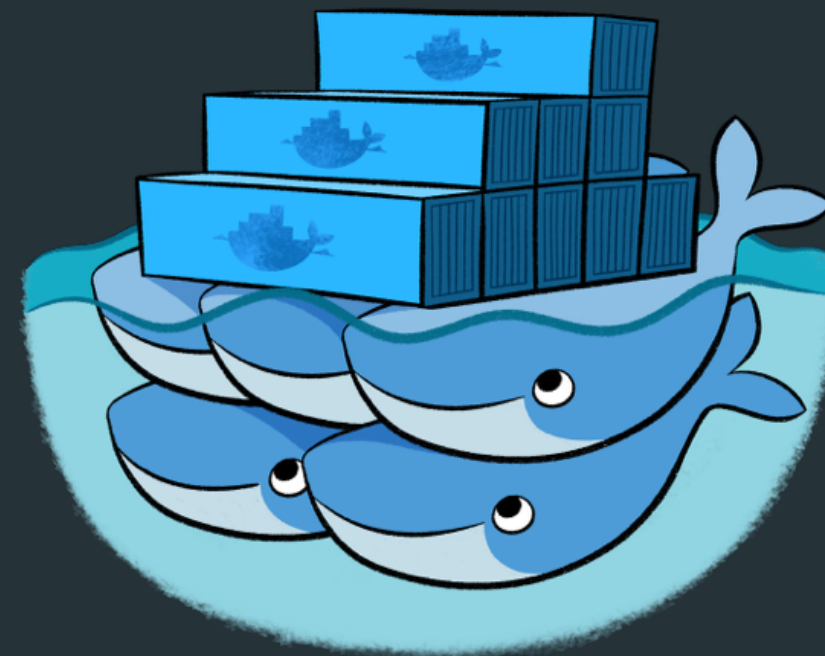
Mesos cluster + Docker Swarm



Thank You. Questions?

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

#docker-swarm on freenode



@aluzzardi - @vieux - @abronan