Security Level:

Docker卷存储和Elara介绍

华为中央软件研究院

www.huawei.com

Author/ Email: 张文涛 /zhangwentao234@huawei.com
Version: V1.0(2016-3-11)

HUAWEI

我的经历

2009年-2014年 北京爱立信通信技术有限公司

- 负责linux 平台底层软件开发工作。

2014年底-今 华为杭州研究所

- 负责容器OS、docker开发工作,提交过一些patch
- 负责docker卷插件的开发工作（Elara）

《Docker进阶与实战》

机械工业出版社出版

最全面
的内容

进阶与实战
的最佳选择

华为Docker实践小组 出品

Docker社区贡献全球排名前三、国内排名第一

【团队介绍】

我们致力于打造未来ICT领域的操作系统、云平台，我们致力于研究未来ICT、NFV/SDN、云服务场景下的OS、容器、虚拟化前沿技术。

这里有Linux kernel、Docker社区多名maintainer，有ACPI标准提案committer，有精通Linux kernel各子系统（调度、内存管理、文件系统、网络等）的资深专家，有畅销书《Docker进阶与实战》的作者团队和您一起探讨容器虚拟化的未来技术。

如果你相信技术可以改变生活、改变世界；如果你喜欢Linux、喜欢钻研技术；如果你热爱开源、想成为技术大咖；别犹豫，加入我们！

Docker/容器虚拟化架构师/高级工程师（北京/杭州）

【职责】

- 负责x86、arm 64架构下容器虚拟化需求分析、关键技术和特性的设计；
- 负责Docker/容器关键技术研究和特性开发；
- 负责Docker及相关开源社区互动、运作，如推送bug fix和新特性。

【要求】

- 计算机相关专业本科及以上学历，4年以上Linux系统或内核开发经验；
- 熟悉Linux容器相关技术（namespace、cgroup）者优先；
- 熟悉Docker源码、有Docker社区开发经验者优先；
- 熟悉容器编排/调度（Kurbernetes，Mesos等）技术者优先；

【招贤纳士】【华为-Docker团队】

诚邀容器、虚拟化领域专家

【电 话】18501294585/北京、13732261657/杭州
【邮 箱】hr.kernel@huawei.com

本次讲解的内容

- Docker卷操作接口
  - Docker对数据卷的支持
  - Docker的卷插件机制
- 存储简介
- 容器生态中的存储现状
  - 目前容器社区存储的问题
  - 为什么设计Elara卷插件
- Elara卷插件
  - Elara的功能特性和规格
  - Elara应用场景
  - 存储资源调度
  - 主流卷插件对比
  - Elara操作实例

Docker卷操作接口

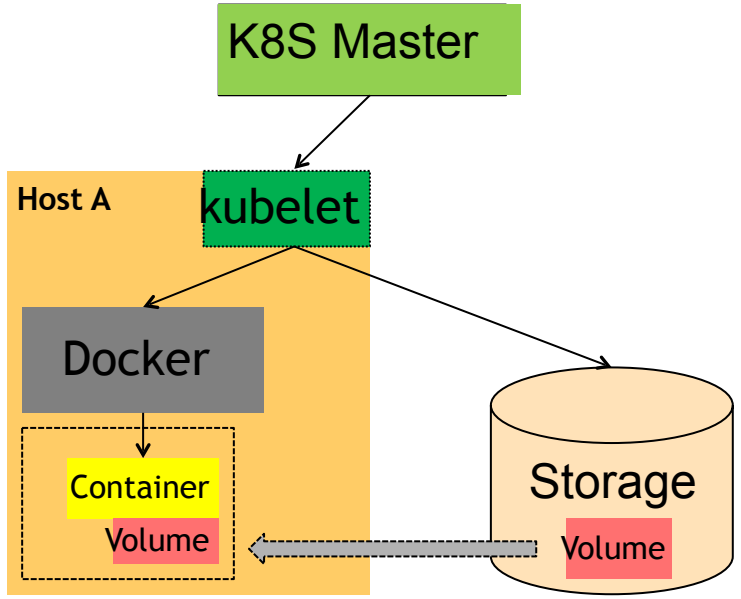
- Docker对数据卷的原始支持
  - 在docker低版本就支持通过 –v参数来挂在数据卷

```
docker run -v /xx/volume:/data  niginx:web_server
```

- K8S的存储

➢K8s的存储是在卷插件出现之前就可以使用了，所以k8s并没有对接docker的卷插件机制

➢K8s可以对接多种后端存储，相对比较成熟



Docker卷操作接口——卷插件总览

- Docker Volume Plugin的卷存储使用方式

  1. docker run -v new_volume:/test --volume-driver=flocker ubuntu bash
  2. docker volume create --name dataVol --driver=flocker -o size=10G

- 目前Docker社区比较推荐这种方式
  - Docker借助卷插件来管理数据卷
  - 接口统一（docker的接口）
  - 合入比较晚（docker1.9)
- 缺点：
  - 接口比较少，无Snapshot，backup，Qos（iops）等接口操作，还需要编排做额外的对接

https://github.com/docker/docker/blob/master/docs/extend/plugins_volume.md

Docker卷操作实例

```
root@steady-wentao-use:~/work/elara# docker volume -h

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
```

说明： docker volume的help，目前该命令只集成了 create/rm/ls/inspect 四个命令，无snapshot相关操作

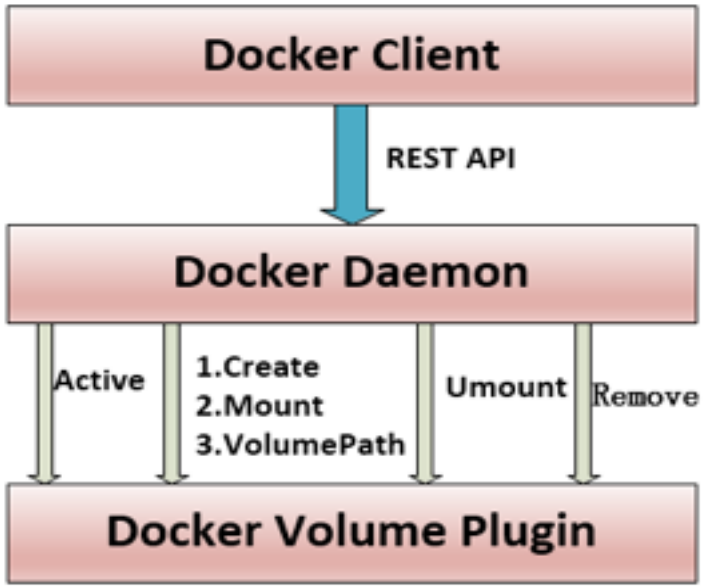
```
root@steady-wentao-use:~# docker volume create --name tester_3 --driver elara -o size=10G -o driver-name=nfs
tester_3
root@steady-wentao-use:~# docker volume ls
DRIVER              VOLUME NAME
elara               tester_3
elara               tester_7
elara               tester_9
root@steady-wentao-use:~# docker volume rm tester_3
tester_3
root@steady-wentao-use:~# docker volume inspect tester_9
[
    {
        "Name": "tester_9",
        "Driver": "elara",
        "Mountpoint": ""
    }
]
root@steady-wentao-use:~#
```

说明：docker volume命令创建/删除卷，可以看出，所有的卷都是通过Elara来创建的。

说明： 使用Docker volume inspect 一个卷，可以得到该volume的mountpoint，本例中该卷还未挂载

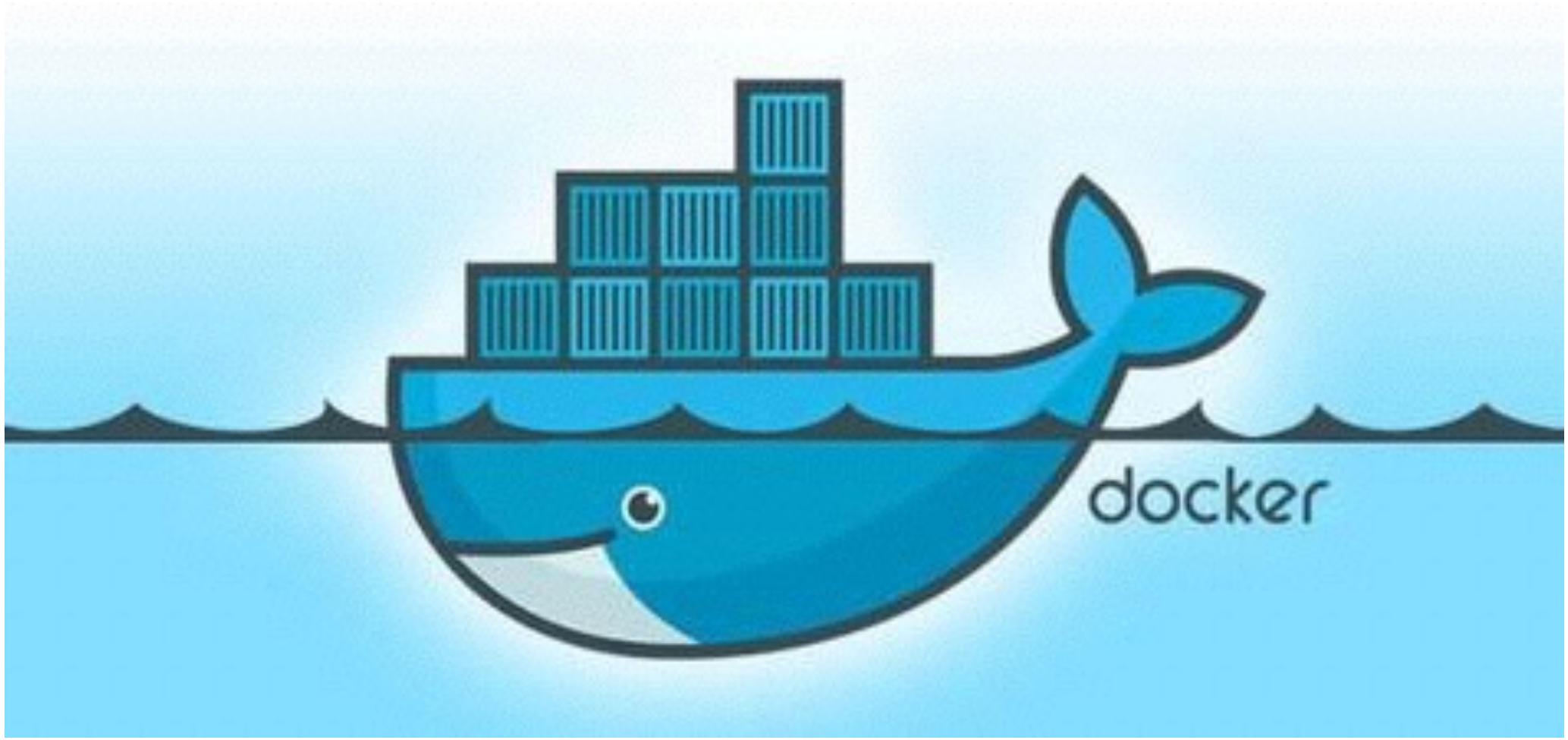
Docker卷操作接口——卷插件

- Docker Volume Plugin原理
  - Volume Plugin是由docker社区定义的一套和插件之间交互的API接口
  - 有一套卷插件发现机制（配置文件）



- Docker Volume Plugin的API
  - **VolumeDriver.Active** 握手信息，用于激活插件。
  - **VolumeDriver.Create** 创建一个卷
  - **VolumeDriver.Remove** 删除一个卷
  - **VolumeDriver.Mount** 挂载一个卷，将本地或者远端的存储**mount**本地目录
  - **VolumeDriver.Mount** 卸载一个卷
  - **VolumeDriver.Path** 获取卷的挂载目录
  - **VolumeDriver.Get** 获取卷信息
  - **VolumeDriver.List** 获取插件的所有卷

存储简介



存储简介——本地存储（可选）

本地存储

- ➢使用本机的磁盘，提供存储服务。
- ➢ZFS，Devicemapper，XFS等
- ➢功能上类似于docker的graph driver

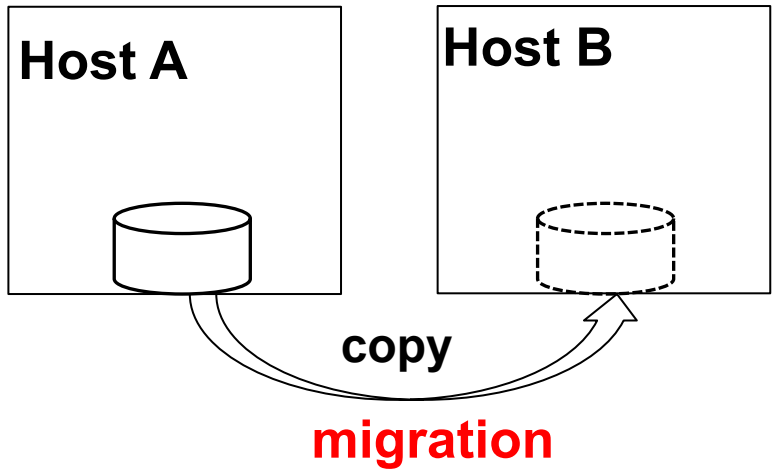
缺点：

- ➢跨主机迁移开发难度大，Cost高，效率低。
- ➢Down机后，数据无法恢复。

结论：

不便于大规模部署。

但是社区很多的插件都是本地存储。比如Flocker最开始就是基于ZFS做的。



存储简介——共享存储（可选）

按特性分类有三种：

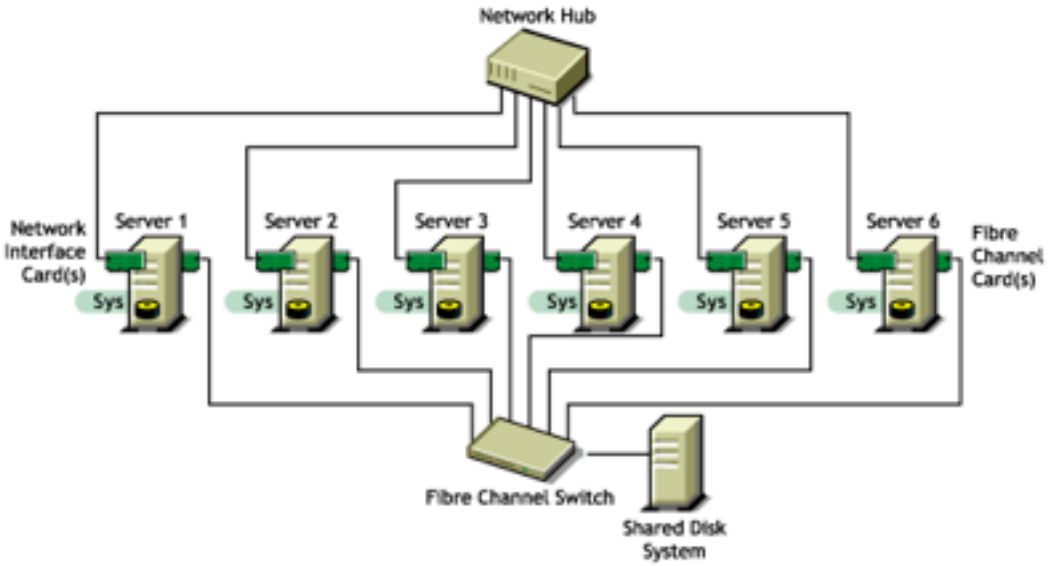
- 块存储(SAN, ceph rbd，EMC) 对应Openstack Cinder组件

  提供块设备，类似于硬盘，比如iSCSI接口的SAN，就是把SCSI接口通过internet封装，由网络设备提供

- 文件存储(NAS,) 对应Openstack的Manila组

  一般都是在块设备上，提供文件系统式服务。比如NFS，FTP等。

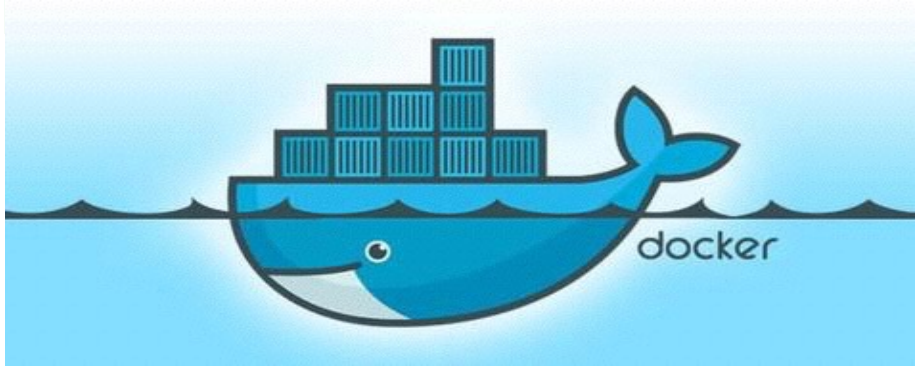
- 对象存储(UDS) 对应 Openstack的swift组



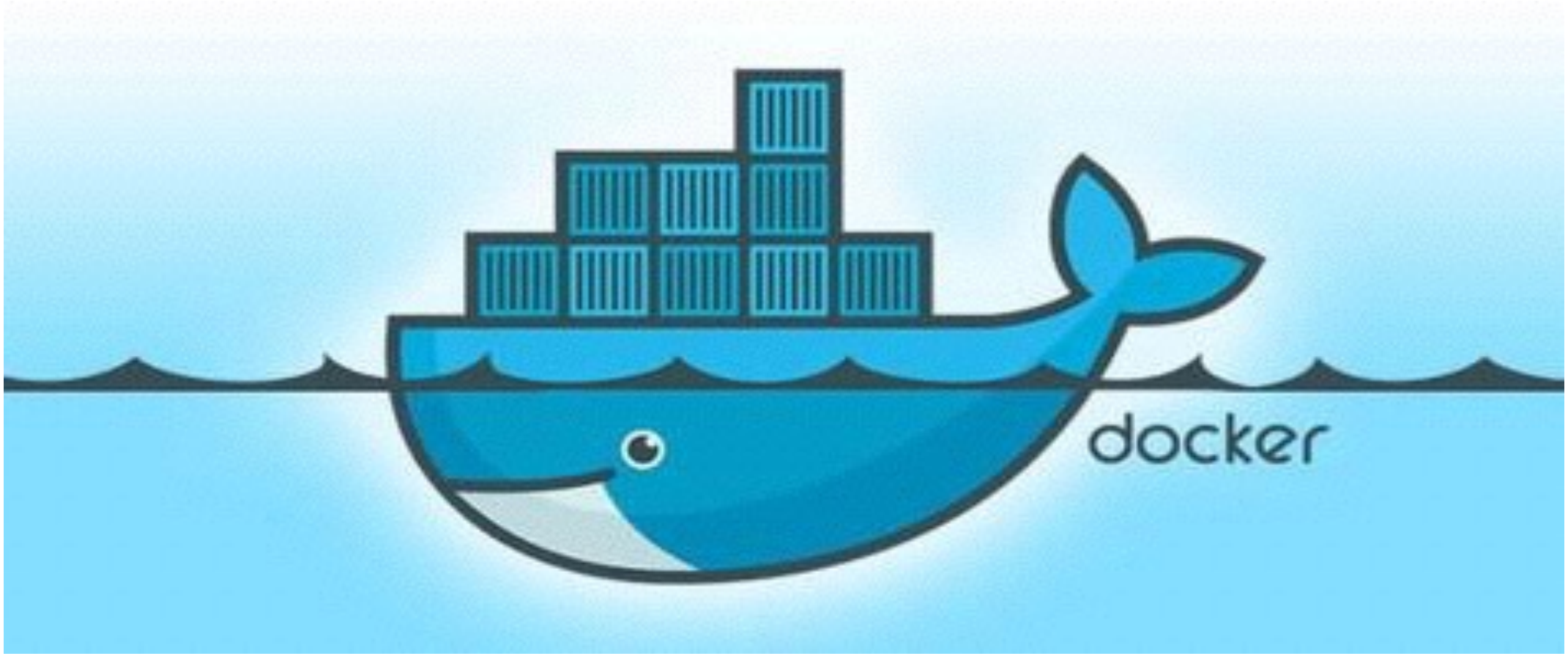
存储简介——块存储接口（可选）

- 卷操作
  - 创建一个数据卷（create）
  - 删除一个数据卷（delete）
  - 扩容数据卷（resize）
- 快照和备份
  - 创建一个数据 卷的快照（snapshot_create)
  - 删除一个数据卷（snapshot_delete）
  - 基于快照恢复一个卷(snapshot_restore)
- Qos
  - IOPS
  - 其他根据不同的产品会有不同的实现

- 存储资源池
  - 由于存储设备的空间很大（几T到几十T甚至更大）所以一般划分存储资源池的方式来管理
- 存储资源调度（软件概念）
  - 由于存储资源池化，会需要存储管理模块对资源进行管理和筛选，从而得到满足条件的池，来创建数据卷。



容器生态中的存储现状



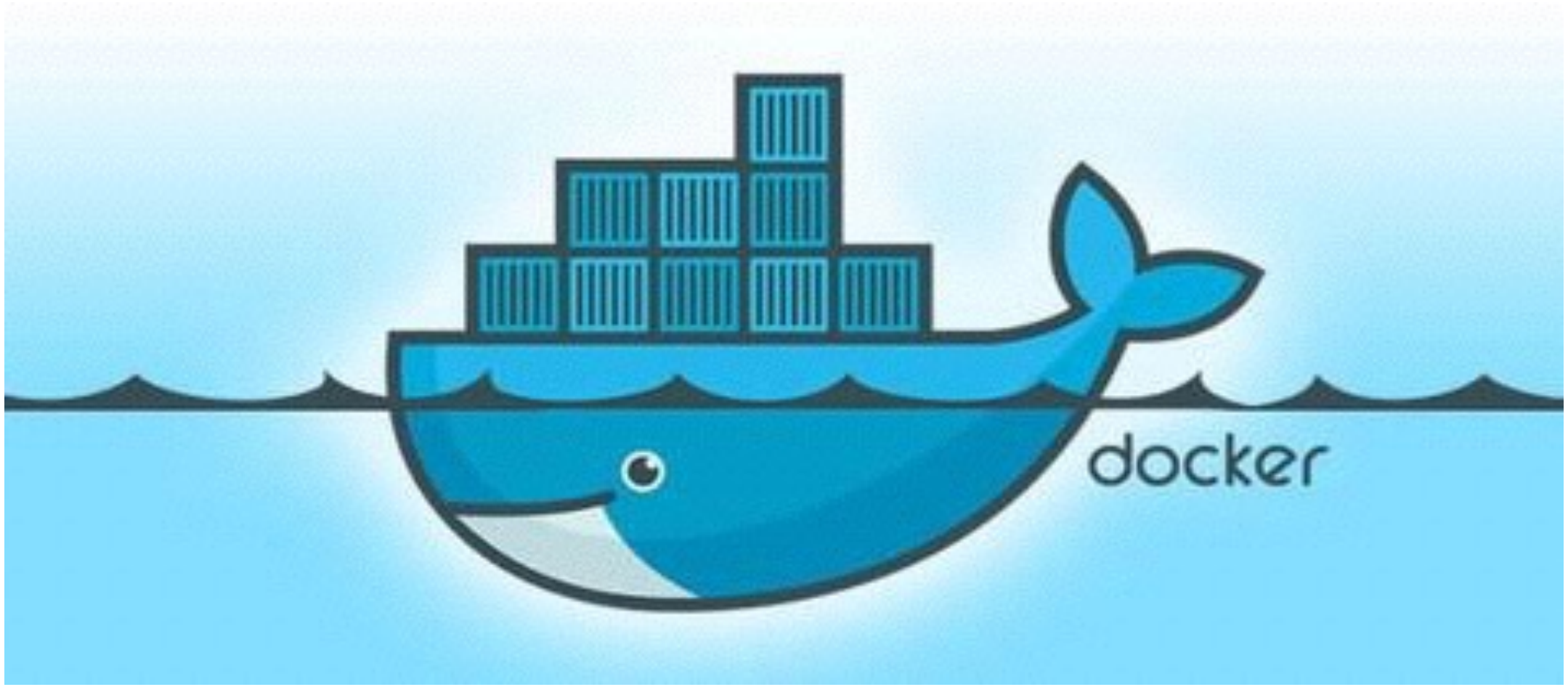
容器社区的存储弊端

- 大多卷插件都是基于本地存储
  - Convoy等支持device mapper
  - Flocker支持的ZFS
  - 好现象是这些插件都开始做共享存储（Flocker支持cinder，convoy支持AWS）
- 数据库本地化
  - 数据库本地化，会导致多节点部署，数据同步很不方便。严重阻碍大规模集群部署
- 无存储资源调度
  - 目前容器社区主流的卷插件，均无存储资源调度模块。（比如flocker只能接一种存储，而且调度要有驱动来实现，这就意味着，每种驱动都要实现一遍存储资源池化和调度的策略）
  - Openstack的cinder是有该模块的，容器生态中，就缺一个类似于cinder的存储管理模块。

为什么设计Elara

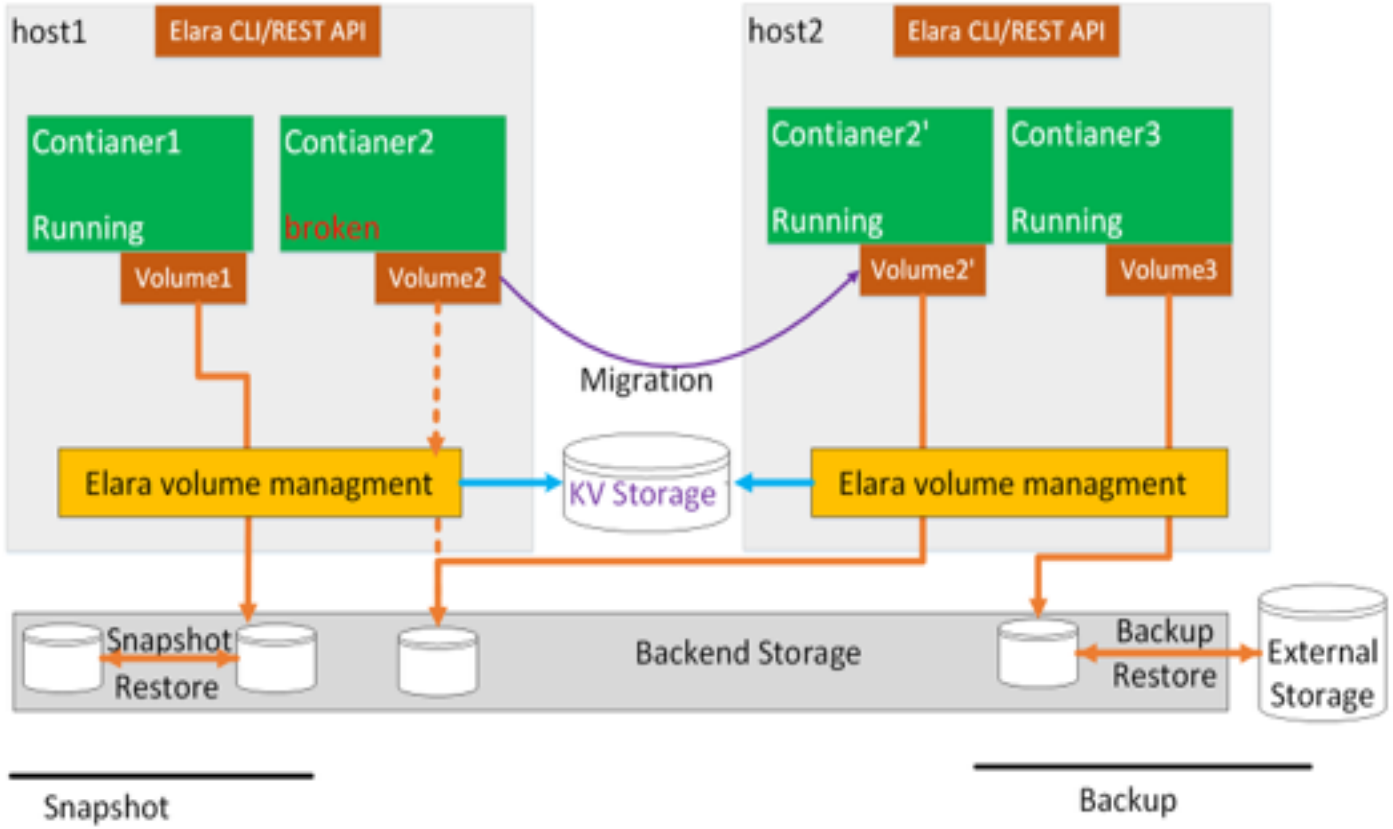
- 社区中的卷插件有各种缺陷，正如之前分析
- 为大规模集群部署提供可靠的存储支持
  - 比如公有云场景

Elara 简介



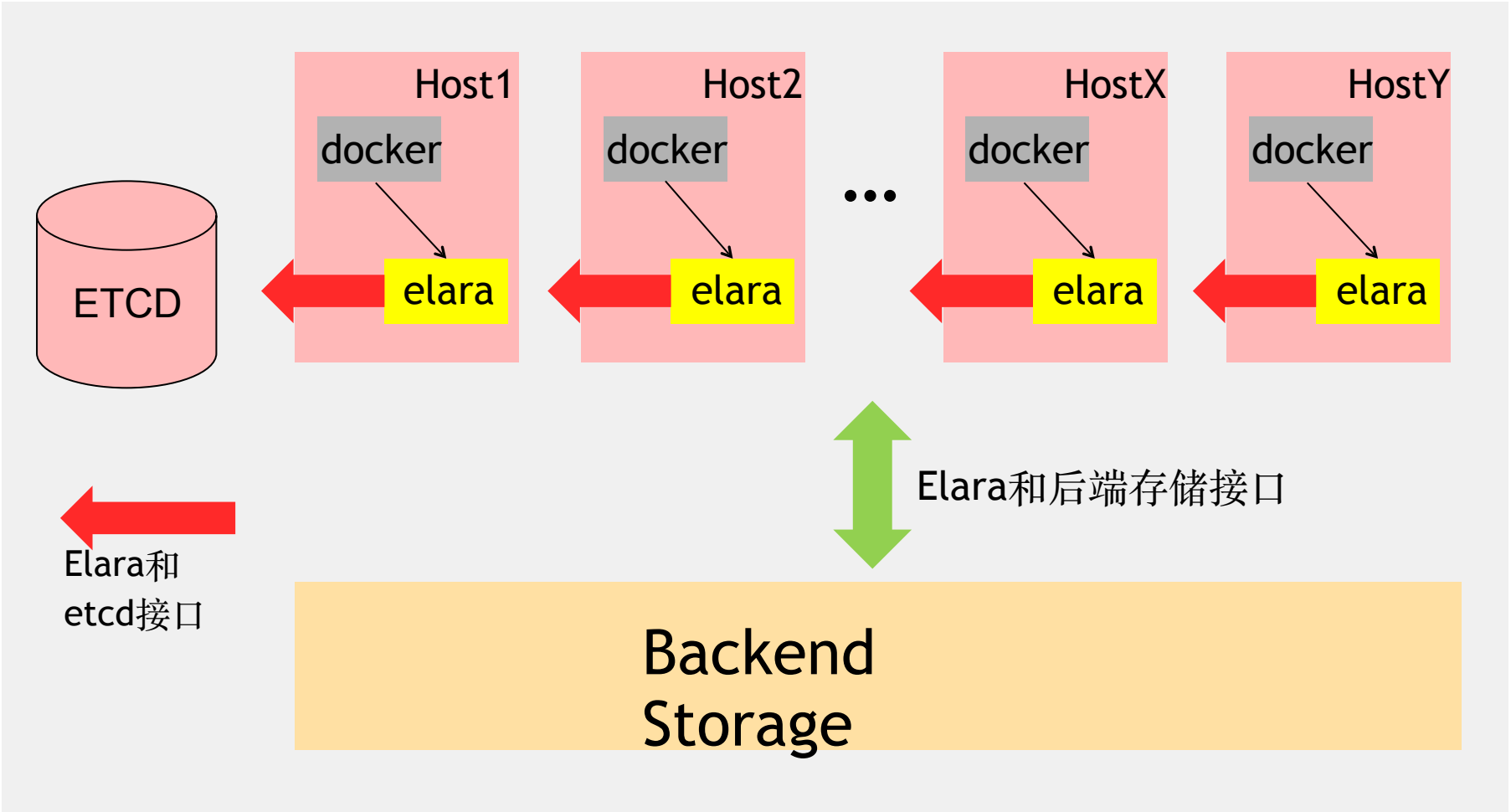
Elara 功能特性和规格

- 定位于集中存储
- 无中心节点，节点无状态，易于维护和扩展
- 高可用键值存储方案保存数据卷信息，容灾性更高
- 支持快照，备份，迁移数据卷，支持IOPS和磁盘限额
- 后端接口标准统一，易于扩展后端存储
- 支持存储资源调度
- 支持块设备，文件系统，对象存储，三种后端存储。



Elara的运行方式——节点平等&去中心化

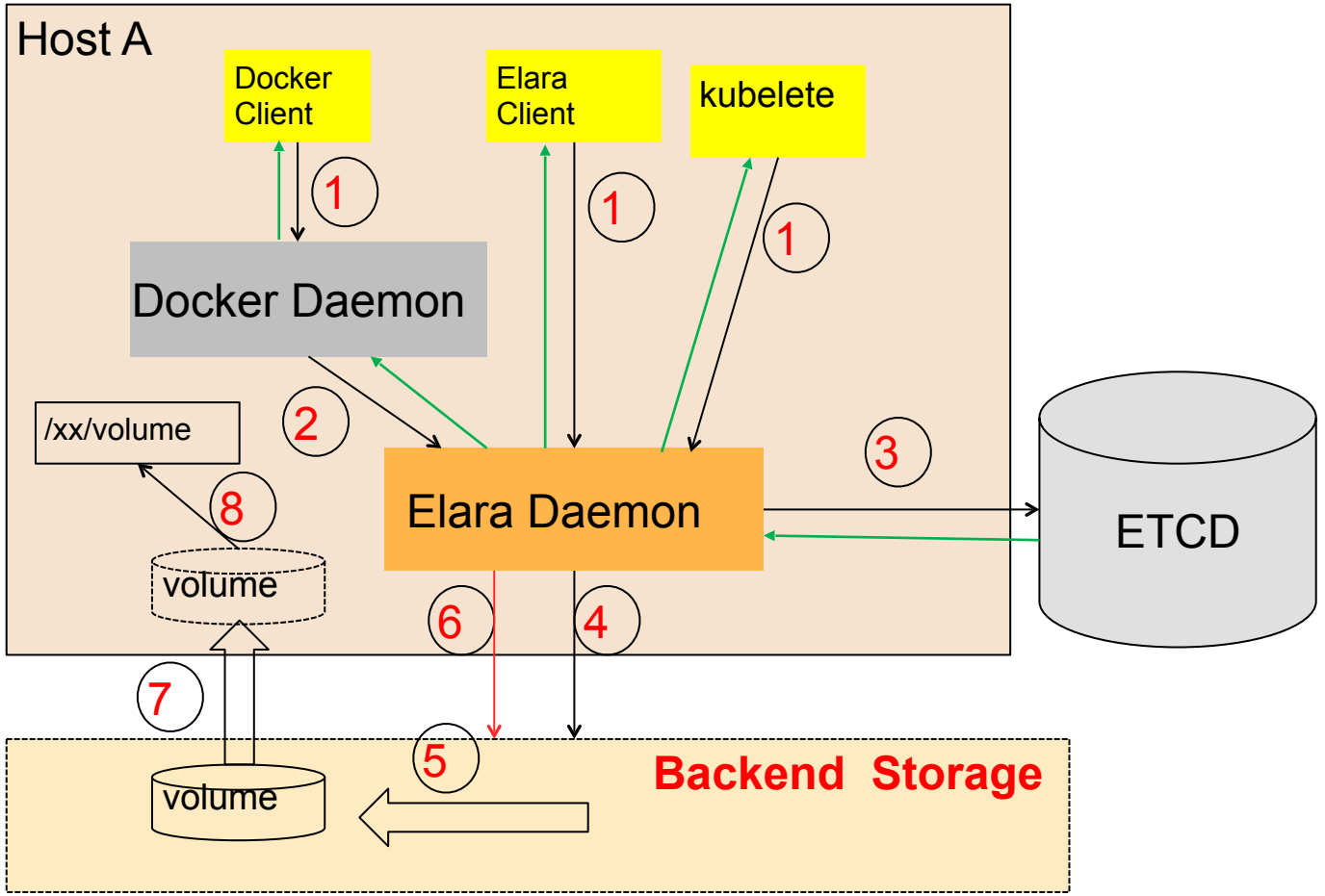
1. Elara各节点是平等的，无中心节点，节点故障影响最小
2. Elara本身无状态，故障后，直接重启service即可
3. Elara本身不关心存储也不会实现存储，只关心存储接口，降低了Elara设计的复杂度



Elara的运行方式

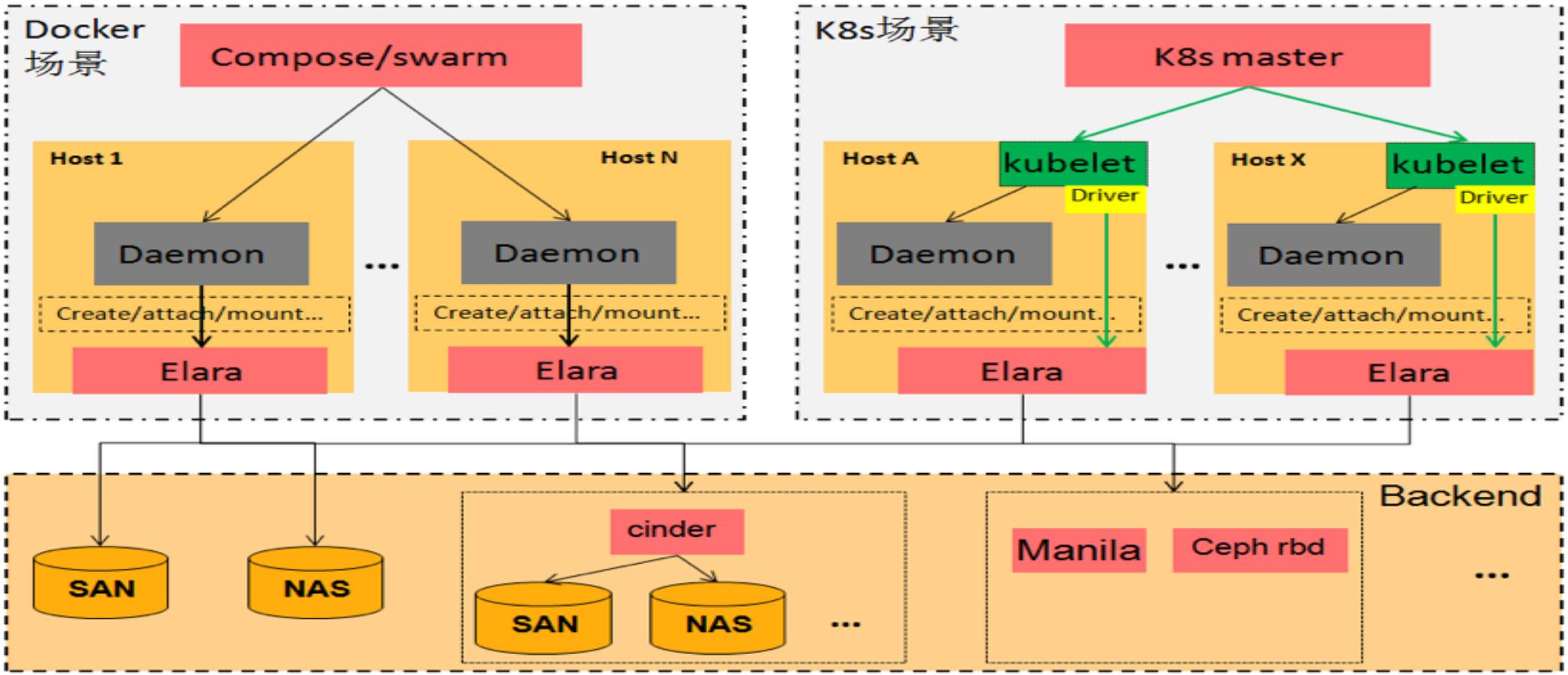
1. 发起创建volume请求
2. Docker daemon发送Volume Plugin API(VolumeDriver.Create)到Elara Deamon
3. Elara Daemon会去etcd查询并创建卷信息
4. Elara 通过计算，调度后把请求发送到特定的存储后端，去创建该Volume
5. 后端存储创建出该Volume（创建结束）
6. 和以上流程一样，elara收到上层的mount请求。之后Elara发送Attach请求，将该Volume映射到本机（本例是块设备后端）
7. 存储设备通过标准协议（比如iSCSI）将该卷映射到特定主机（比如/dev/sda
8. Elara会把该块设备格式化，并mount到某个路径，返回给docker或者其他使用者

注意：docker只能使用目录，不能直接使用设备文件 (比如/dev/sda)



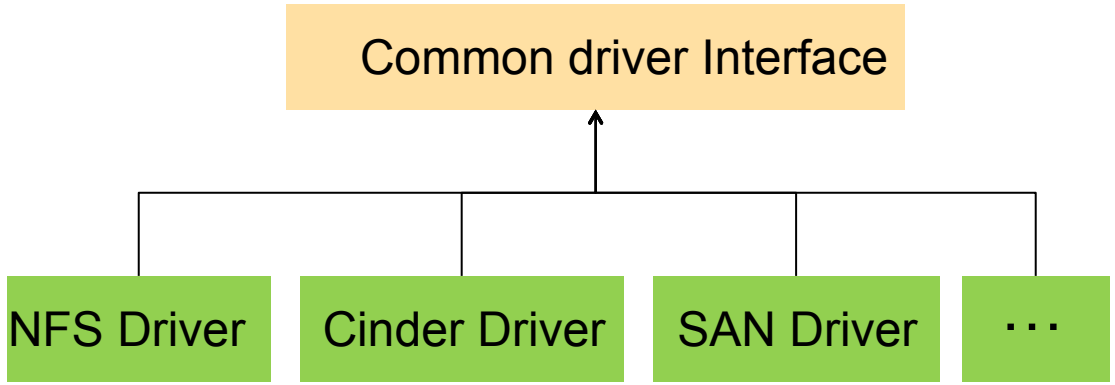
Elara的应用场景

场景定位：连接编排工具和底层存储的桥梁



Elara的对接方法(块设备为例)

Elara设计通用存储接口，对不同的存储，只需设计一个driver实现一系列存储的接口即可。
简单，方便。




```
type BlockBackendOperations interface {
    // Create, create a volume on backend.
    Create(string, ...interface{}) (string, error)

    // Delete, to delete a volume on backend.
    Delete(string, ...interface{}) error

    // SnapshotCreate, create a snapshot for a volume.
    SnapshotCreate(string, string, ...interface{}) (string, error)

    // SnapshotDelete, to delete a snapshot.
    SnapshotDelete(string, ...interface{}) error

    // Attach, to attach volume to a server, by default is local.
    Attach(string, ...interface{}) (string, error)

    // Detach, to detach volume from a server
    Detach(string, ...interface{}) error

    // Detach, to detach volume from a server
    Stats(string, ...interface{}) uint64
}
```

存储资源调度——Filter

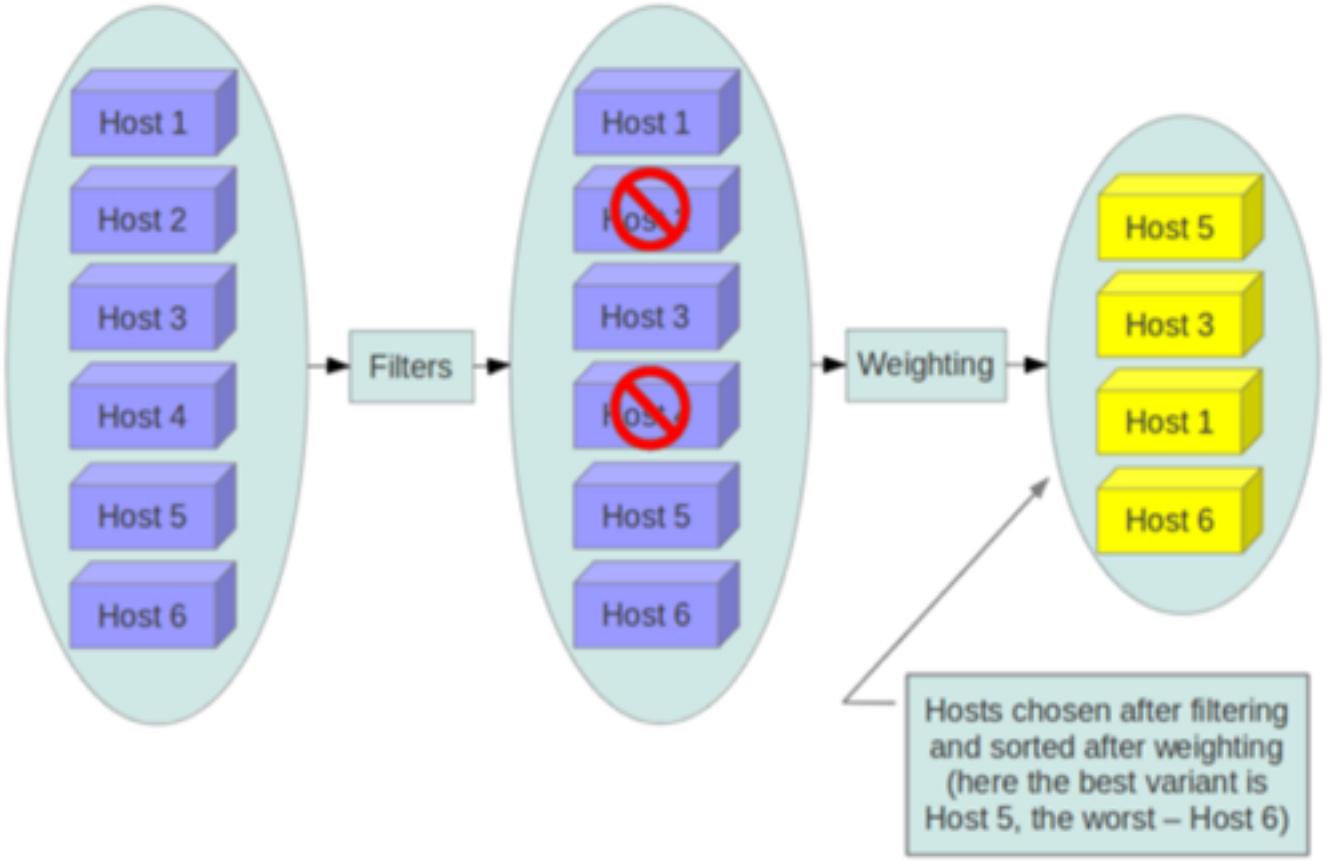
总体上讲，Elara的存储资源调度参考了Cinder的资源源调度策略，采用Filter+Weight的策略

Filter:

- 由一个通用的可以接受条件字符串的匹配规则库完成(比如接受size=10，ipos>1Gbps,type=block),然后筛选出所有合适的后端。

Weight:

- 在经过Filter筛选后，可能不止一个满足条件的后端存储符合条件，然后通过权重，给后端存储排序，权重最高的存储后端胜出

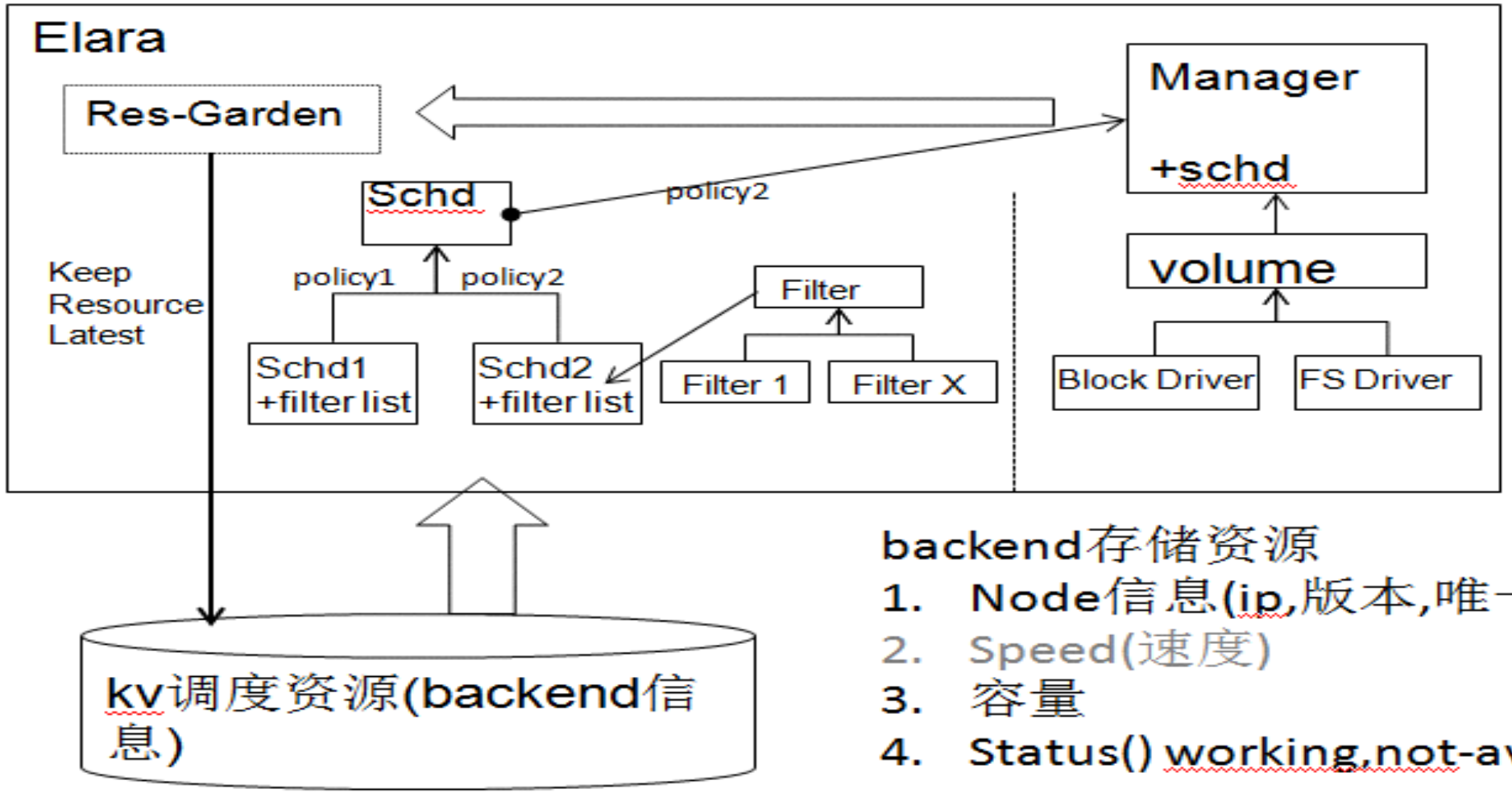


存储资源调度——Weight

weighting

- reserved_percentage：每个存储节点可以设置空闲存储容量的底限，也就是说达到这个底线时，将不再分配。比如预留10%的空间。
- free_capacity_gb：每个存储节点周期性上报该数据给所有的scheduler
- capacity_weight_multiplier：容量权重因子，即free空间大小在计算最终得分中所占的比重
- 最终free空间计算：capacity_weight_multiplier *host_state.free_capacity_gb * (1 - float(host_state.reserved_percentage/100))

存储资源调度——实现（可选）



backend存储资源

1. Node信息(ip,版本,唯一标识)
2. Speed(速度)
3. 容量
4. Status() working,not-available

和Flocker的对比

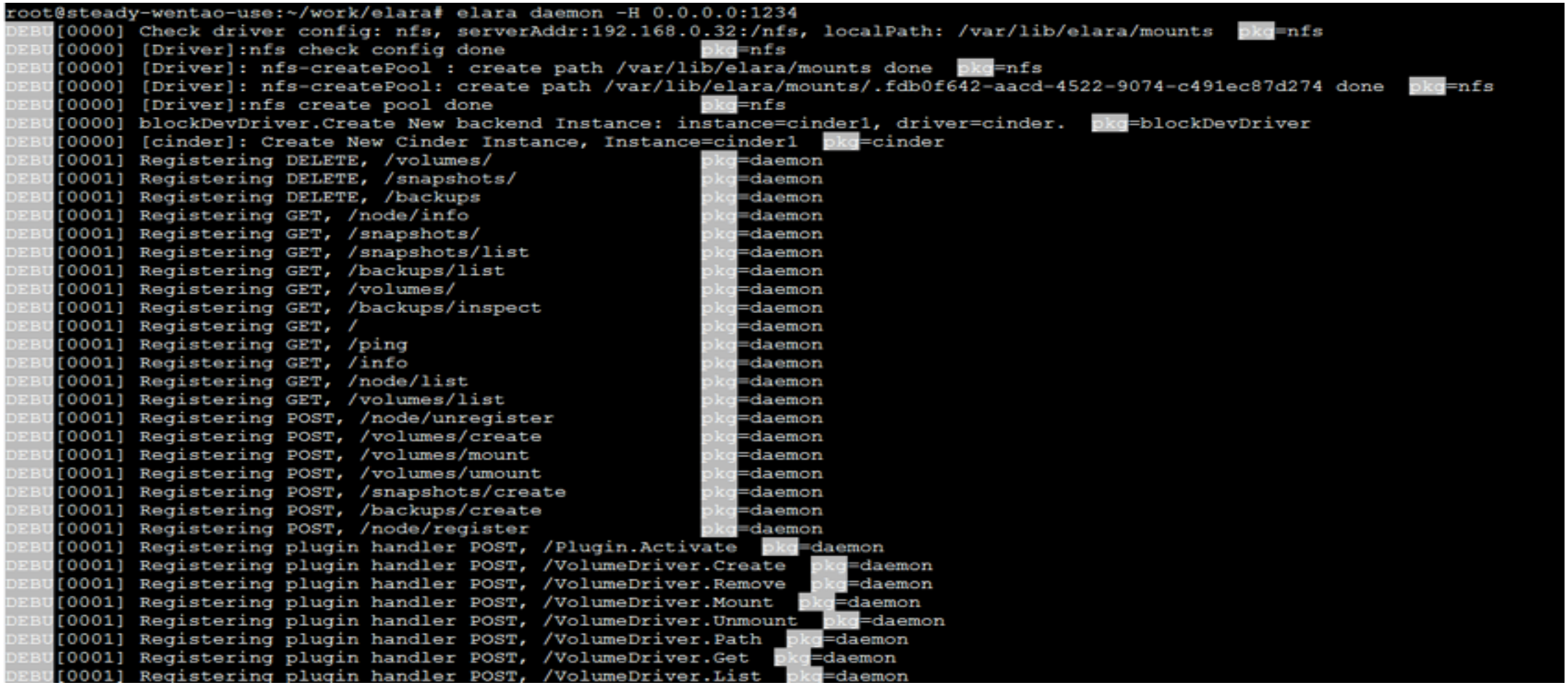
对比项	Elara	Flocker	影响
是否有中心节点	无	有	中心节点down掉全网瘫痪
节点是否有状态	无	依赖中心节点	
对存储操作接口	还支持迁移，备份snapshot等多种接口	只支持create/delete，attach/detatch 等基本后端接口	功能有局限性
存储资源调度	有	无	无法大规模接入后端存储
体积大小	很小，一个进程加一个kv存储（很多情况kv存储还可以和其他组件共用）	要部署service，agent等（flocker不光有存储所以组件比较多）	部署不方便，service很重

Elara操作实例

```
root@steady-wentao-use:~# elara -h
NAME:
   elara - A volume manager capable of snapshot and delta backup

USAGE:
   elara [global options] command [command options] [arguments...]

VERSION:
   0.9.0

AUTHOR(S):
   Euler <Euler@huawei.com>

COMMANDS:
   daemon       start elara daemon
   info         information about elara
   node         handle elara node relative commands
   volume       volume relative operatons
   snapshot     snapshot relative operations
   backup       backup relative operations
   help, h      Shows a list of commands or help for one command

GLOBAL OPTIONS:
   --socket, -s "/var/run/elara/elara.sock"     Specify unix domain socket for communication between server and client
   --debug, -d                                  Enable debug level log with client or not
   --verbose                                    Verbose level output for client, for create volume/snapshot etc
   --help, -h                                   show help
   --version, -v                                print the version

root@steady-wentao-use:~#
```

Elara操作实例——elara daemon启动

1. 和Docker一样，Elara是一个C/S结构，可以通过 elara daemon来启动elara服务，其他接口一般是client。



Elara操作实例——volume操作

```
root@steady-wentao-use:~# elara volume -h
NAME:
   elara volume - volume relative operatons

USAGE:
   elara volume command [command options] [arguments...]

COMMANDS:
   create       create a new volume: create [volume_name] [options]
   delete       delete a volume: delete <volume> [options]
   mount        mount a volume to an specific path: mount <volume> [options]
   umount       umount a volume: umount <volume> [options]
   list         list all managed volumes
   inspect      inspect a certain volume: inspect <volume>
   help, h      Shows a list of commands or help for one command

OPTIONS:
   --help, -h   show help
```

Elara操作实例——snapshot操作

```
root@steady-wentao-use:~# elara snapshot -h
NAME:
   elara snapshot - snapshot relative operations

USAGE:
   elara snapshot command [command options] [arguments...]

COMMANDS:
   create       create a snapshot for certain volume: snapshot create <volume>
   delete       delete a snapshot: snapshot delete <snapshot>
   inspect      inspect an snapshot: snapshot inspect <snapshot>
   list         list all the snapshots: snapshot list
   help, h      Shows a list of commands or help for one command

OPTIONS:
   --help, -h   show help
```

Elara操作实例——Node操作

1. Elara node主要用于管理节点，每个节点有唯一的名字来标示。
2. 目前只有注册，重命名，取消注册节点操作
3. 通过这些信息Elara可以做节点监控等功能。

```
root@steady-wentao-use:~# elara node -h
NAME:
   elara node - handle elara node relative commands

USAGE:
   elara node command [command options] [arguments...]

COMMANDS:
   info         Dump information for current node
   list         list all the Elara nodes in current domain
   register     Register this node in current domain
   unregister   Unregister this node in current domain
   rename       Rename this node in current domain
   help, h      Shows a list of commands or help for one command

OPTIONS:
   --help, -h   show help
```

《Docker进阶与实战》

机械工业出版社出版

最全面
的内容

进阶与实战
的最佳选择

华为Docker实践小组 出品

Docker社区贡献全球排名前三、国内排名第一

【团队介绍】

我们致力于打造未来ICT领域的操作系统、云平台，我们致力于研究未来ICT、NFV/SDN、云服务场景下的OS、容器、虚拟化前沿技术。

这里有Linux kernel、Docker社区多名maintainer，有ACPI标准提案committer，有精通Linux kernel各子系统（调度、内存管理、文件系统、网络等）的资深专家，有畅销书《Docker进阶与实战》的作者团队和您一起探讨容器虚拟化的未来技术。

如果你相信技术可以改变生活、改变世界；如果你喜欢Linux、喜欢钻研技术；如果你热爱开源、想成为技术大咖；别犹豫，加入我们！

Docker/容器虚拟化架构师/高级工程师（北京/杭州）

【职责】

- 负责x86、arm 64架构下容器虚拟化需求分析、关键技术和特性的设计；
- 负责Docker/容器关键技术研究和特性开发；
- 负责Docker及相关开源社区互动、运作，如推送bug fix和新特性。

【要求】

- 计算机相关专业本科及以上学历，4年以上Linux系统或内核开发经验；
- 熟悉Linux容器相关技术（namespace、cgroup）者优先；
- 熟悉Docker源码、有Docker社区开发经验者优先；
- 熟悉容器编排/调度（Kurbernetes，Mesos等）技术者优先；

【招贤纳士】【华为-Docker团队】

诚邀容器、虚拟化领域专家

【电 话】18501294585/北京、13732261657/杭州
【邮 箱】hr.kernel@huawei.com

Thank you

www.huawei.com