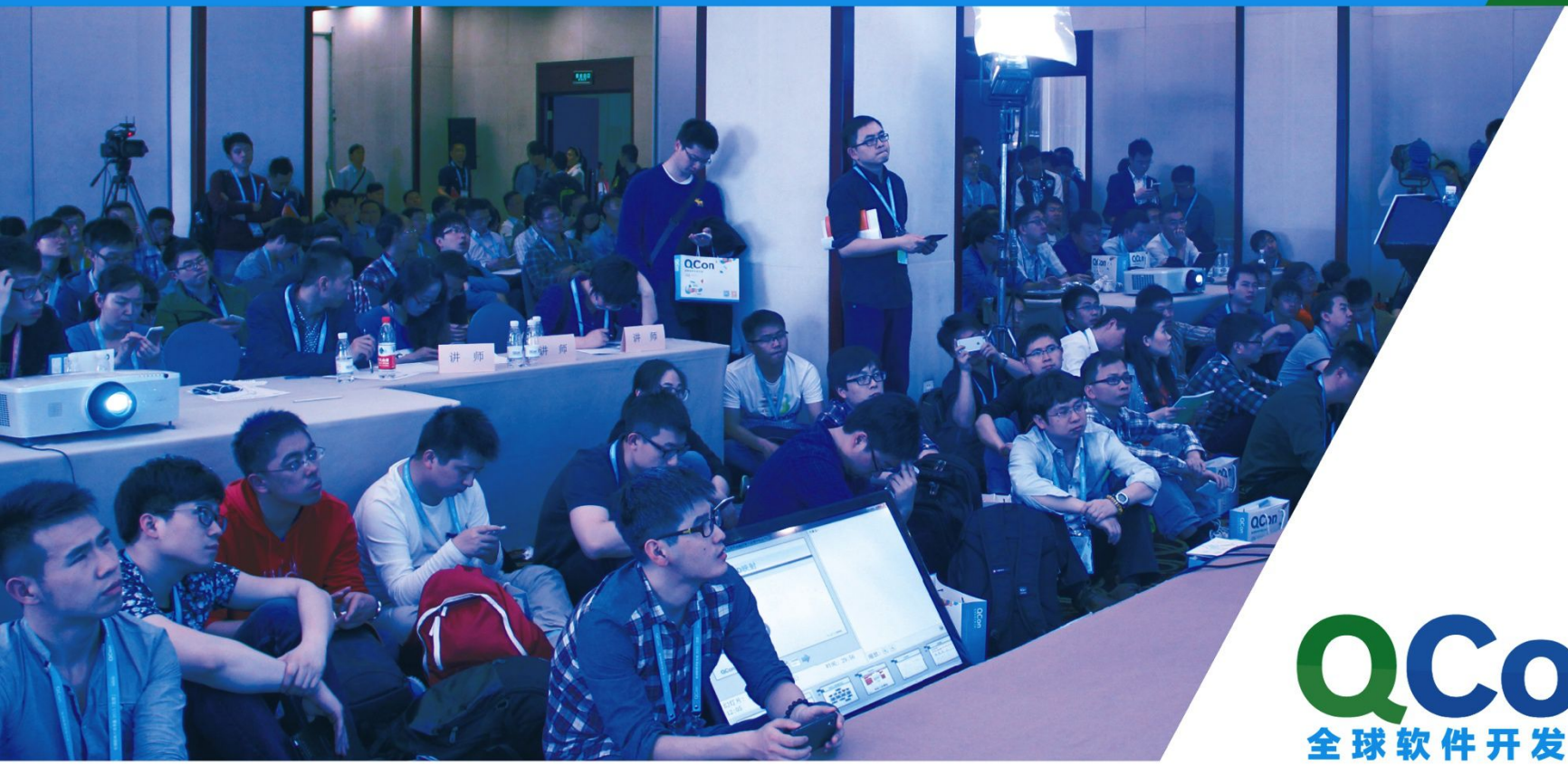
QCon全球软件开发大会

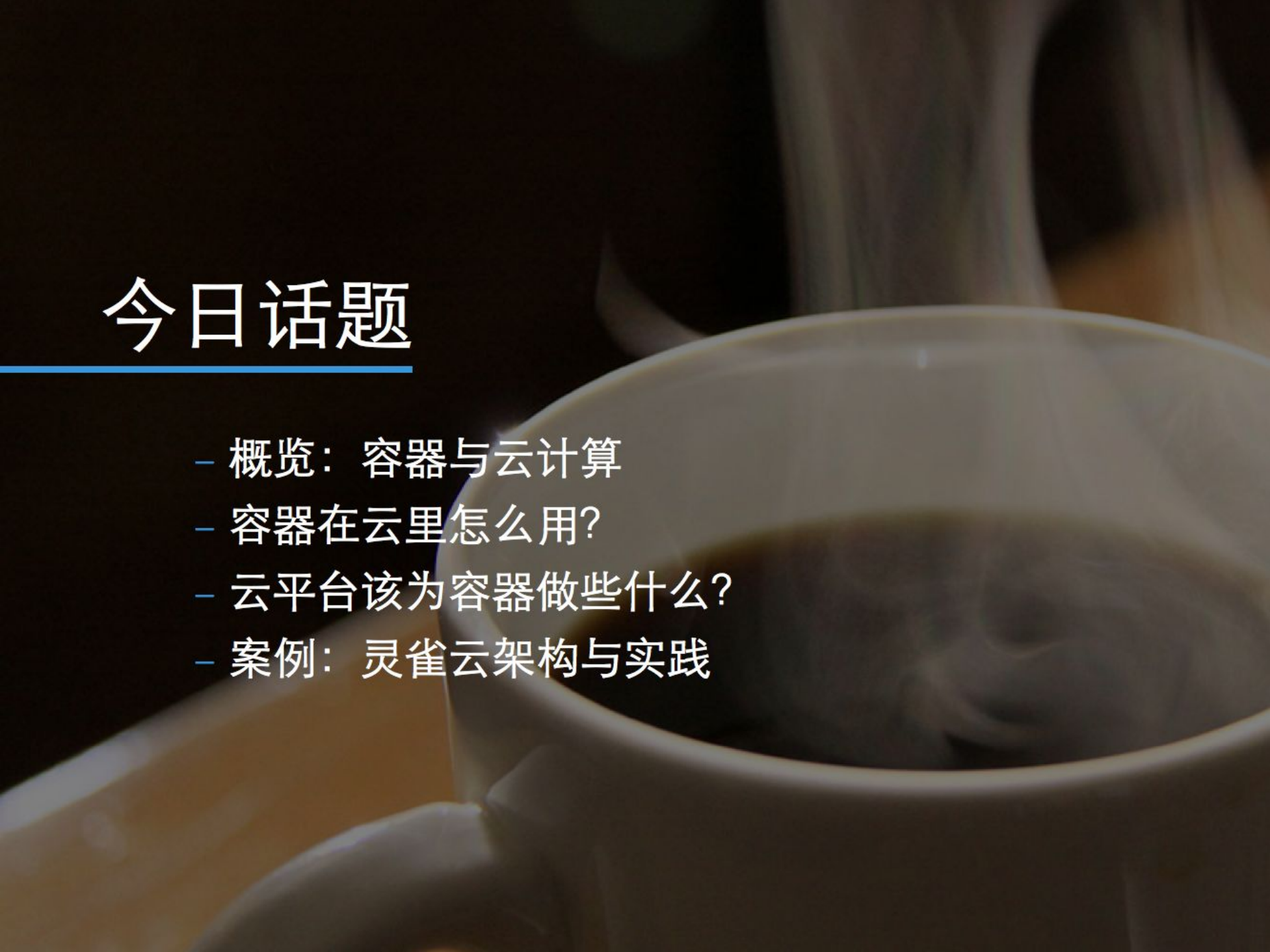
International Software Development Conference



QCon
全球软件开发大会

深解容器云平台：趋势，架构与实践

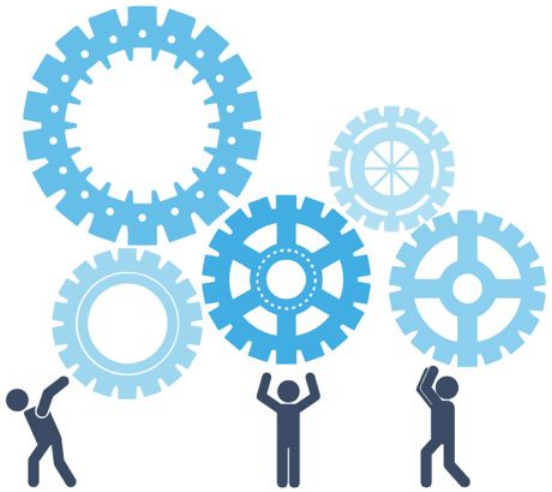
陈恺，灵雀云CTO



今日话题

- 概览：容器与云计算
- 容器在云里怎么用?
- 云平台该为容器做些什么?
- 案例：灵雀云架构与实践

先容器时代云计算状况



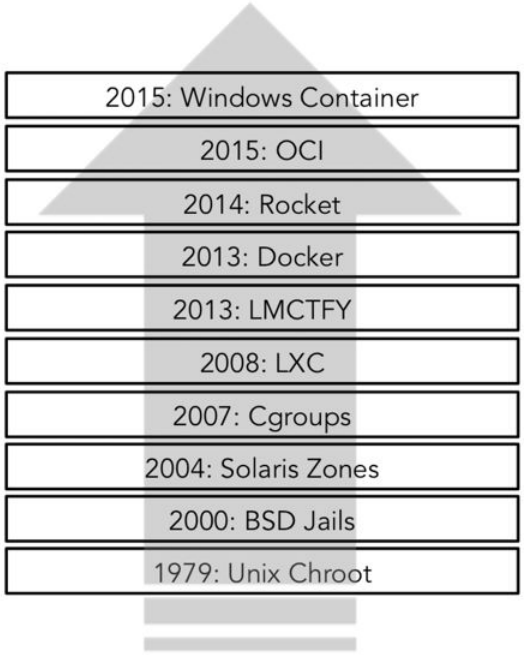
Infrastructure-as-a-Service

没有将开发者从繁重的运维工作中解放出来

Platform-as-a-Service

对应用限制过多，无法为开发者广泛接受

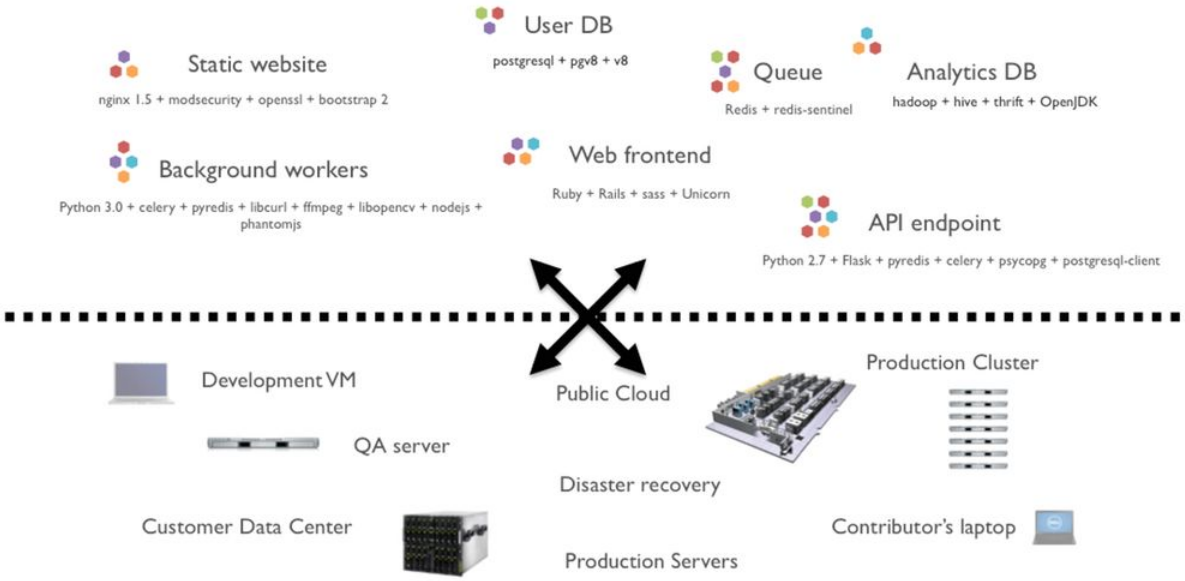
容器



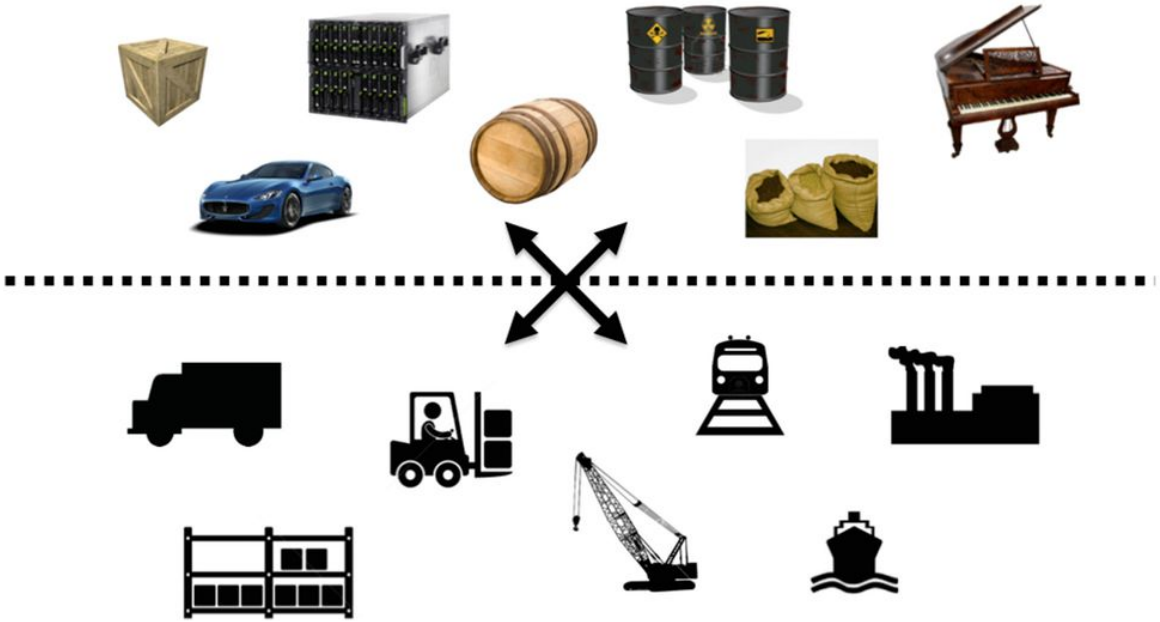
“容器简史”

- 操作系统虚拟化
- 命名空间
- 资源隔离
- 轻量级（启动、迁移、扩展）

应用交付面临的挑战

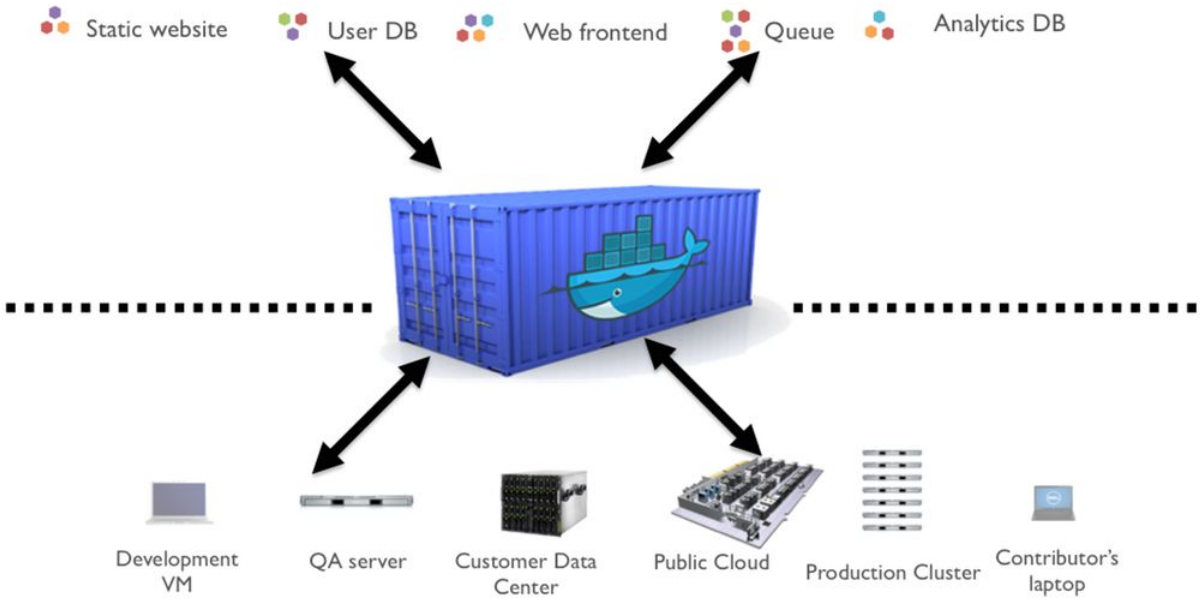

User DB
postgresql + pgv8 + v8
Static website
nginx 1.5 + modsecurity + openssl + bootstrap 2
Queue
Redis + redis-sentinel
Analytics DB
hadoop + hive + thrift + OpenJDK
Background workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs
Web frontend
Ruby + Rails + sass + Unicorn
API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client
Development VM
Public Cloud
Production Cluster
QA server
Disaster recovery
Customer Data Center
Contributor's laptop
Production Servers

这个情形似曾相识

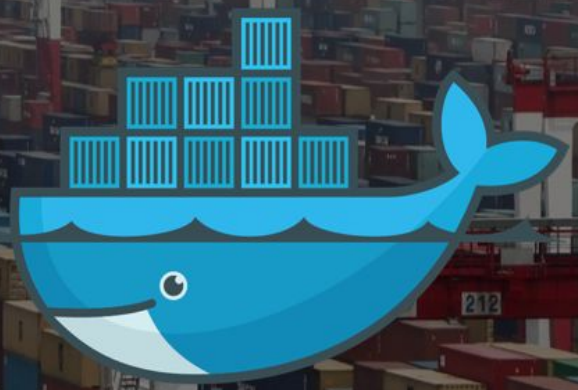


解决方案：集装箱

Docker：应用交付的“集装箱”



从容器到“集装箱”



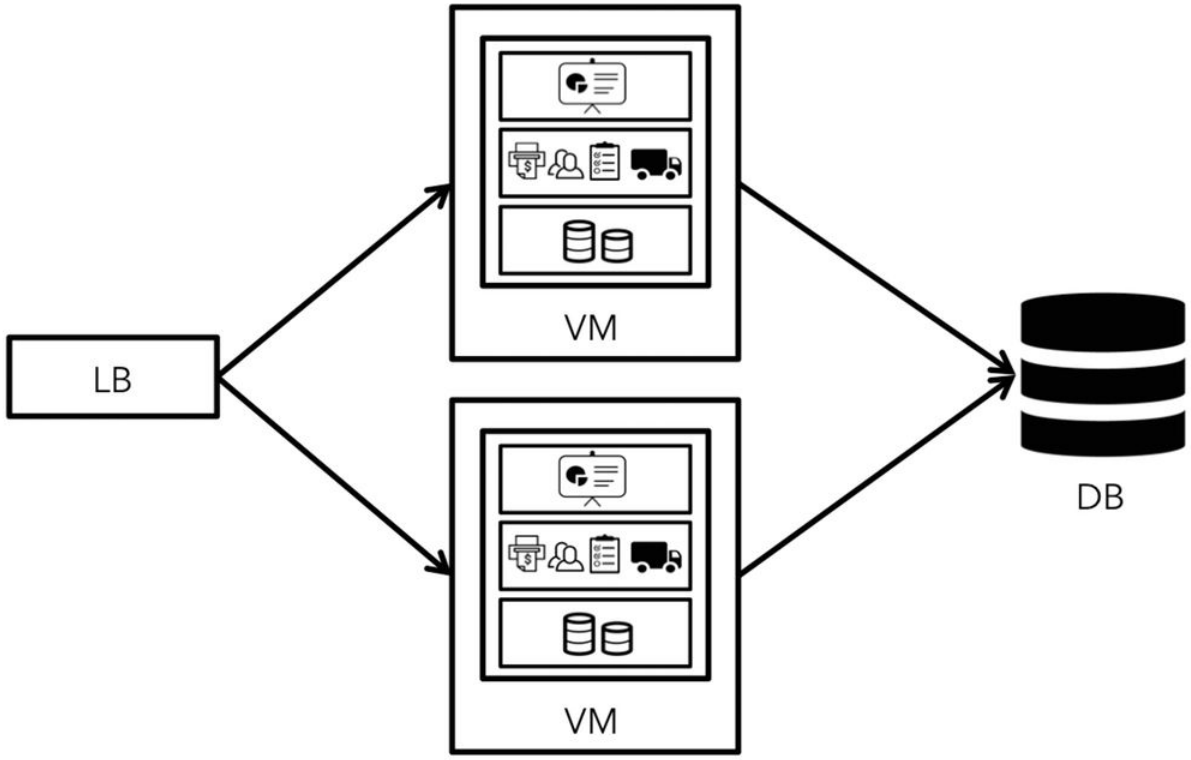
docker

- 容器 + 镜像
- 体验良好的工具
- 充满活力的社区 + 繁荣的生态圈
- 全新应用交付标准

容器在云里怎么用?

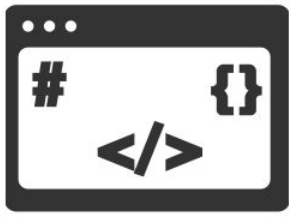
以一个互联网应用为例

云环境部署架构



应用开发周期

架构设计
Multi-layered architecture



开发
Python + Django

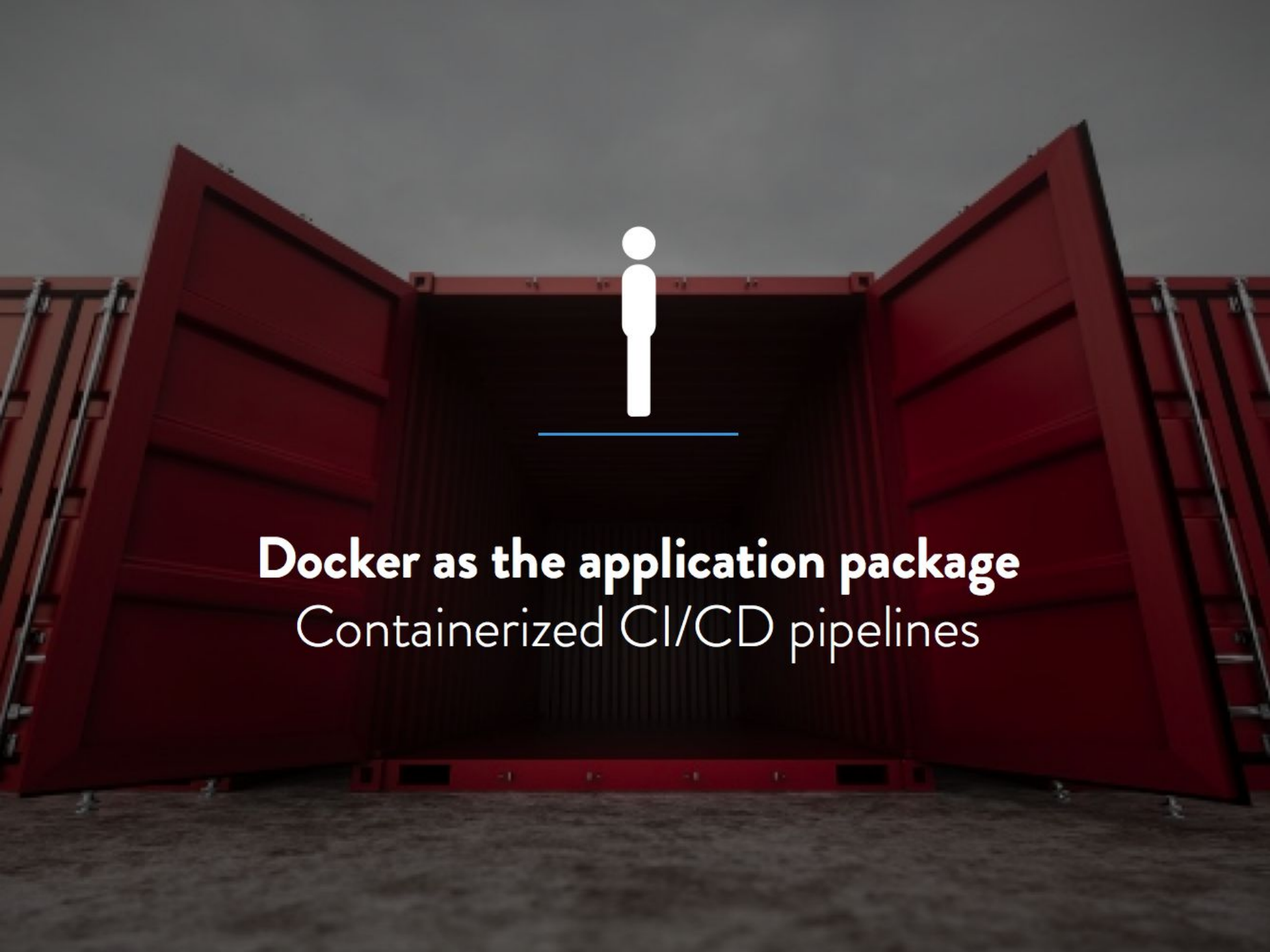
构建
Jenkins

应用开发周期

测试
UT + Selenium

部署
Ansible

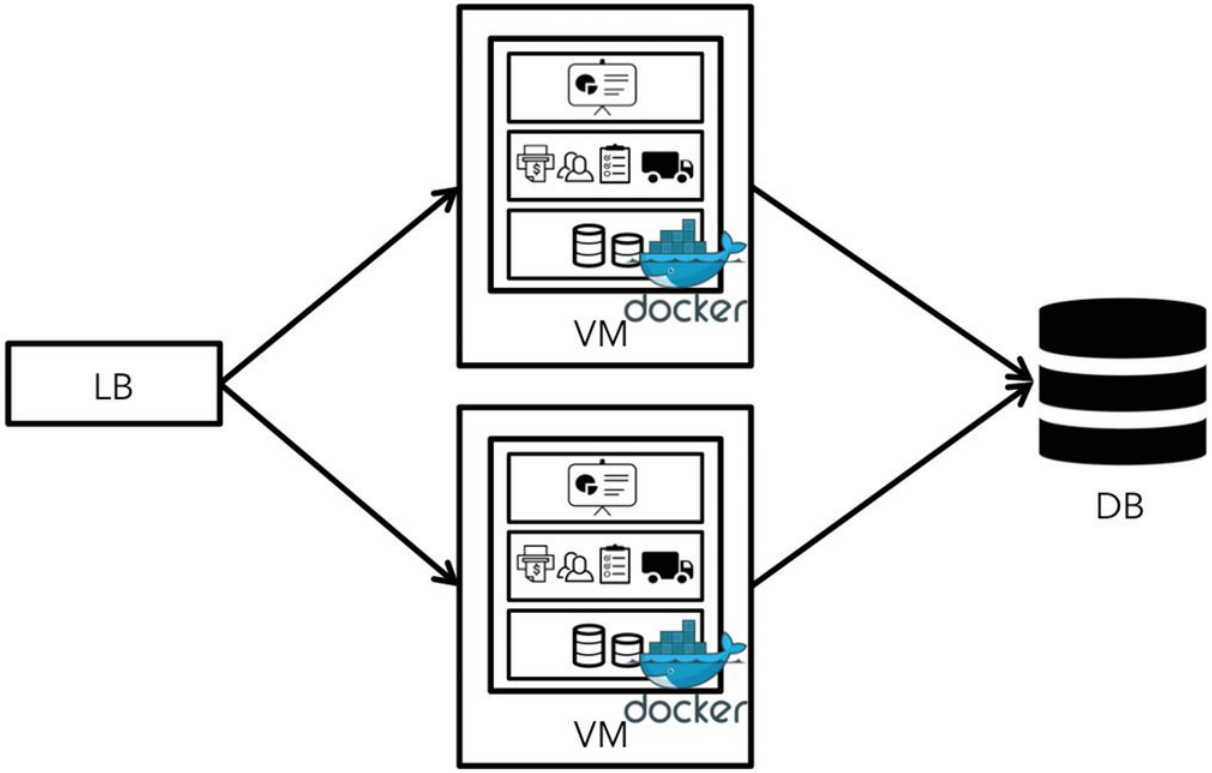
运维
Virtualized infrastructure + app



Docker as the application package

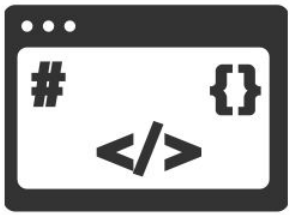
Containerized CI/CD pipelines

容器化服务部署架构



应用开发周期 – 容器化

架构设计
No impact



开发
+Dockerfile

构建
$ docker build

应用开发周期 – 容器化

测试
Consistent environments

部署
Repeatable deployments

运维
Immutable infrastructure

“单块”架构

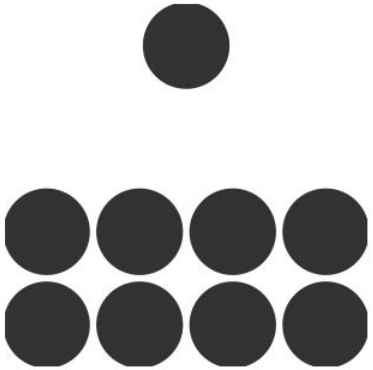
- 在物理部署层面是一个“单块”
- 以一个整体编译、打包、部署、运维
- E.g. Java WAR, Rails or Node.js directory

微服务架构

“In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.”

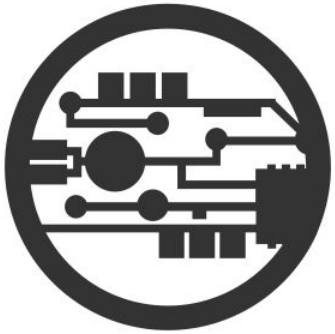
Martin Fowler

微服务架构优势



Independence

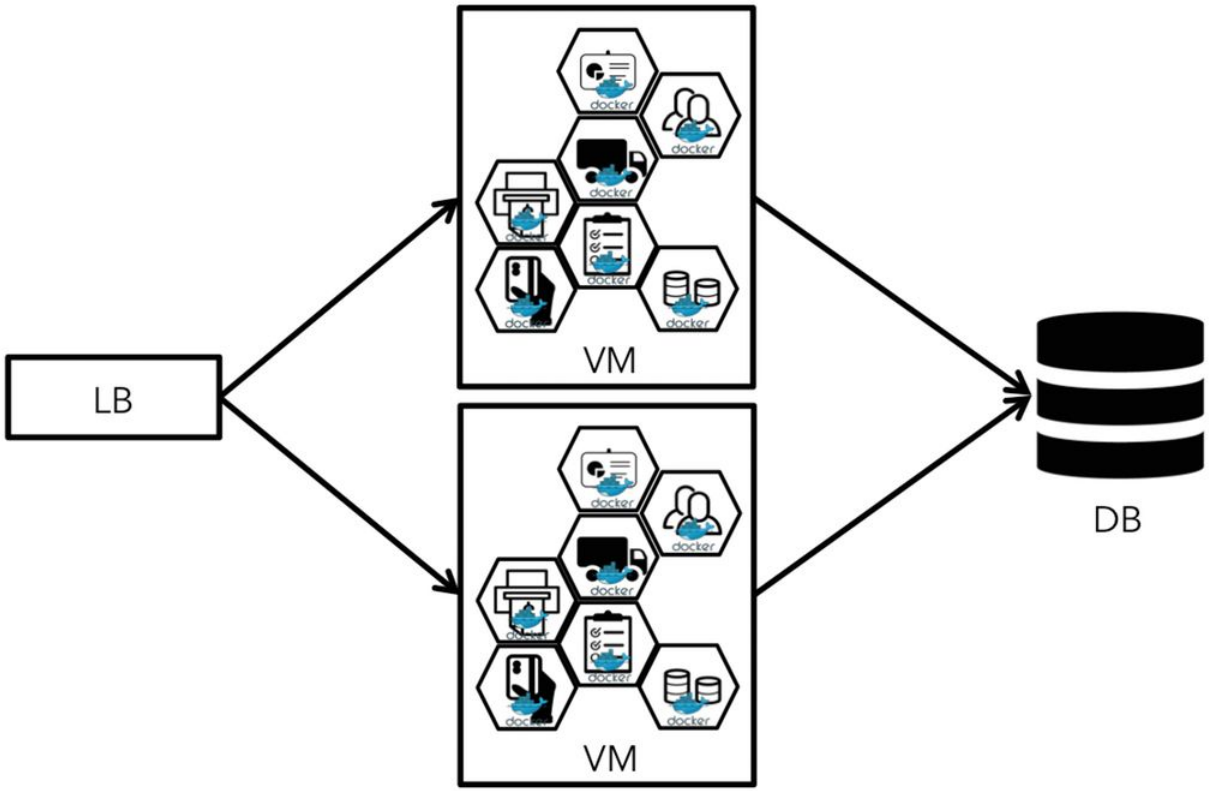
Agility



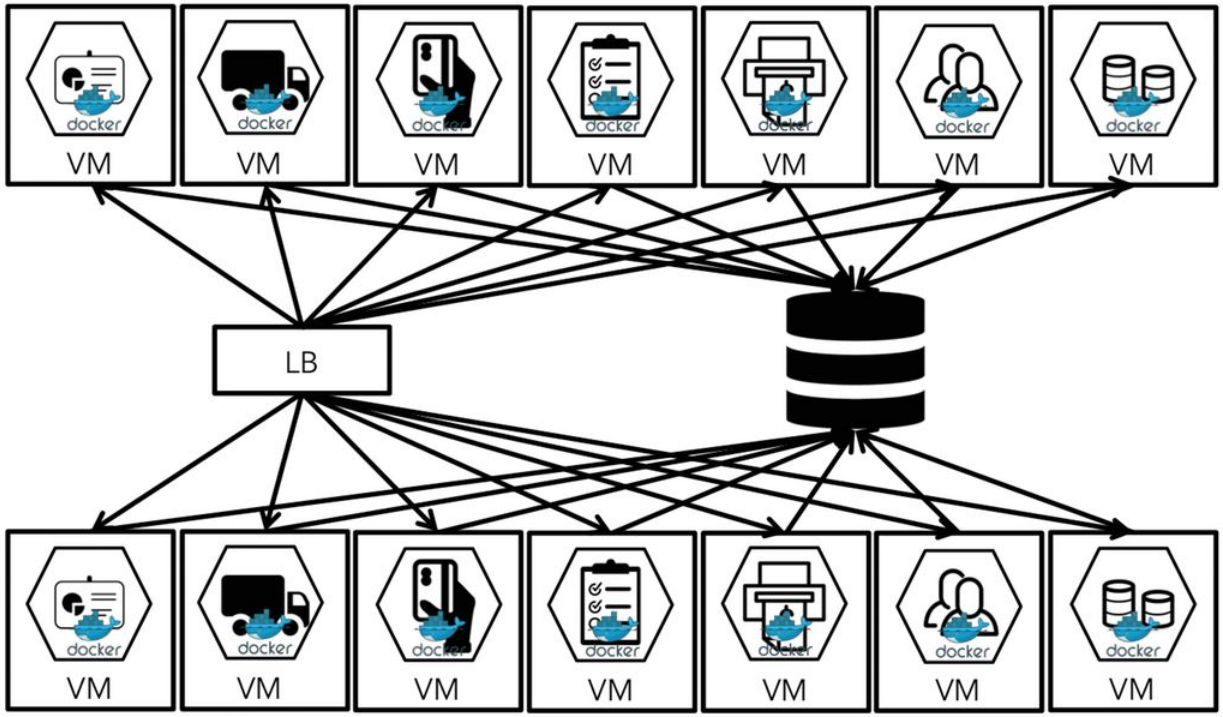
Technology Adoption

微服务架构下的应用

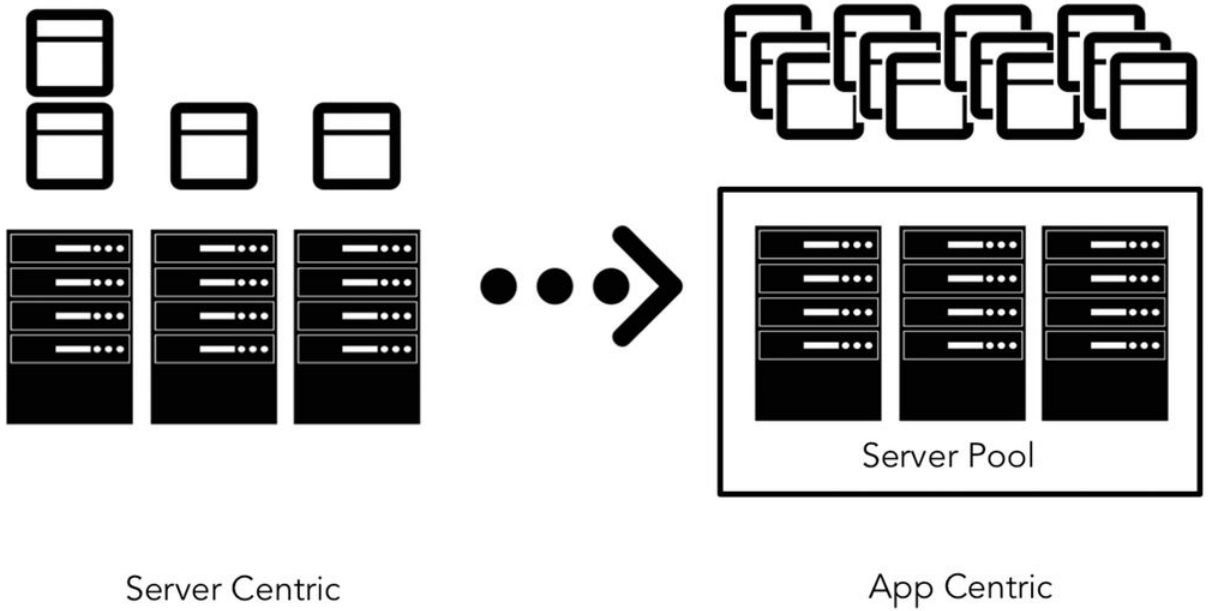
微服务如何部署?



或者?



“应用为本”理念



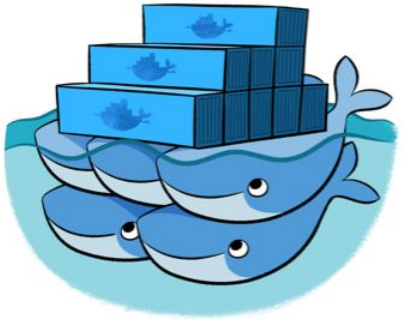
Container Orchestration

Distributed applications made simpler

常用解决方案

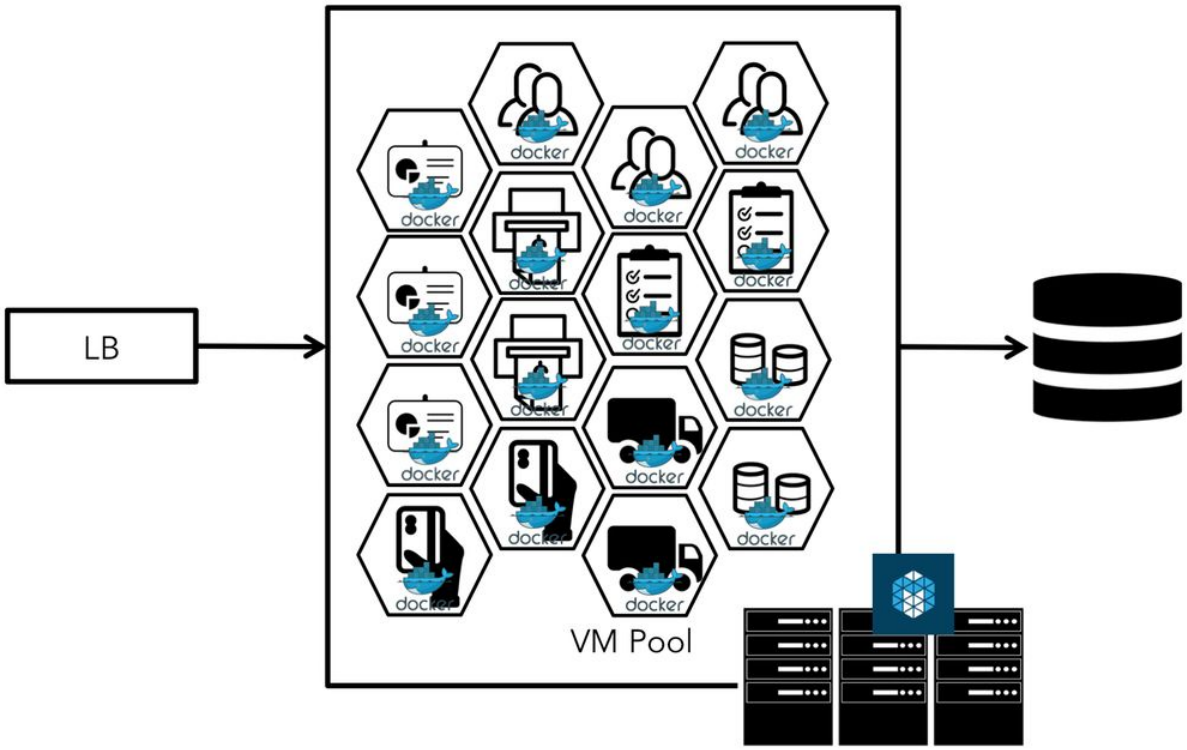
Apache Mesos

Kubernetes



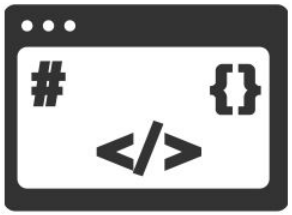
Docker Orchestration

容器化微服务部署架构



应用开发周期 – 容器化微服务

架构设计
Microservices



开发
Agile, polyglot

构建
Each service independently packaged as a Docker image

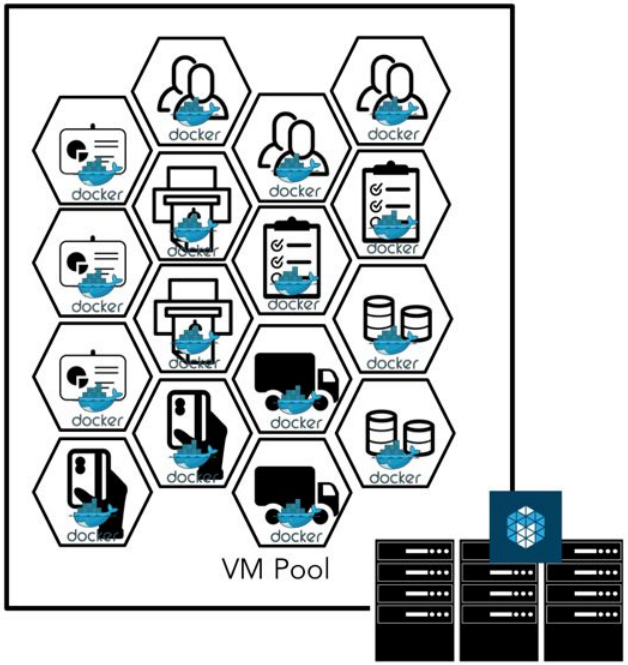
应用开发周期 – 容器化微服务

测试
Independently testable

部署
Independently deployable

运维
Virtual infrastructure + orchestration system + app

新的问题



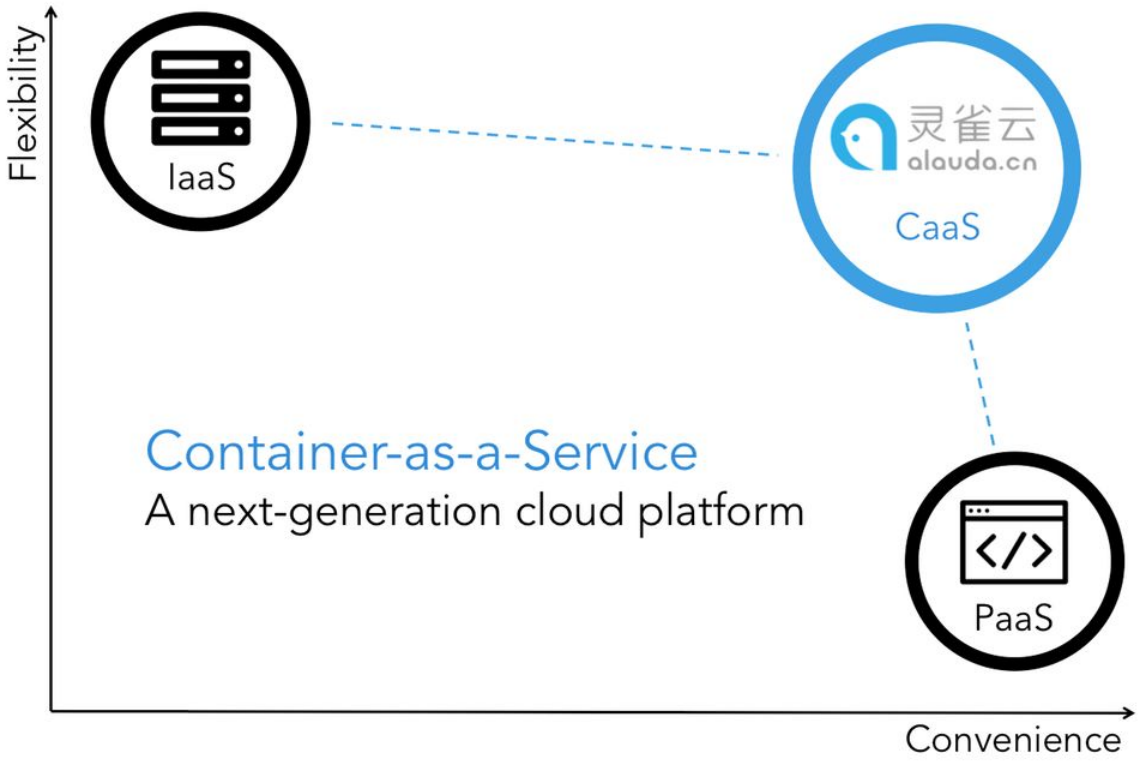
- 服务发现?
- 跨主机连接?
- 负载均衡?
- 存储?
- 集群管理系统运维?

云平台该为容器做些什么?

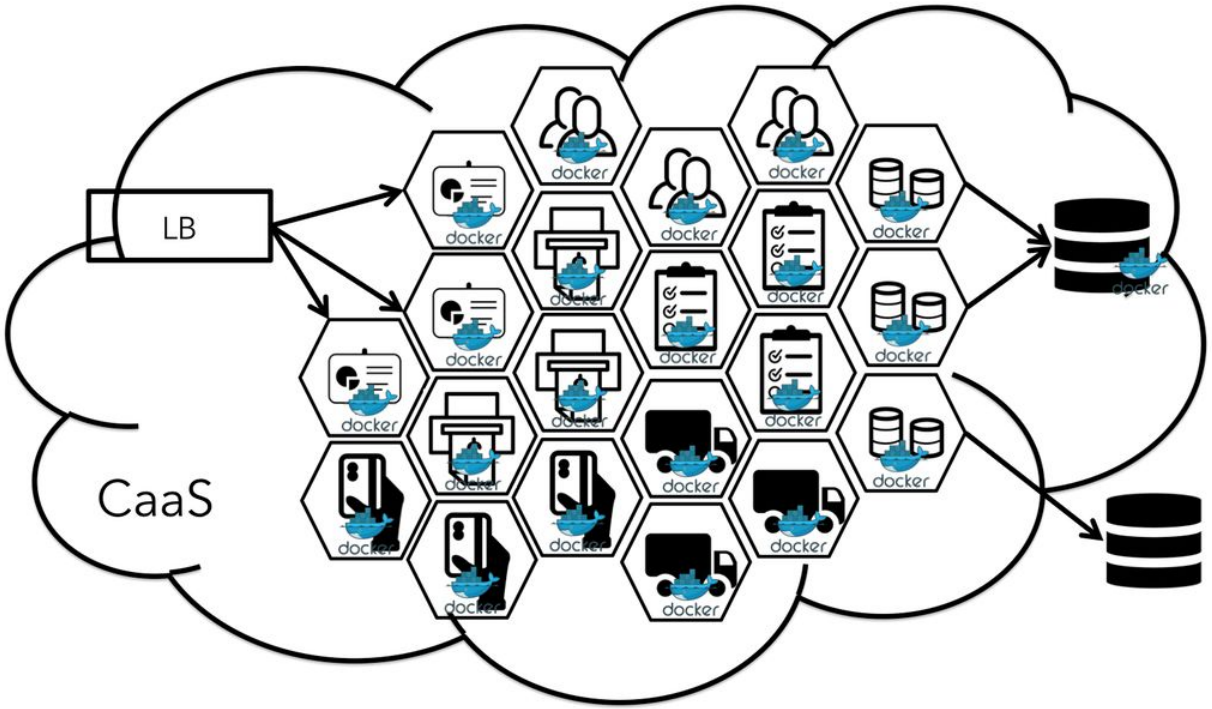
Container-as-a-Service

A platform to manage it all

容器云平台

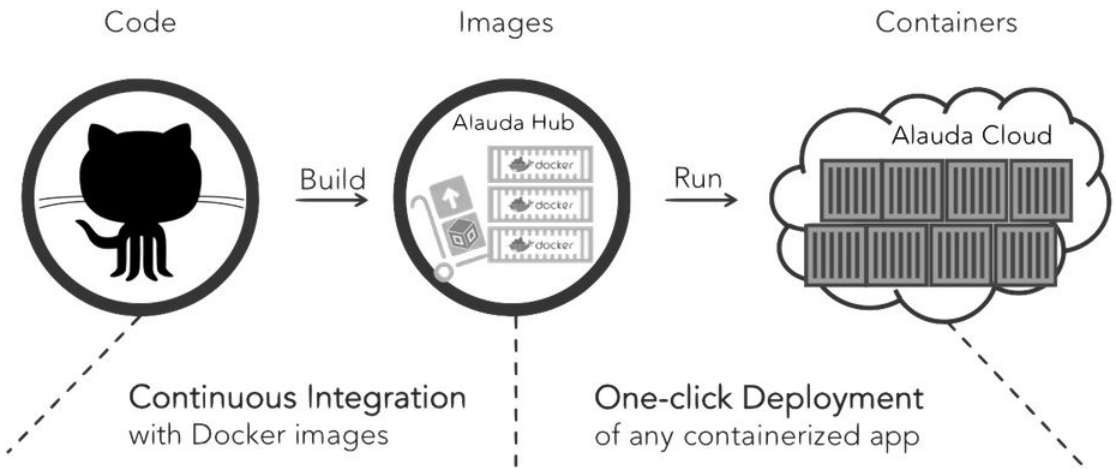

Flexibility
IaaS
灵雀云
alauda.cn
CaaS
Container-as-a-Service
A next-generation cloud platform
PaaS
Convenience

容器云环境部署架构

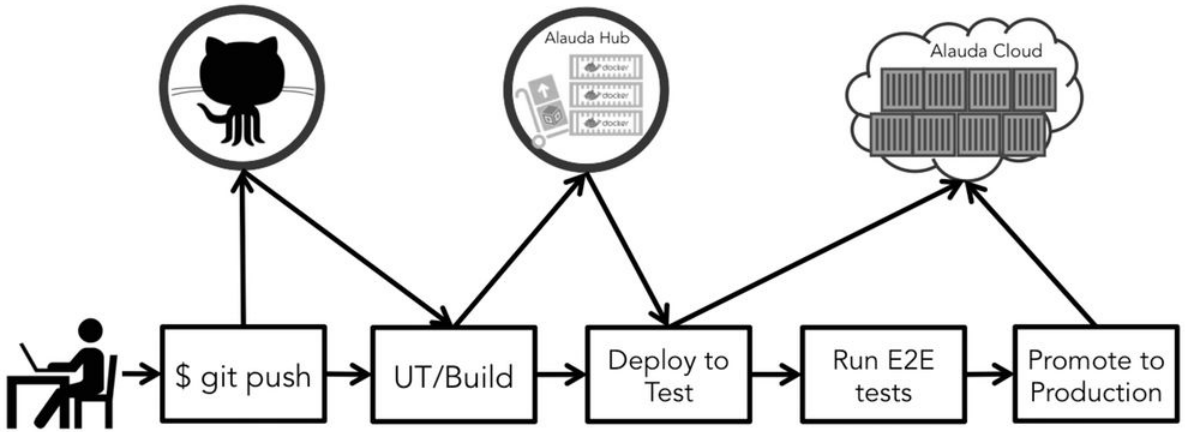


灵雀云架构与实践

灵雀云容器云平台

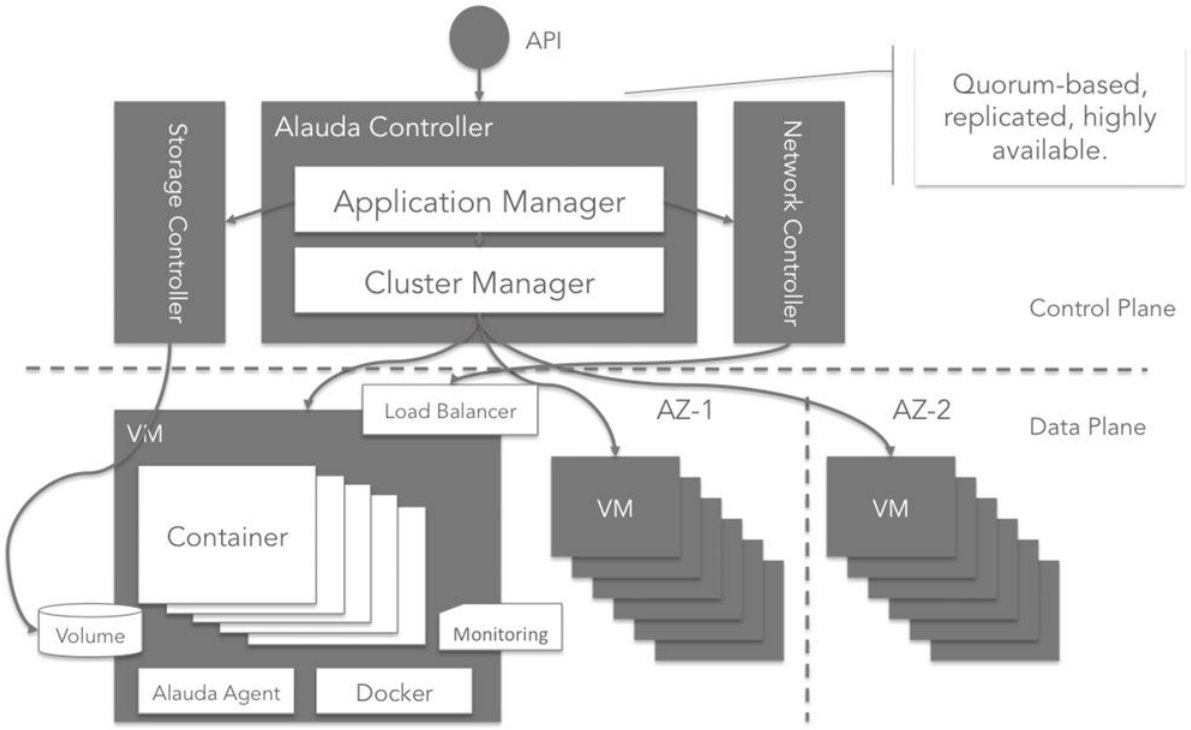

Code
Images
Containers
Build
Alauda Hub
Run
Alauda Cloud
Continuous Integration
with Docker images
One-click Deployment
of any containerized app

应用开发周期 – 灵雀云



Continuous Integration/Continuous Delivery

灵雀云集群架构


API
Alauda Controller
Application Manager
Cluster Manager
Storage Controller
Network Controller
Quorum-based, replicated, highly available.
Control Plane
Data Plane
Load Balancer
AZ-1
AZ-2
VM
Container
Volume
Monitoring
Alauda Agent
Docker
VM
VM

灵雀云跨区跨云架构

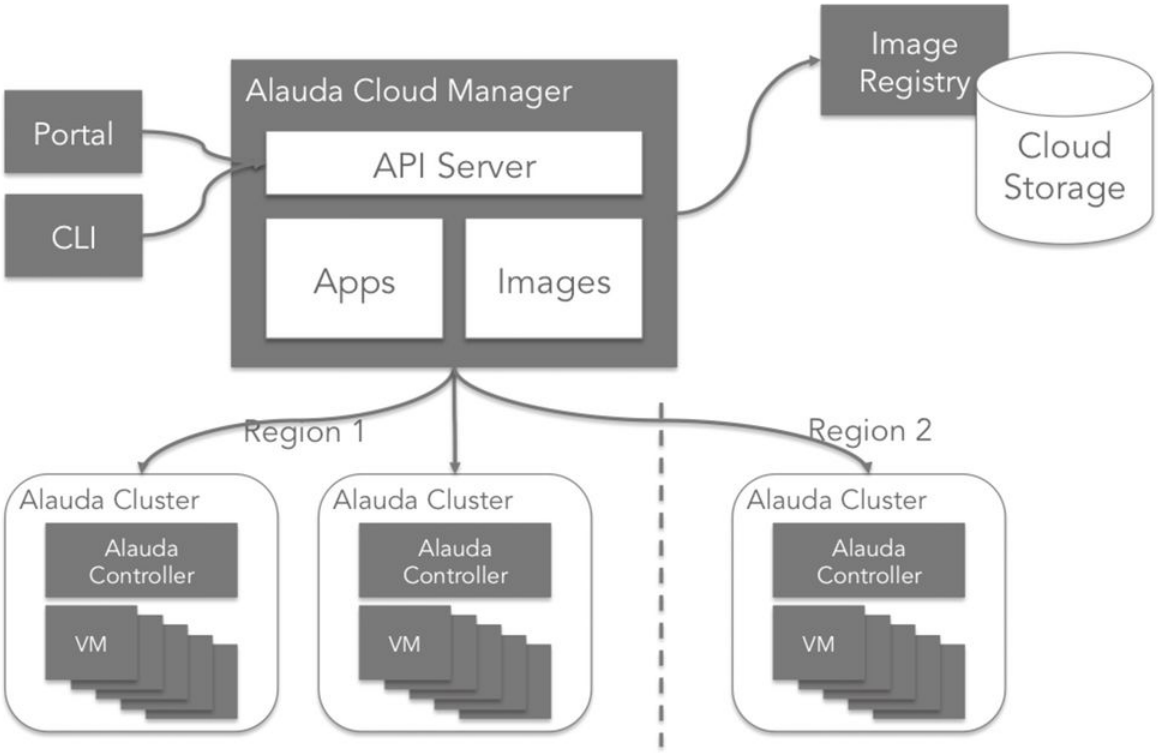
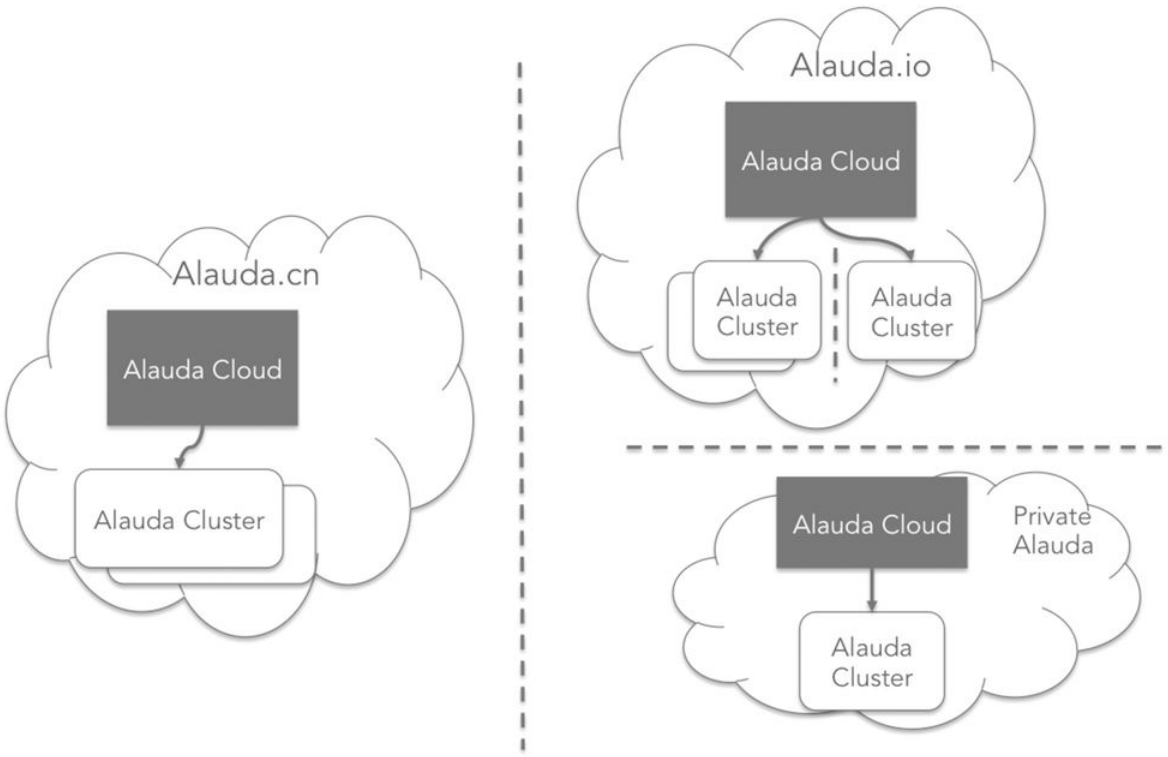
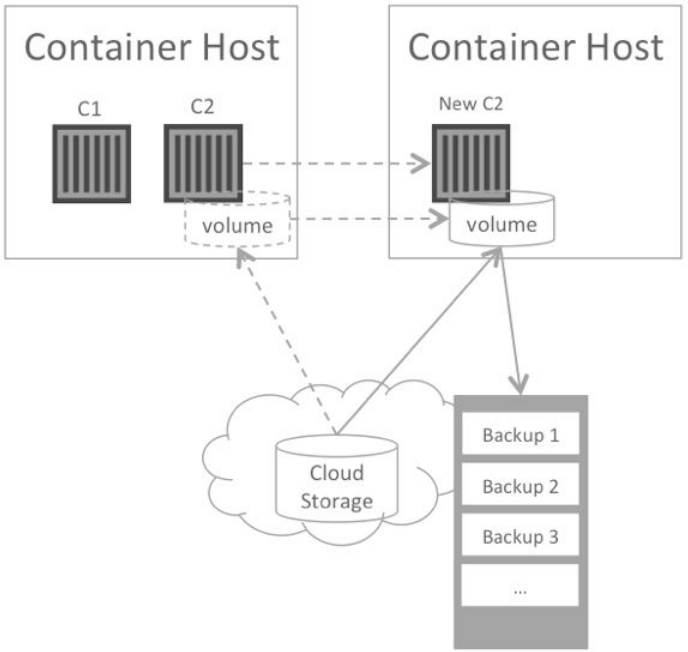


Image Registry
Cloud Storage
Alauda Cloud Manager
Portal
CLI
API Server
Apps
Images
Region 1
Region 2
Alauda Cluster
Alauda Controller
VM
Alauda Cluster
Alauda Controller
VM
Alauda Cluster
Alauda Controller
VM

灵雀云平台架构

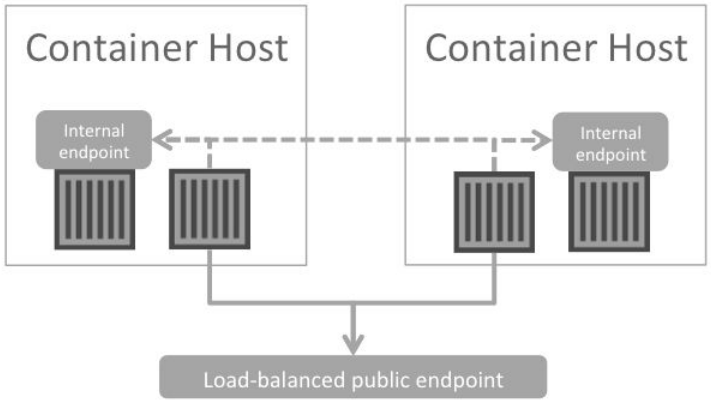

Alauda.cn
Alauda Cloud
Alauda Cluster
Alauda.io
Alauda Cloud
Alauda Cluster
Alauda Cluster
Alauda Cloud
Private Alauda
Alauda Cluster

灵雀云存储



- 持久性存储卷
- 重启、迁移时自动挂载
- 存储卷备份
- 支持有状态服务

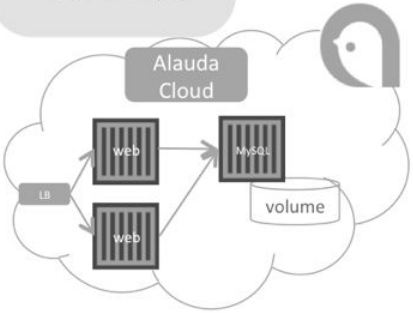
灵雀云网络



- 跨主机关联
- 四层、七层负载均衡
- 服务发现
- 自定义域名
- 应用内网
- 专属IP

灵雀云服务编排

```
web:
  image: kaichen/web
  ports:
   - “80”
  links:
   - mysql
mysql:
  image: mysql
  expose:
   - “3306”
  volumes:
   - /var/lib/mysql
```




```
$ alauda compose up
```

- 与docker-compose兼容
- 自由集成任意镜像
- 积木式搭建应用
- 一键部署管理完整应用

灵雀云镜像中心

- 实时同步官方镜像
- 公有、私有镜像仓库
- 镜像管理
- 镜像市场与社区

Thank You!

http://www.alauda.cn
kaichen@mathildetech.com

THANKS