阿里巴巴在线技术峰会
Alibaba Online Technology Summit

我的Docker

Docker插件机制详解

Alibaba Group 阿里巴巴集团

阿里云 aliyun.com

日程

- Docker插件机制简介
- Docker数据卷与存储插件
- Docker插件系统的发展

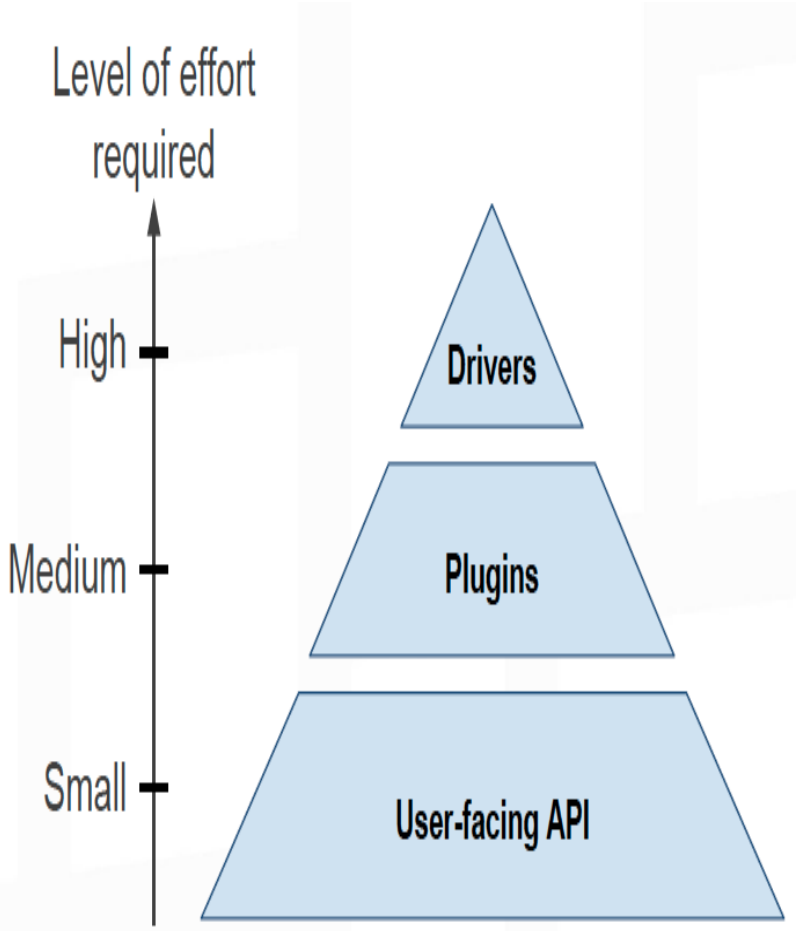
为什么需要Docker插件

开箱即用的Docker

- 自包含的镜像
- 网络、进程、文件系统等的隔离
- 安装后立马可用

不满足个性化的环境

- 使用网络存储
- 自定义保留IP地址
- 自定义容器互联网络



Docker插件简介

增加Docker引擎功能的进程外扩展

- 插件运行在docker daemon外
- 通过插件发现机制交互
- 更新插件不需要更新docker

插件类型

- 授权（authz）
- 数据卷（VolumeDriver）
- 网络（NetworkDriver）
- IP地址管理（IpamDriver）

Docker插件简介

插件发现机制

- 每个插件都包含一个http服务器
- .sock：在/run/docker/plugins下的unix socket文件
- .spec文件：纯文本文件，内容是插件访问地址
- .json文件：内容包括Name、Addr、TLSConfig
- .spec, .json文件在/etc/docker/plugins或/usr/lib/docker/plugins下
- 文件名即插件名

Docker插件简介

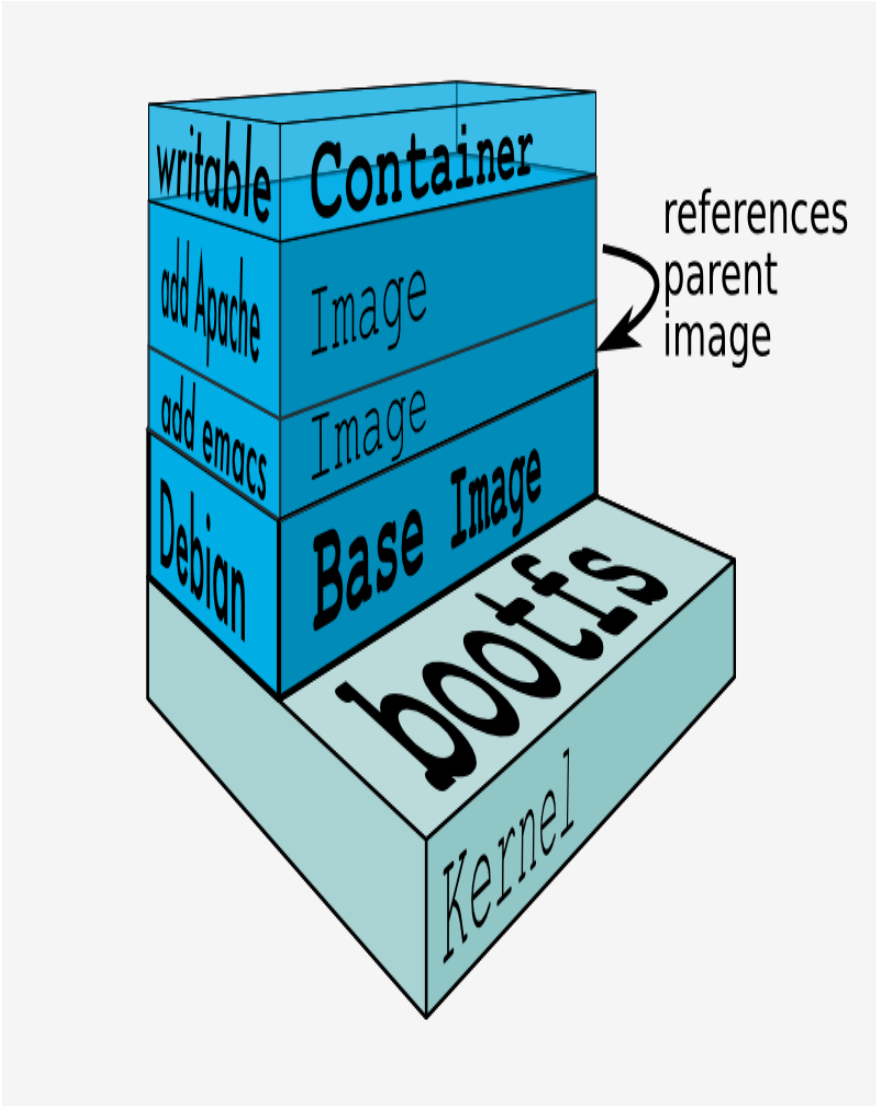
插件生命周期

- 官方文档
  - 先于docker daemon启动
  - 后于docker daemon停止
- 第一次访问插件时才激活，因此其实可以晚于docker daemon启动
- 插件SDK：docker/go-plugins-helpers

数据卷与存储插件

数据卷（Volume）

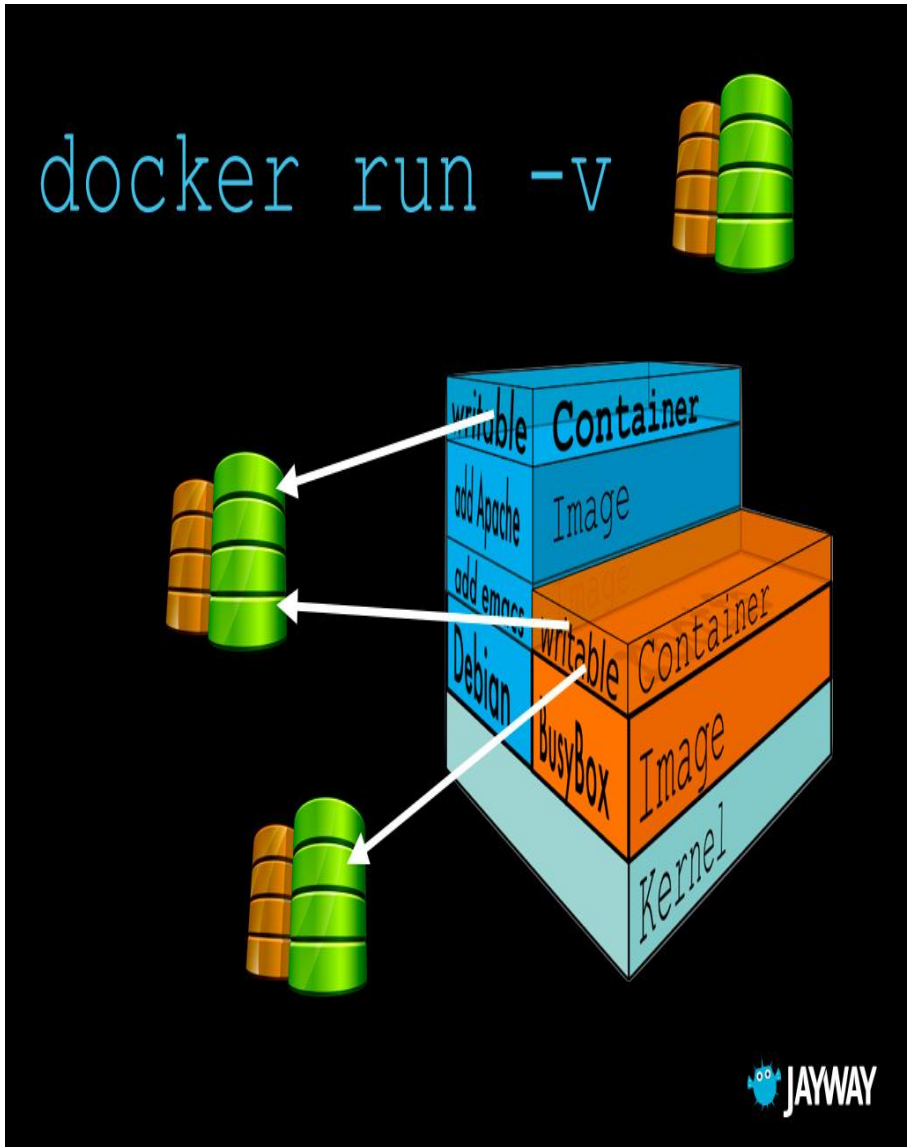
- Docker的分层文件系统
  - 性能很差
  - 生命周期跟容器相同
- 数据卷
  - Mount到主机中
  - 绕开分层文件系统



数据卷与存储插件

数据卷（Volume）

- 优点
  - 跟主机磁盘性能一样
  - 容器删除后依然保留
- 缺点
  - 仅限本地磁盘
  - 不能随容器迁移



数据卷与存储插件

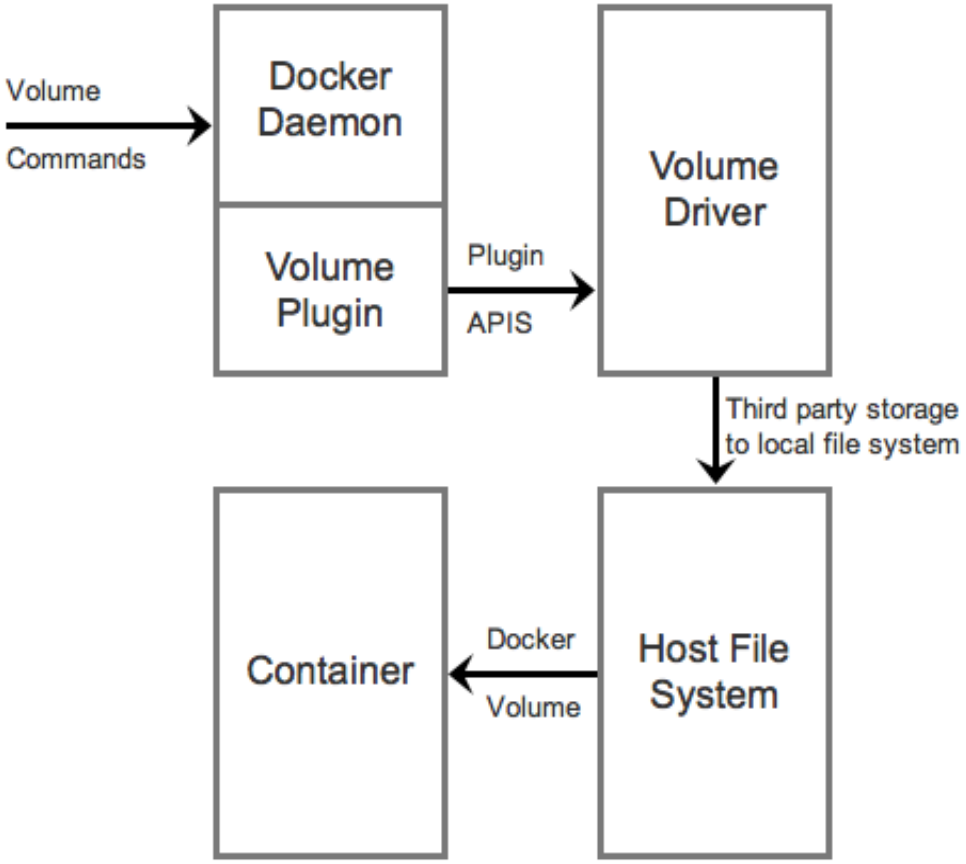
阿里云容器服务的网络存储数据卷

- 演示数据卷管理
- 演示使用oss数据卷作为wordpress的附件存储

数据卷与存储插件

存储插件

- 管理数据卷的生命周期
- 将第三方存储映射到Host本地文件系统中



数据卷与存储插件

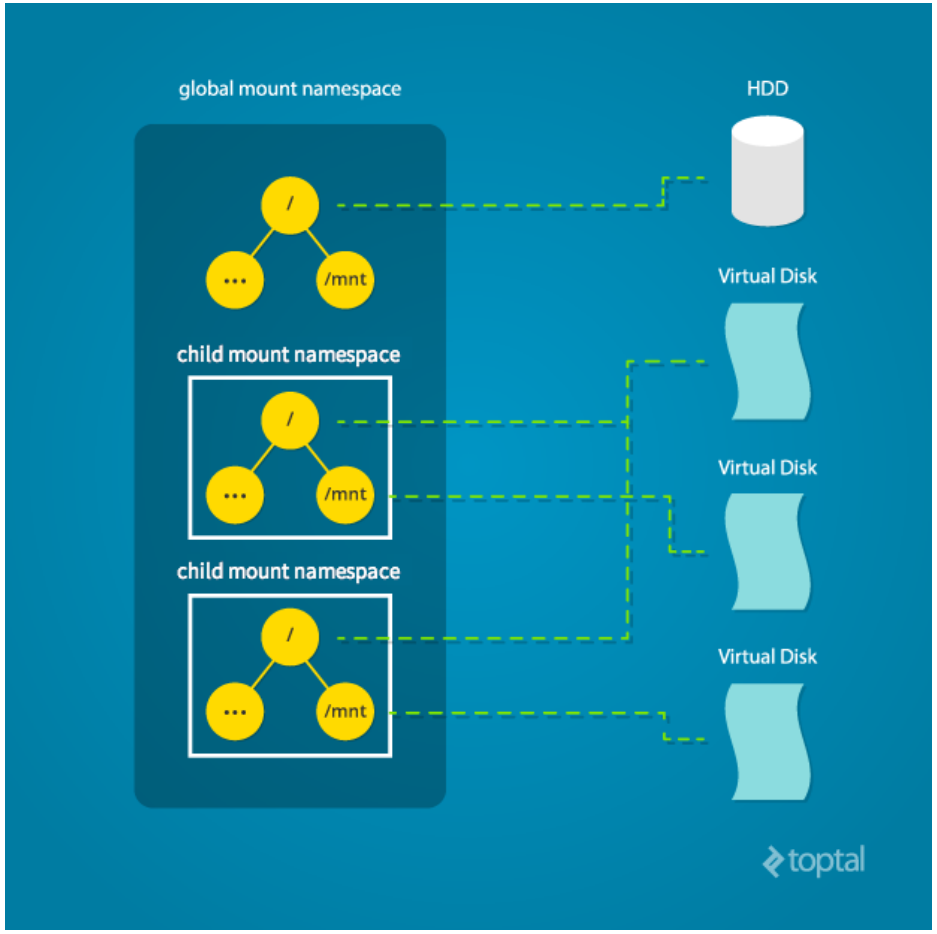
存储插件API

- /VolumeDriver.Create： docker volume create
- /VolumeDriver.Remove： docker volume rm
- /VolumeDriver.Mount： 容器每次启动时
- /VolumeDriver.Path
- /VolumeDriver.Unmount： 容器每次停止时
- /VolumeDriver.Get： docker volume inspect
- /VolumeDriver.List： 激活插件时

数据卷与存储插件

在容器中运行存储插件

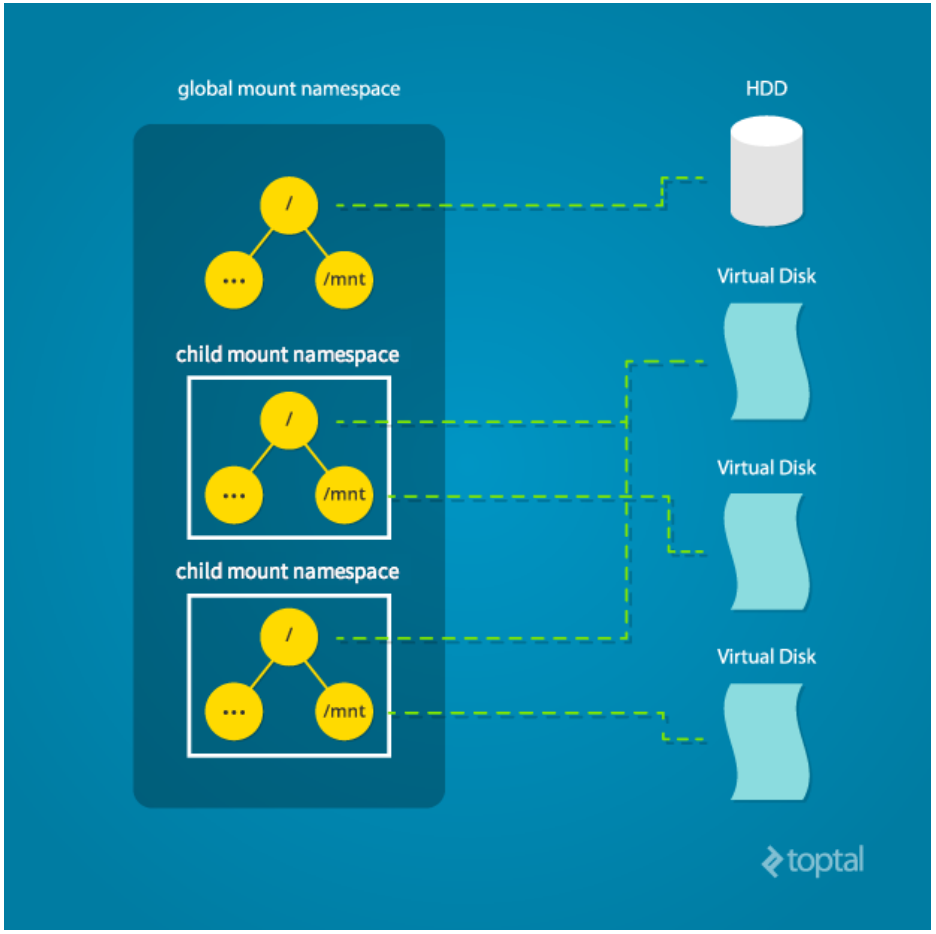
- 插件运行在主机中
  - 生命周期脱离docker
  - 简单方便
  - 执行时需要root权限，不适用于平台式场景
- 插件运行在容器中
  - 方便平台部署
  - 需要突破容器的mount namespace



数据卷与存储插件

在容器中运行存储插件

- 突破mount space
  - 在容器中修改主机的mount点
  - nsenter
  - Docker1.10开始提供shared_mount



数据卷与存储插件

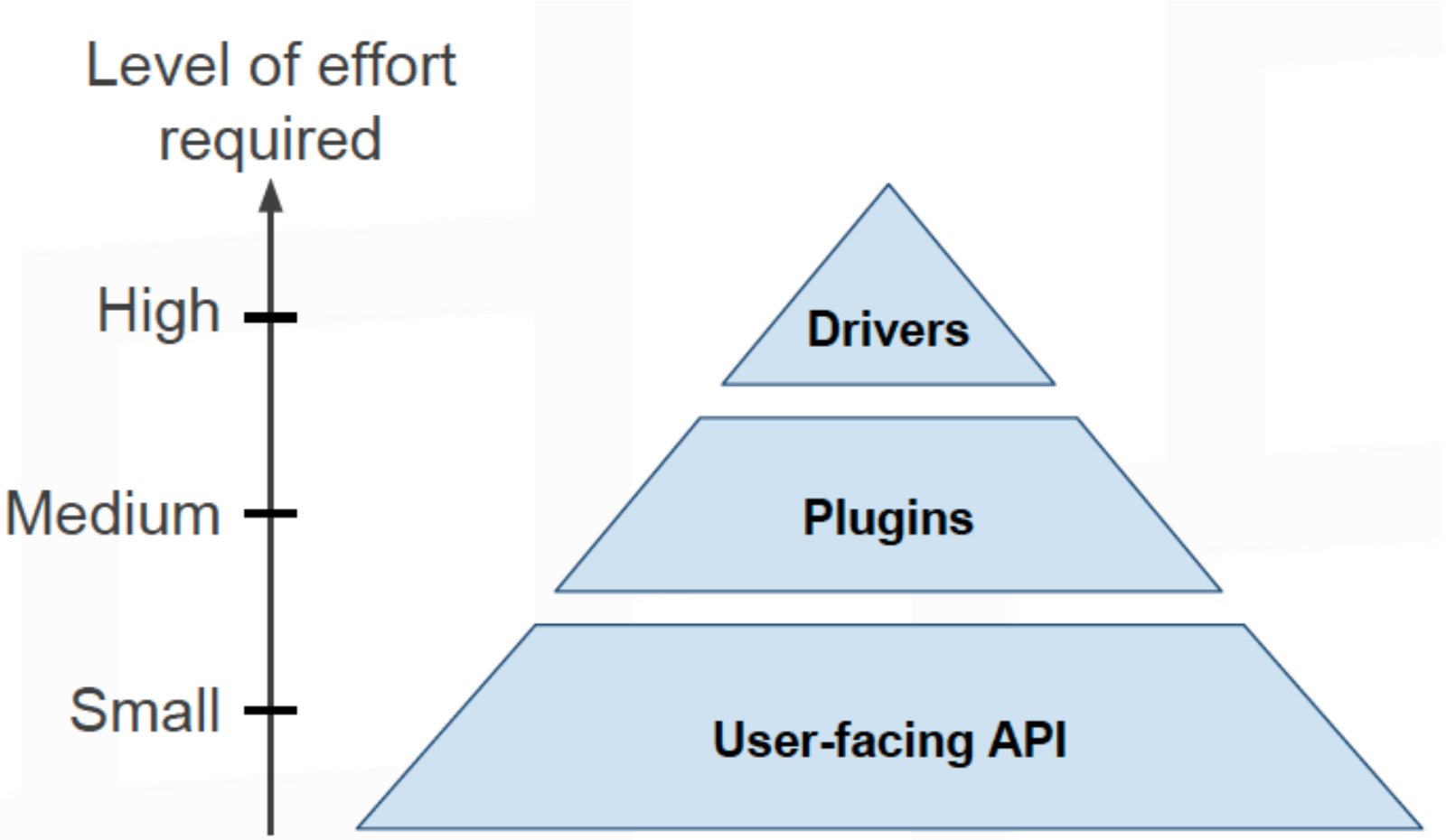
阿里云容器服务提供的数据卷

- OSSFS数据卷
  - 基于FUSE，将OSS Bucket映射为本地文件系统
- NAS数据卷
  - NFS协议，按需扩容，高性能高可靠性
- 云盘数据卷
  - 将云盘当成一个数据卷

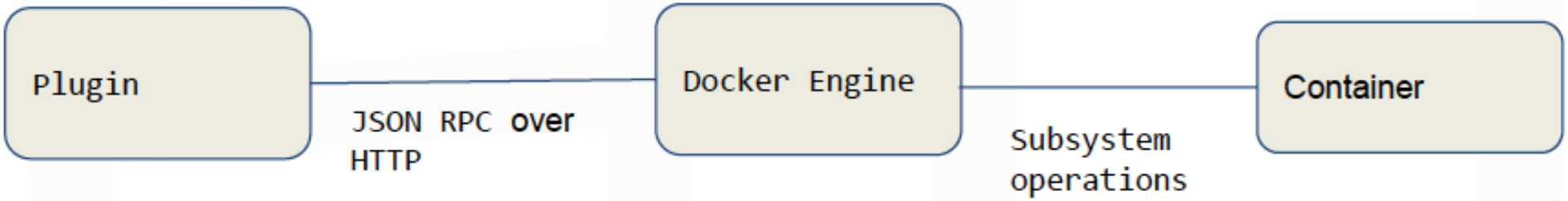
数据卷与存储插件

	优点	缺点	适用范围
OSSFS数据卷	跨主机共享	读写、ls性能低 修改文件会导致文件重写	共享配置文件 附件上传
NAS数据卷	跨主机共享 按需扩容 高性能、高可靠性 挂载速度高	成本略高	需要共享数据的重IO应用，如文件服务器等 需要快速迁移的重IO应用
云盘数据卷	性能高	不能跨主机共享 挂载速度慢	不需要共享数据的应用，如数据库、Redis等

Docker插件系统的发展



Docker插件系统的发展



历史问题	新feature
无法控制启动顺序 不能保证插件在其他容器之前启动	Highly available services 插件跟docker engine同时启动、停止
插件分发混乱，缺乏统一渠道	Docker Store 所有插件以镜像形式分发
插件缺少行为规范	

Docker插件系统的发展

docker1.13及以后

- Per node plugin
  - 保证plugin部署在每个节点
- Swarm-deployed Plugins
  - 在集群范围内部署plugin
- 提供编排插件
  - 调度策略