



前端转行 DevOps 经验分享

王龔 · 北京

内容

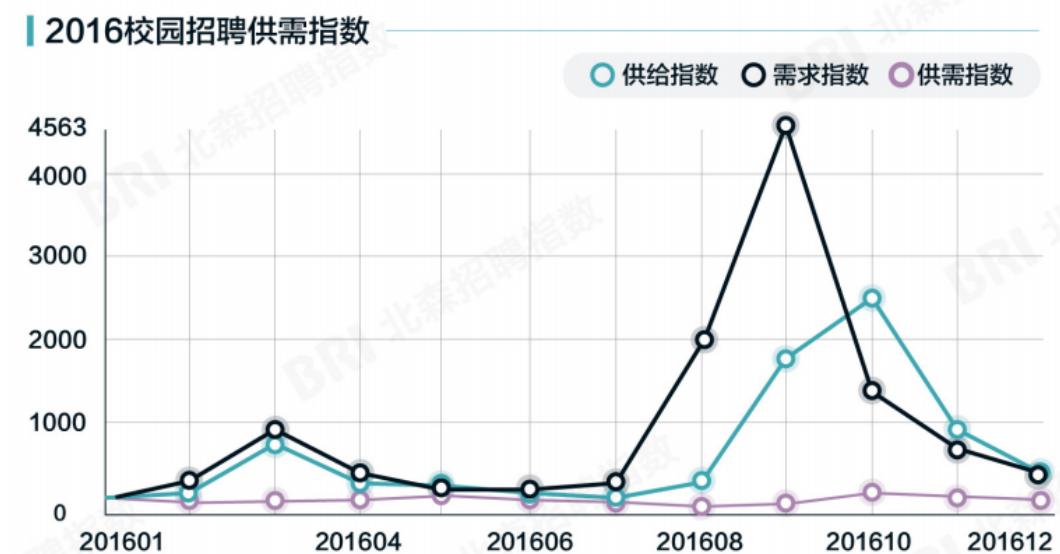
- 背景介绍
- 前端工程化到 DevOps
- DevOps 利器 Docker
- Ansible 介绍和使用
- 使用 Node.js 开发一个 Docker 插件

背景介绍

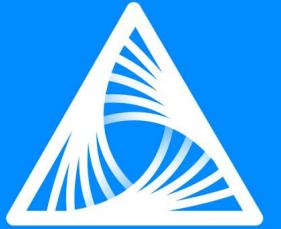


《北森2016-2017中国企业招聘指数报告》

不仅是前端工程师和程序员

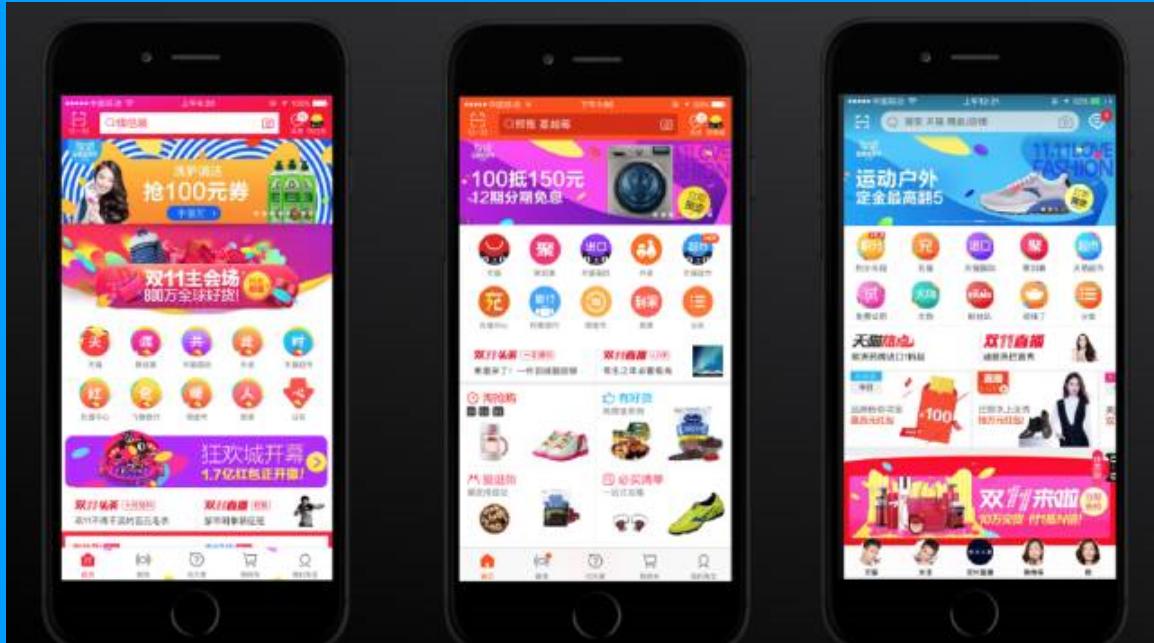


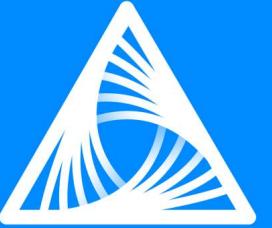
“鲁班”AI 设计系统



Alibaba AI Design Lab

- 双 11 期间 1.7 亿个 Banner





Alibaba AI Design Lab

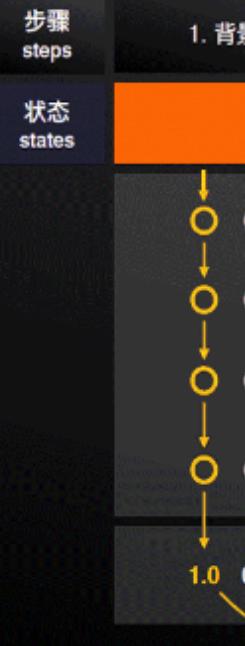
“鲁班”AI 设计系统

- 1.设计数据化：更多维，机器可学习
- 2.数据规模与质量，标注让机器可理解

3.机器学习，训练模型

4.生成树，生成设计结果并评估

5



6.文字组



5.高亮



5分



4分



3分



2分



1分



《2017互联网人才流动报告》

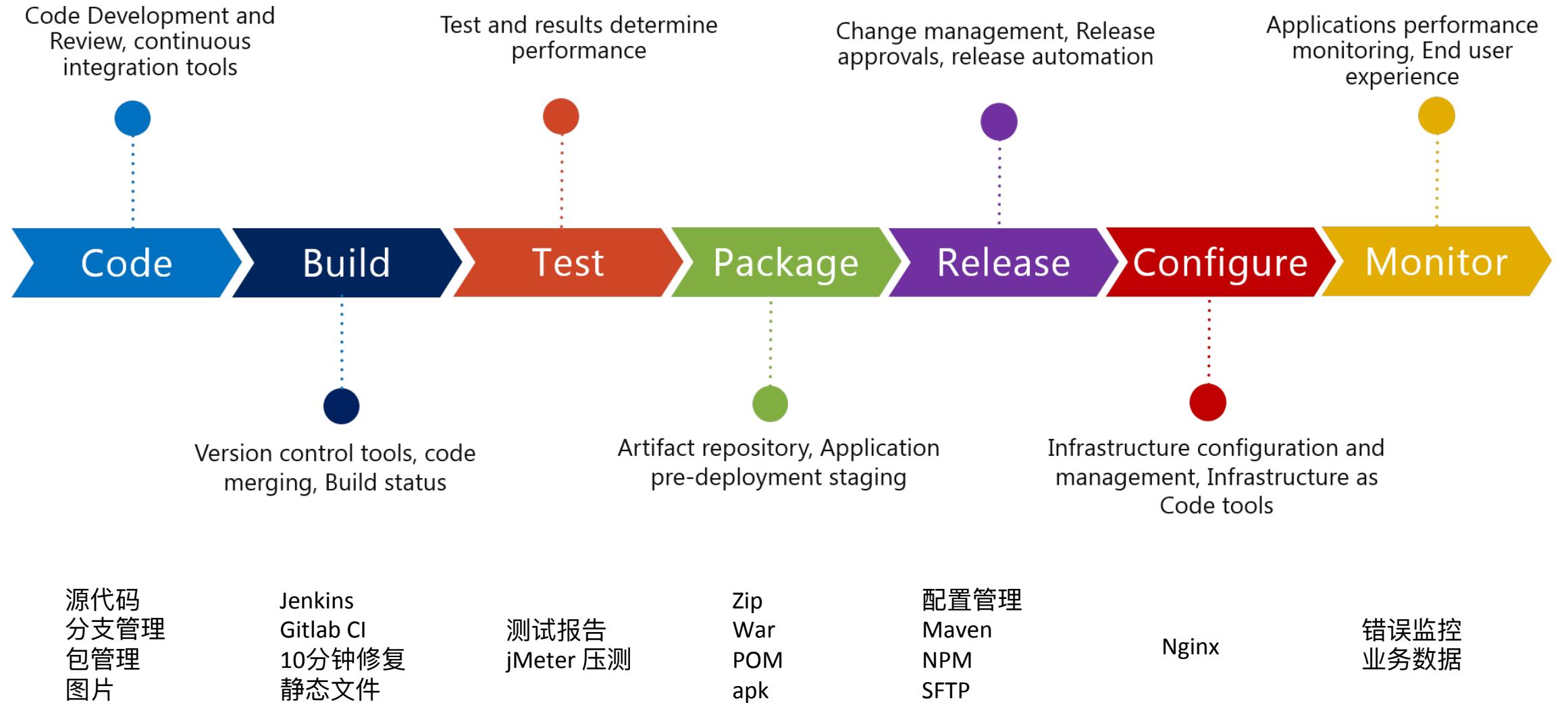


工作职责

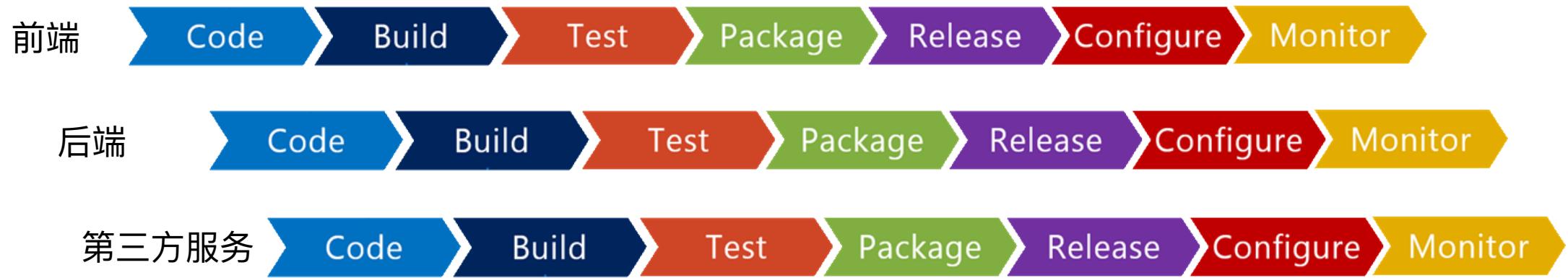
- 负责 React 前端开发
- 主导前端工程化，一键部署
- 运维了 N 多个服务，博客、网站
- 监控和业务数据分析
- 推动自动化程度的提升

前端工程化

- 技术升级
 - 框架技术选型
 - 模块化、组件化
- 构建工具
 - 预处理技术 Jade / Less / Babel
 - 文件压缩
- 持续集成
 - 单元测试
 - 语法规范
 - 每次 Push 触发 Jenkins 构建
 - 平均每天 1~2 次发布
- 持续部署
 - CDN
 - 配置管理



DevOps 流水线

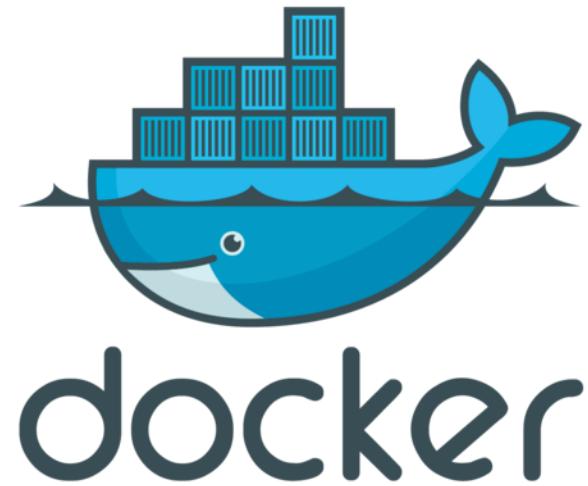


保障服务存活

- 方法
 - Nohup 挂后台
 - Daemon Process 交给操作系统
 - Supervisord 支持多个命令
 - PM2 + Keymetrics 设计超前
 - Docker
- 需求
 - 查看管理服务的状态
 - 监控 CPU, 磁盘, 网络
 - 服务发现

Docker

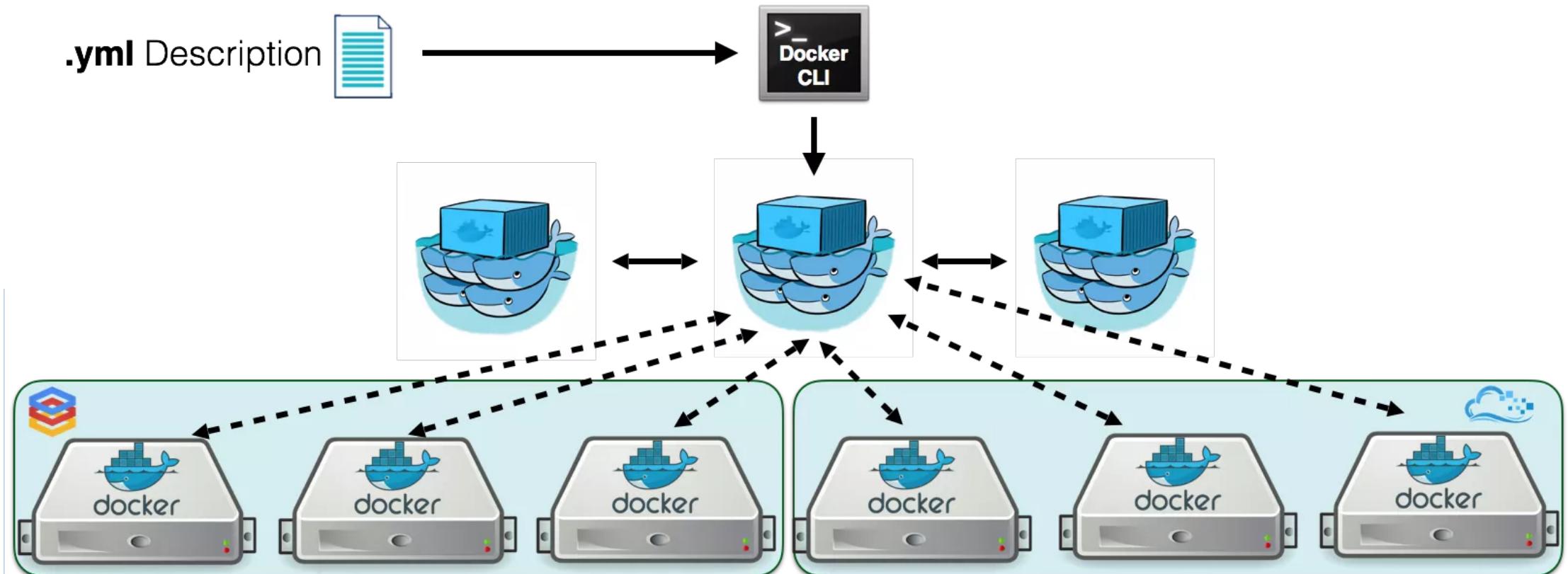
- 标准化
- 多平台
- 版本控制
- 隔离性
- 安全性



从单机到集群

- 1 台 Docker 主机
 - InfluxDB
 - Grafana
 - Sentry
 - Postgres
 - Ghost
 - Metabase
 - MongoDB
- 十几个容器，负载不断上升
- 升级重启需要花费 15 分钟
- 调研集群方案，最终确定 Swarm
 - 最小成本 Hold 住

Docker Swarm

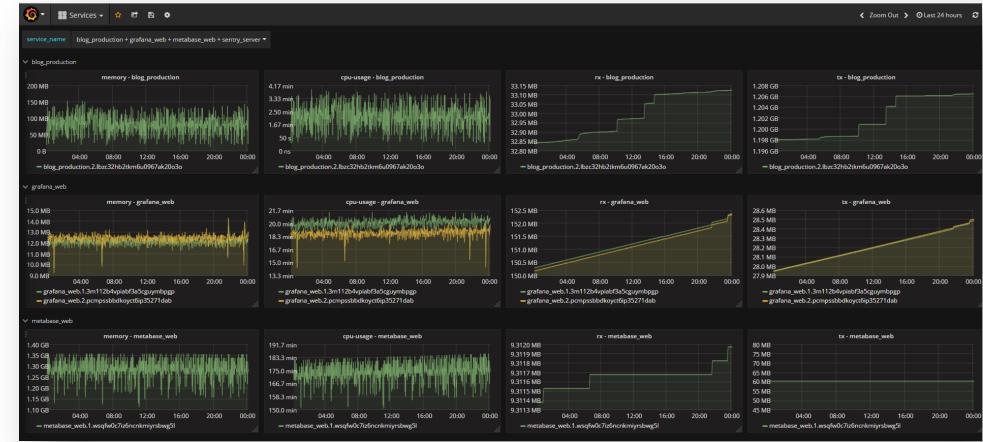


关于集群

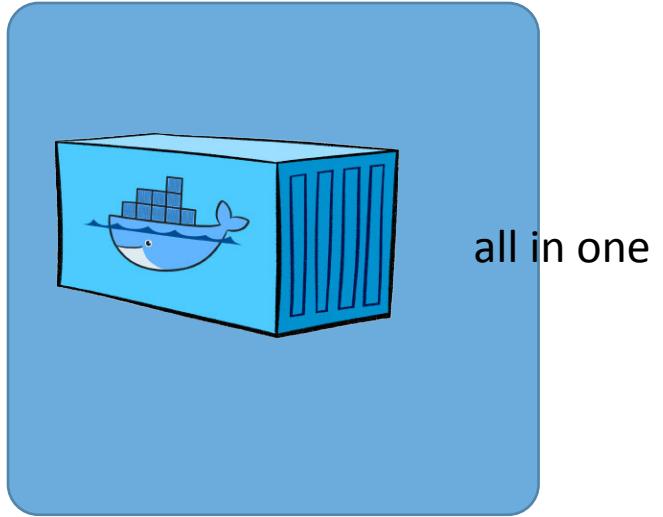
- 权限
 - 自签名证书
 - Portainer
- 监控
 - cAdvisor + InfluxDB + Grafana
- 日志
 - ELK Stack + Gelf Driver
- 负载均衡
 - Traefik
 - *.swarm.company.com

修改服务以更好的适应集群

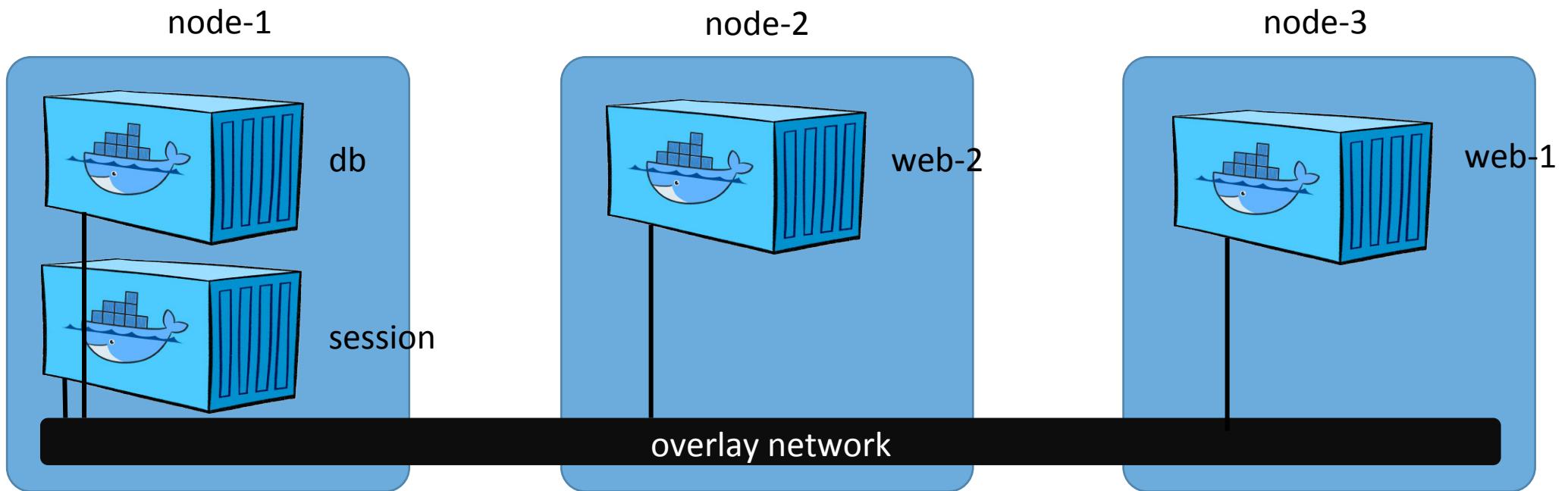
- 以 Grafana 为例
- 改动前
 - 只有一个容器 App
 - 使用 SQLite 数据库
- 改动后
 - 使用三个容器 Web / DB / Session
 - Web 容器可以水平伸缩
 - 数据库迁移到 Postgres
 - 使用 Memcache 做粘性的 Session



单节点



集群



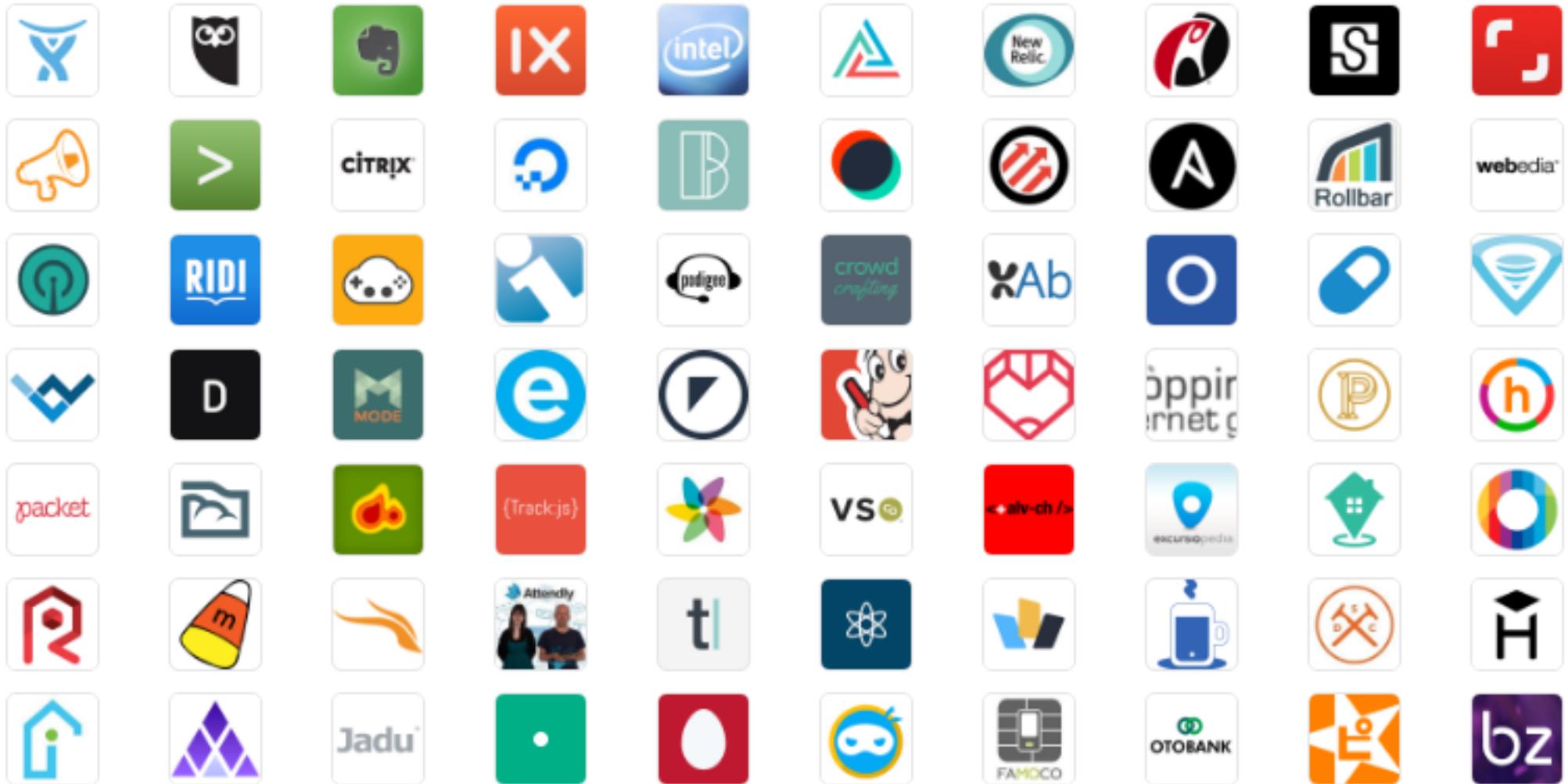


Ansible – 自动化运维工具

- 特点

- 基于 Python 和 SSH 不用安装 Agent
- Inventory 支持主机分组管理
- 能够批量执行命令
- 封装了安装软件包等常用操作
- 支持 YAML 格式的 Playbook 剧本

使用 Ansible 的公司



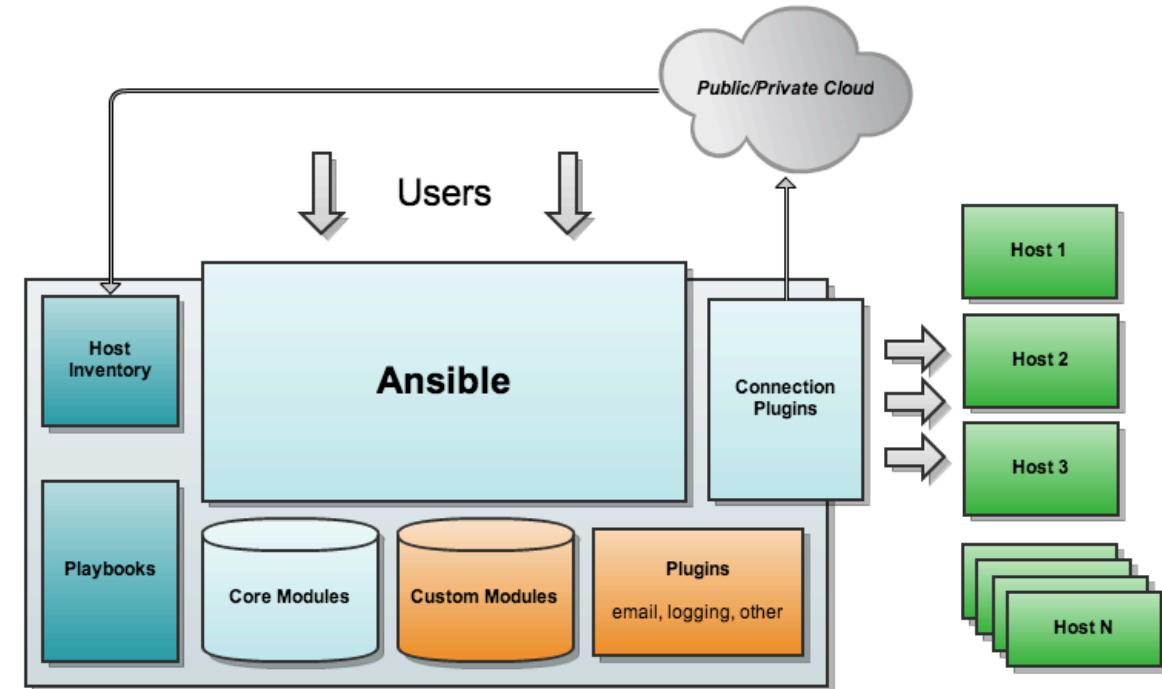
使用 Ansible 安装 Nginx

```
---
- hosts: web-server
  become: true
  tasks:
    - name: 使用默认配置安装 Nginx
      apt:
        name: nginx
        state: present
```

- 命令行
 - ansible-playbook install_nginx.yml
- 可读性
 - 使用 YAML 编写 Playbook
- 声明式
 - 类似于单元测试里的断言 assert. true (nginx is present)
- 幂等性

Ansible 架构

- Ansible Core 核心
- Host Inventory 主机列表
- Playbooks 剧本
- Modules 模块 (900+)
- Plugins 插件
- Ansible Tower 可视化



Inventory

```
[web-servers]
```

```
10.1.1.1 ansible_ssh_port=22 ansible_ssh_user=root
```

```
10.1.1.2
```

```
10.1.1.3
```

```
[db]
```

```
10.1.1.4
```

- 每台主机都可以有自定义的配置
- 检查所有机器连通性
 - ansible all -m ping

使用 Ansible 安装 Docker

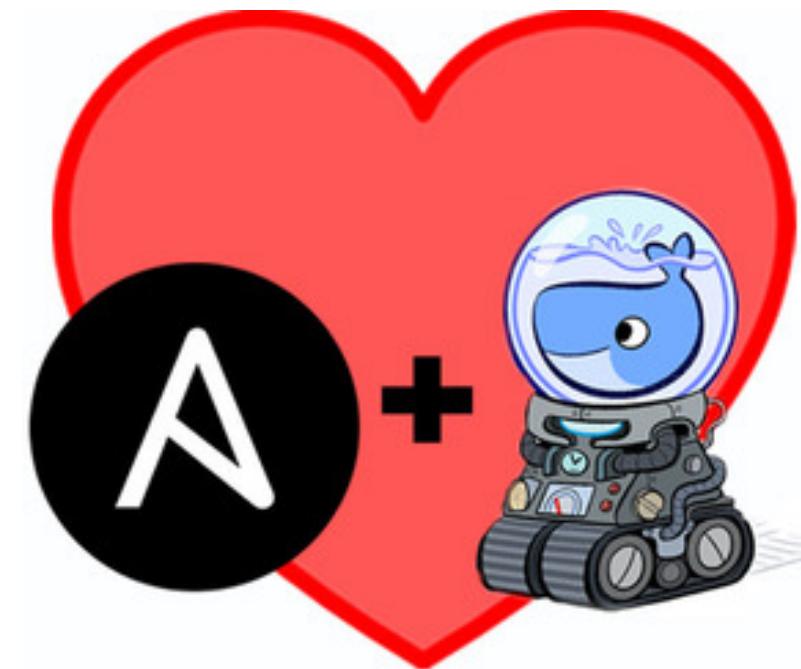
```
- name: 安装 Docker
  shell: curl -sSL https://get.daocloud.io/docker | sh
- name: 确认 user 在 docker 组里
  user:
    name: user
    state: present
    groups: docker
    append: yes
- name: 安装 Docker Compose
  get_url:
    url: https://get.daocloud.io/docker/compose/releases/download/1.11.2 ...
    dest: /usr/local/bin/docker-compose
    mode: 0755
```

```
- name: 分发配置
  copy:
    src: "./config/docker.service"
    dest: "/lib/systemd/system/docker.service"
    owner: root
    group: root
    mode: 0644
- name: 重启 Docker 服务
  systemd:
    name: docker
    daemon_reload: yes
    state: restarted
- name: 查看版本
  command: docker -v
  register: i
- debug: var=i.stdout
```



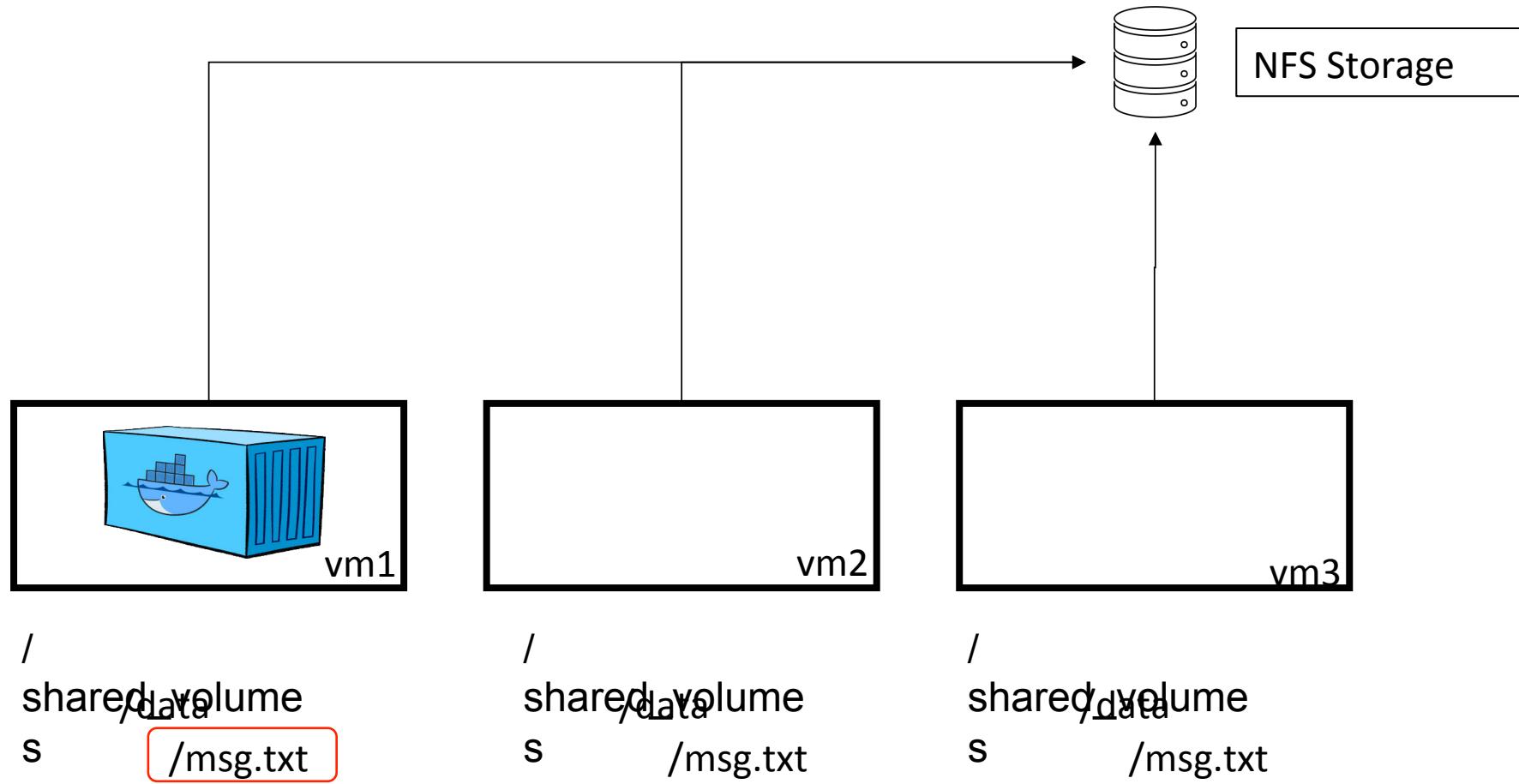
Ansible and Docker

- Ansible 针对操作系统
 - 通过 docker-container 内置模块支持 Docker
 - 利用 docker-py pip 模块和 Docker API 通信
- Docker 针对 Container / Image
- 可用 Ansible 安装 Docker 环境
- 可用 Ansible 管理容器生命周期



使用 Node.js 开发 Docker 插件？

添加 Docker 对 NFS 的支持，使得容器可以在集群内自由迁移



```
docker run --volume-driver nfs -v data:/data -it alpine "echo 1>/data/msg.txt"
```

Docker 插件

- Docker 插件是独立于 Docker Engine 进程的扩展，能够增强 Docker 的功能
- 插件可以运行在容器里也可以运行在容器的外面（插件先启动）
- Docker 通过搜索指定的目录发现插件
 - /etc/docker/plugins/nfs.spec
内容就一行 `tcp://127.0.0.1:3245`
- Docker 插件会在首次使用的时候被激活

Docker Engine 和插件的通信

- 实现这几个 HTTP API
 - /Plugin.Activate
 - /VolumeDriver.Create
 - /VolumeDriver.Remove
 - /VolumeDriver.Path
 - /VolumeDriver.Mount
 - /VolumeDriver.Get
 - /VolumeDriver.List
- 具体代码

```
app.post("/VolumeDriver.Get", (req, res) => {
  const mp = path.join("/shared_volumes",sanitize(req.body.Name));
  if (fs.existsSync(mp)) {
    res.json({
      "Volume": {
        "Name": req.body.Name,
        "Mountpoint": mp,
        "Status": {}
      }
    });
  } else {
    res.json({
      "Err": "Not Exist"
    });
  }
});
```

后话

- 测试完 NFS 性能后开源
- 为什么不直接用 mapping 而是用 volume
 - 管理功能
 - 结合 ZFS 做快照
 - 结合云存储做备份
- 跳出舒适区， DevOps 路漫漫，一起学习吧

谢谢