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

从Docker到容器服务

Docker 云端实践之路

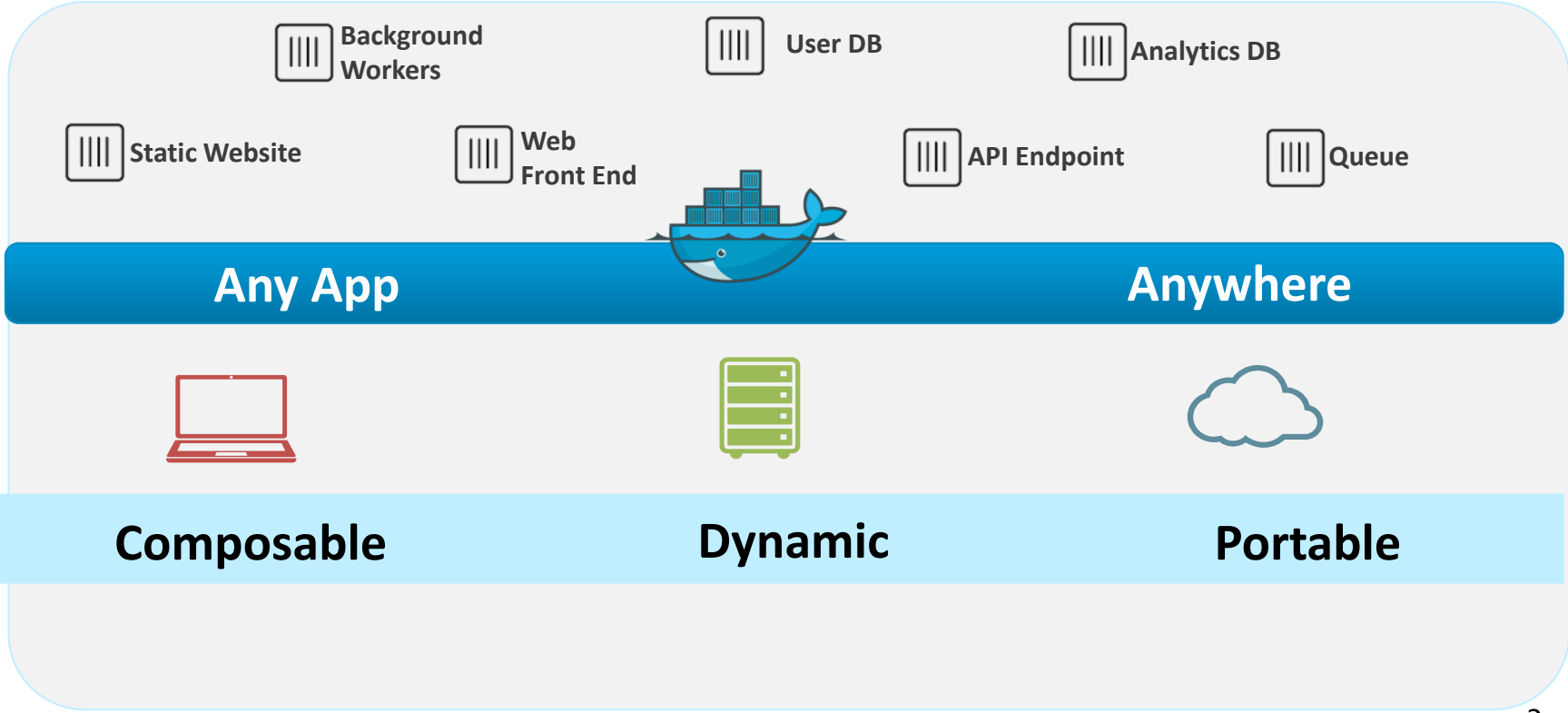
阿里云 易立

Alibaba Group 阿里巴巴集团 阿里云 aliyun.com

日程

- Docker编排技术概述
- 容器即服务(Container as a Service)
  - 微服务支持
  - DevOps
- 未来发展趋势

Docker – 标准化的构建、交付、运维手段



为什么Docker这么火？

Agility

Portability

Control

Innovation at speed

Frictionless movement

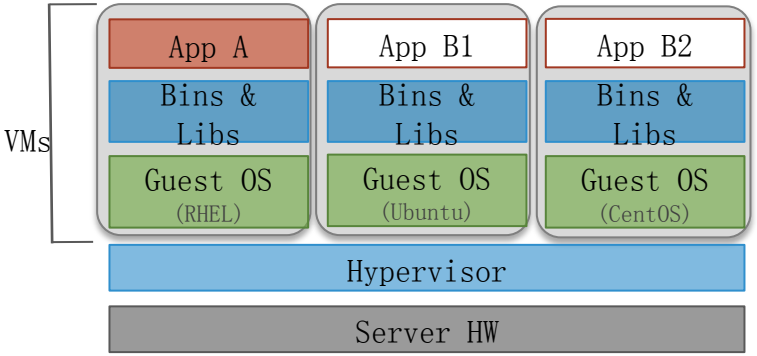
Manage and secure at scale

敏捷：秒级应用启动、轻量级隔离、细粒度资源控制、低性能损耗
可移植性： 环境无关的交付、部署方式；可用于软件生命周期中不同运行环境
可控：标准化推动自动化，提高运维效率和规模；隔离性提升应用安全性；版本管理可追溯

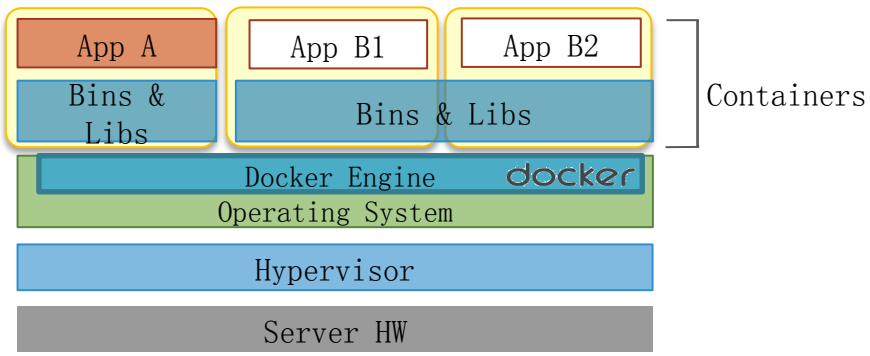
“相爱”或“相杀”：Docker与虚拟化技术

- Docker是一种轻量级的操作系统虚拟化方案
- Docker容器和虚拟化技术 - 互补、双赢
  - 利用虚拟机提供弹性基础架构，更好的安全隔离，动态热迁移
  - 利用容器技术简化迁云之路，实现无边界的云计算

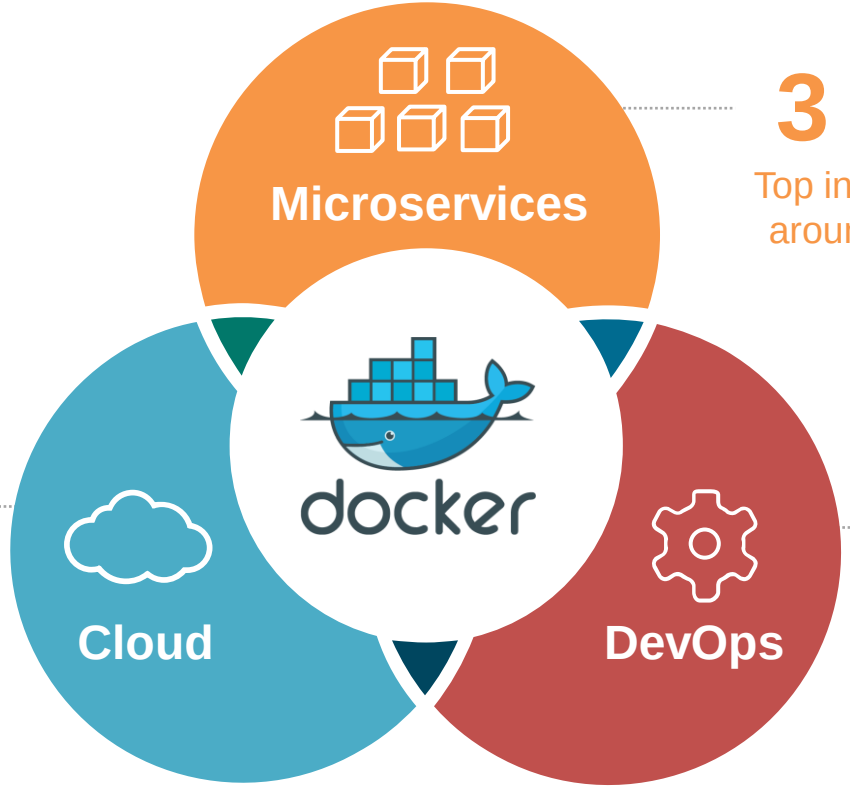
经典虚拟化方式



Docker容器方式



IT转型： Docker 应“云”而生



3 out 4
Top initiatives revolve around applications

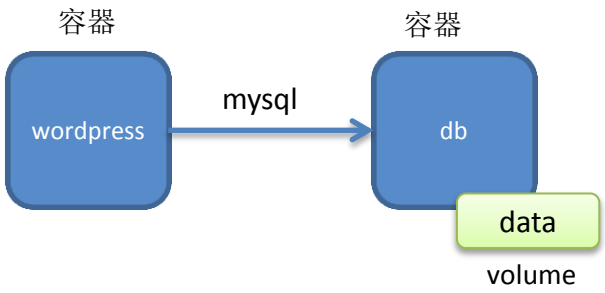
80%
Docker is central to cloud strategy

44%
Looking to adopt DevOps

State of App development Survey: Q1 - 2016

容器编排 - Docker Compose

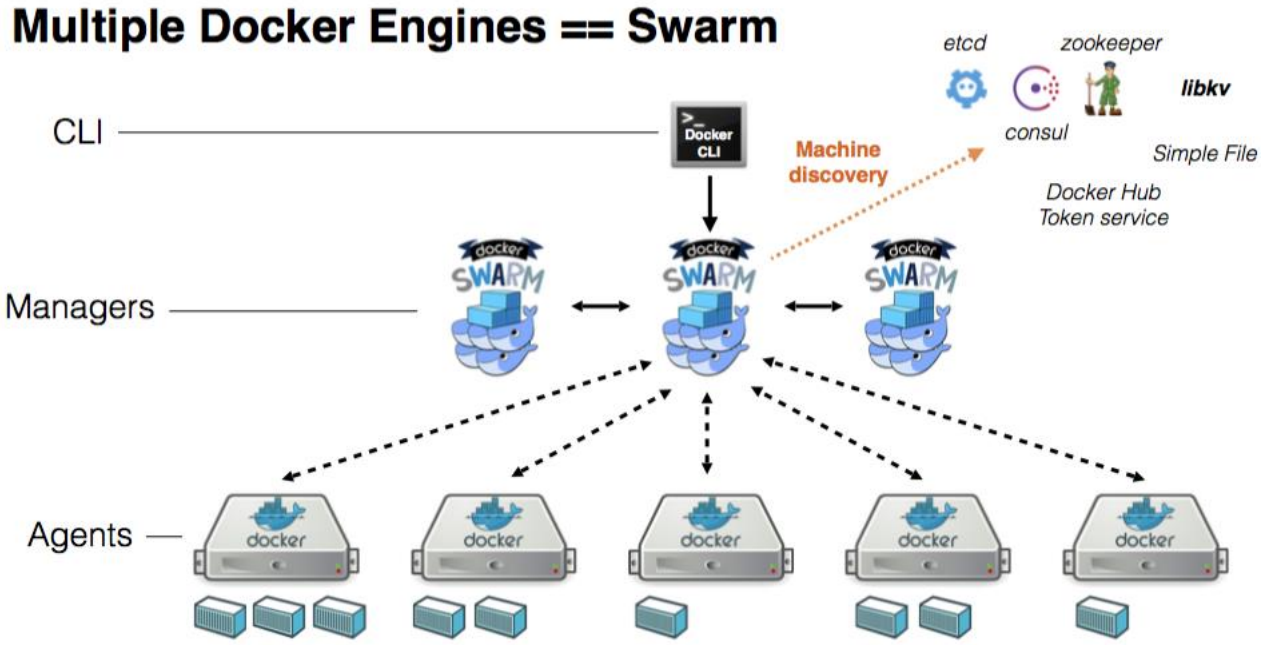
```
version: '2'
services:
    wordpress:
      image: wordpress:4
      ports:
        - 80
      restart: always
      links:
        - db:mysql
      network_mode: bridge
    db:
      image: mysql:5.7
      environment:
        MYSQL_ROOT_PASSWORD: password
      restart: always
      volumes:
        - data:/var/lib/mysql
      network_mode: bridge
volumes:
   data:
      driver: local
```



一键部署：docker-compose up
手动伸缩：docker-compose scale wordpress=3

- 优点
  - 简单好用，便于开发
    - 镜像开发
    - 本地环境沙箱：开发、UT
  - 编排容器、存储和网络
- 不足
  - 面向开发和部署，不支持自动化运维

容器集群管理 - Docker Swarm



优点

- 兼容标准的 Docker API
- 灵活、可插拔的容器调度

不足

- 面向容器、缺少微服务支持

生产环境中使用Docker

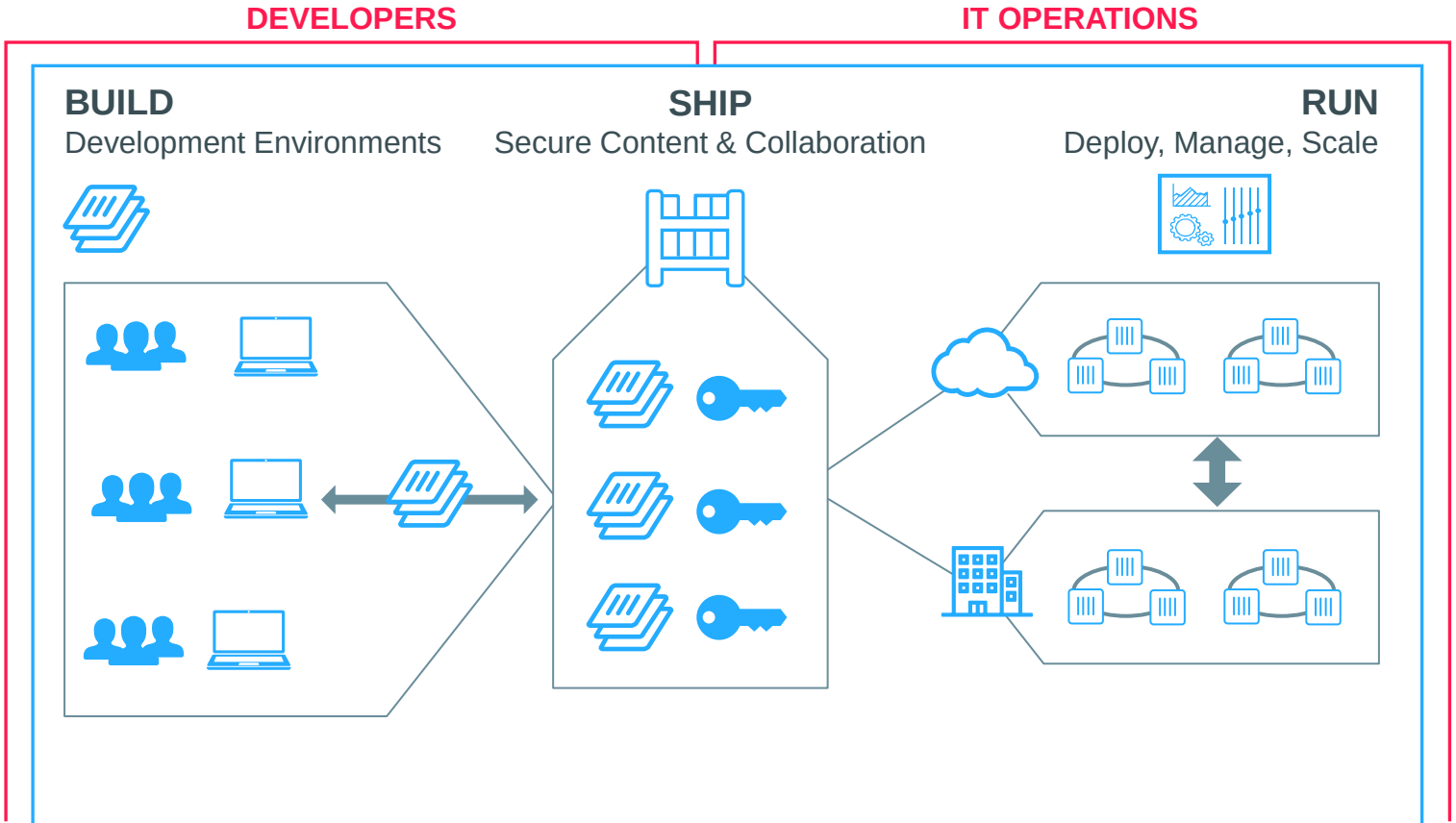
您还需要:
- 集群管理
- 安全
- 网络
- 存储
- 调度
- 编排
- …

图片来源 http://www.popularmechanics.com/home/how-to/g172/shipping-container-homes-460309/

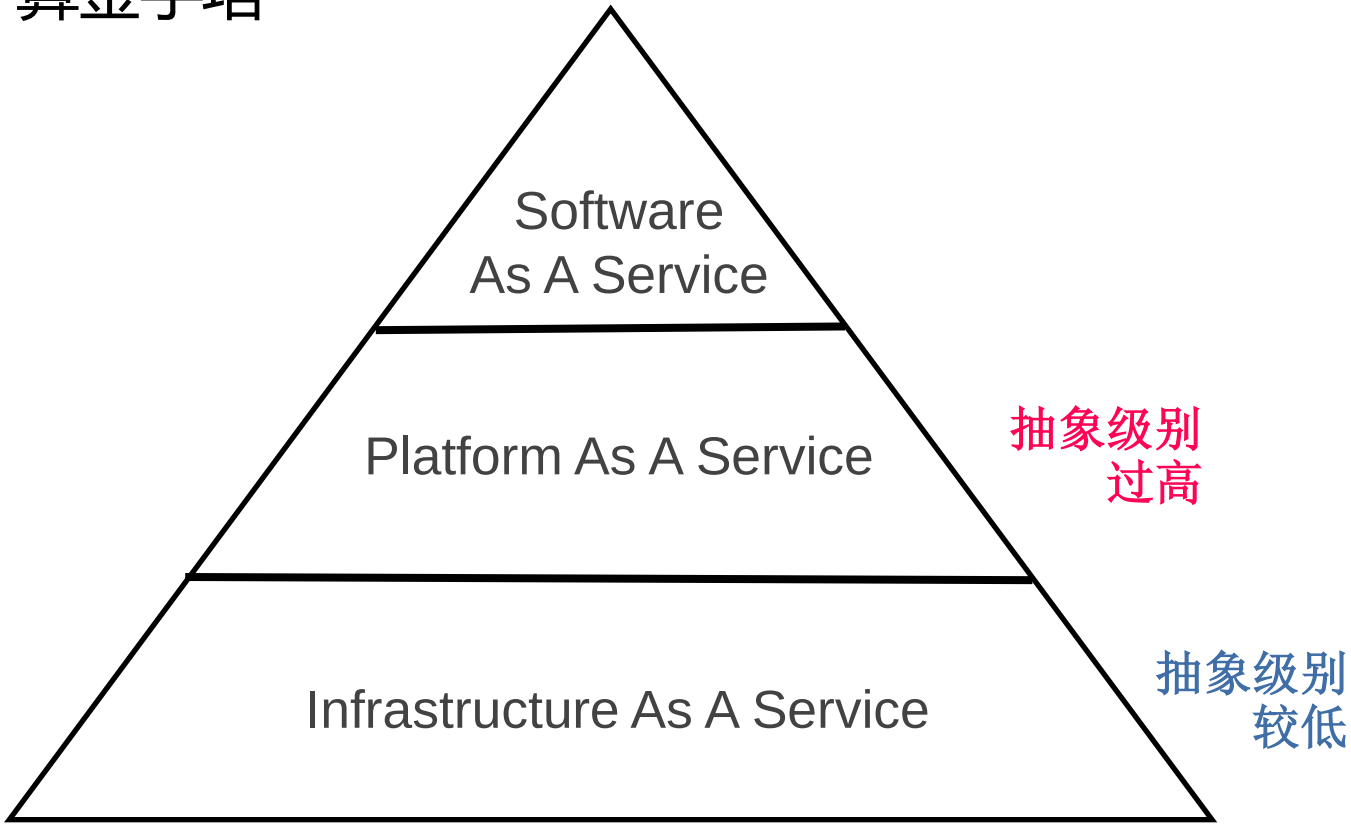
日程

- Docker编排技术概述
- 容器即服务(Container as a Service)
  - 微服务支持
  - DevOps
- 未来发展趋势

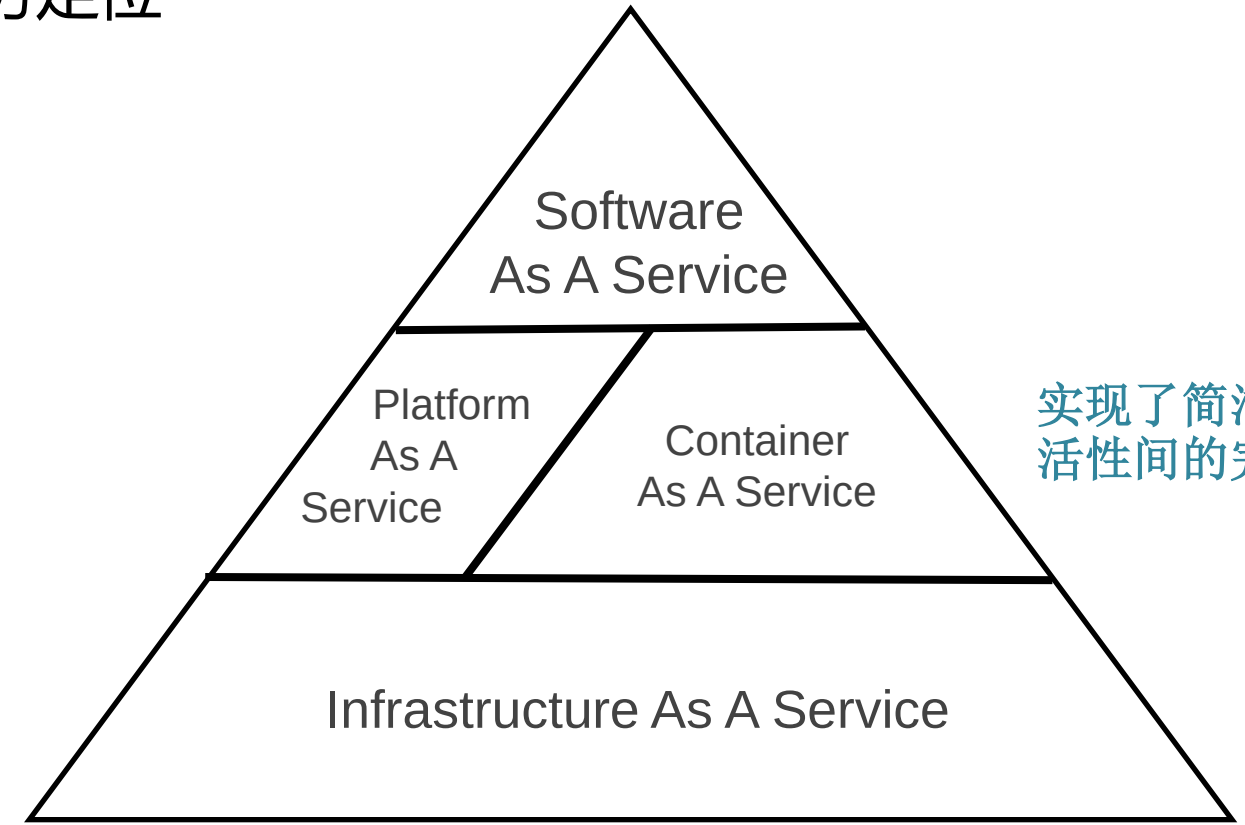
Containers as a Service (CaaS 容器服务)



传统云计算金字塔


Software
As A Service
Platform As A Service
Infrastructure As A Service
抽象级别
过高
抽象级别
较低

容器服务定位



实现了简洁性和灵活性间的完美平衡

容器即服务 Container as a Service

等等

Docker Cloud
(tutum.co)

Docker Swarm API
Compose template

Amazon EC2
Container Service

ECS API
Compose
template/Task
definition

Google
Container Engine

Kubernetes API
Pod/Service

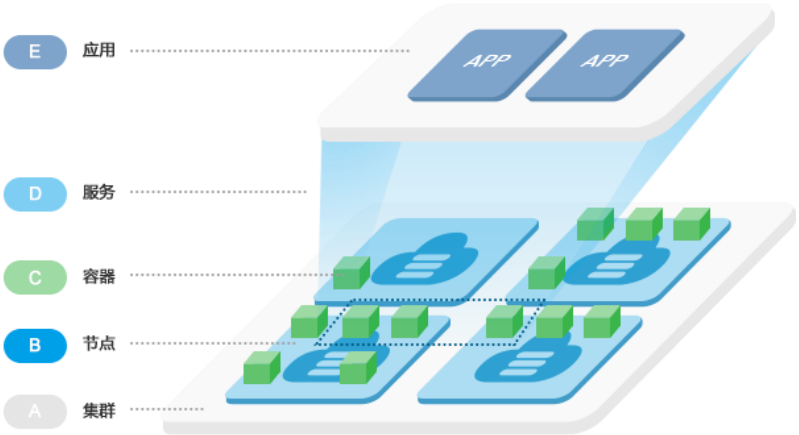
阿里云
容器服务

Docker Swarm API
Compose template

阿里云容器服务概念模型

- 资源层面
  - 集群
  - 节点
- 内容层面
  - Compose模板
  - 镜像
- 应用层面
  - 应用
  - 服务
  - 容器

Docker Compose 模板



演示：容器服务和Docker镜像仓库

从Docker镜像仓库发现镜像
利用Docker镜像创建Tomcat集群

演示：一键开通WordPress博客

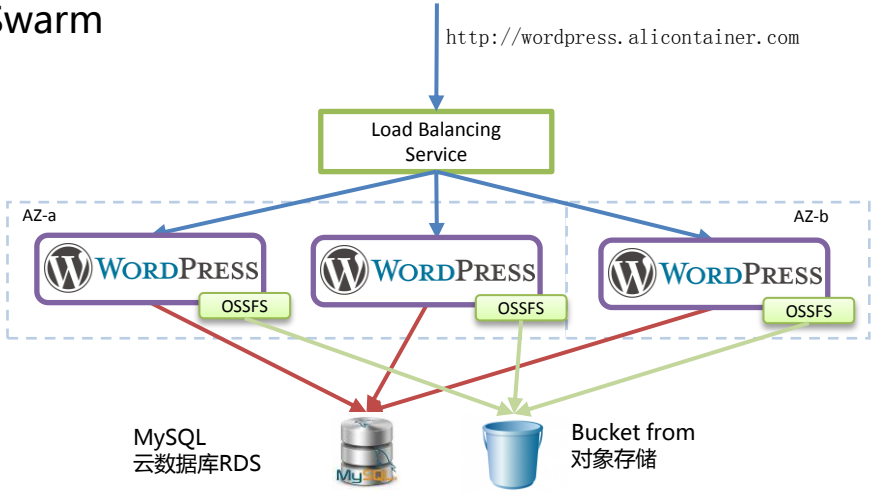
利用Compose模板一键部署WordPress博客

集成容器和云服务

```
version: '2'
services:
  wordpress:
    image: wordpress:4.5
    restart: always
    links:
      - 'db:mysql'
    volumes:
      - 'wp_upload:/var/www/html/wp-content/uploads'
    environment:
      - WORDPRESS_DB_USER=blog
      - WORDPRESS_DB_PASSWORD=xxxxxx
      - WORDPRESS_DB_NAME=wordpress
      - availability:az==2
    labels:
      aliyun.probe.url: http://container
      aliyun.routing.port_80: http://wordpress
      aliyun.scale: '3'
      aliyun.log_store_wordpress: stdout
  db:
    external:
      host: rdsxxxxxx.mysql.rds.aliyuncs.com
      ports:
       - 3306
volumes:
  wp_upload:
    driver: ossfs
    driver_opts:
      bucket: acs-sample-wordpress
```

- 完全兼容 Docker Compose/Swarm

一键部署到云上

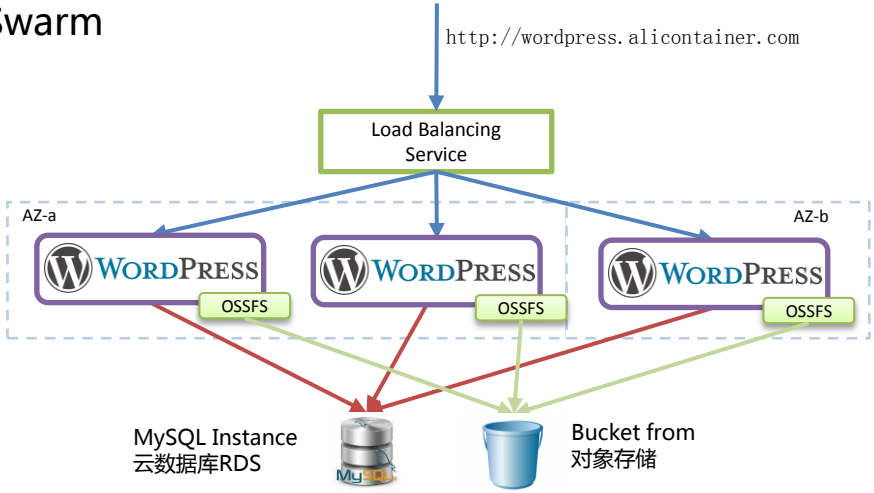


集成容器和云服务

```
version: '2'
services:
  wordpress:
    image: wordpress:4.5
    restart: always
    links:
      - 'db:mysql'
    volumes:
      - 'wp_upload:/var/www/html/wp-content/uploads'
    environment:
      - WORDPRESS_DB_USER=blog
      - WORDPRESS_DB_PASSWORD=xxxxxx
      - WORDPRESS_DB_NAME=wordpress
      - availability:az==2
    labels:
      aliyun.probe.url: http://container
      aliyun.routing.port_80: http://wordpress
      aliyun.scale: '3'
      aliyun.log_store_wordpress: stdout
  db:
    external:
      host: rdsxxxxxx.mysql.rds.aliyuncs.com
      ports:
       - 3306
volumes:
  wp_upload:
    driver: ossfs
    driver_opts:
      bucket: acs-sample-wordpress
```

- 完全兼容 Docker Compose/Swarm
- 声明的方式支持容器及云服务

一键部署到云上

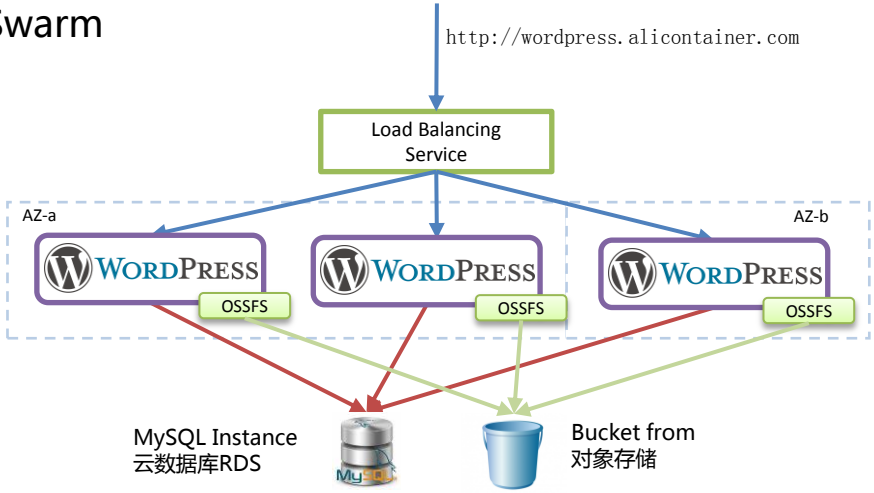


集成容器和云服务

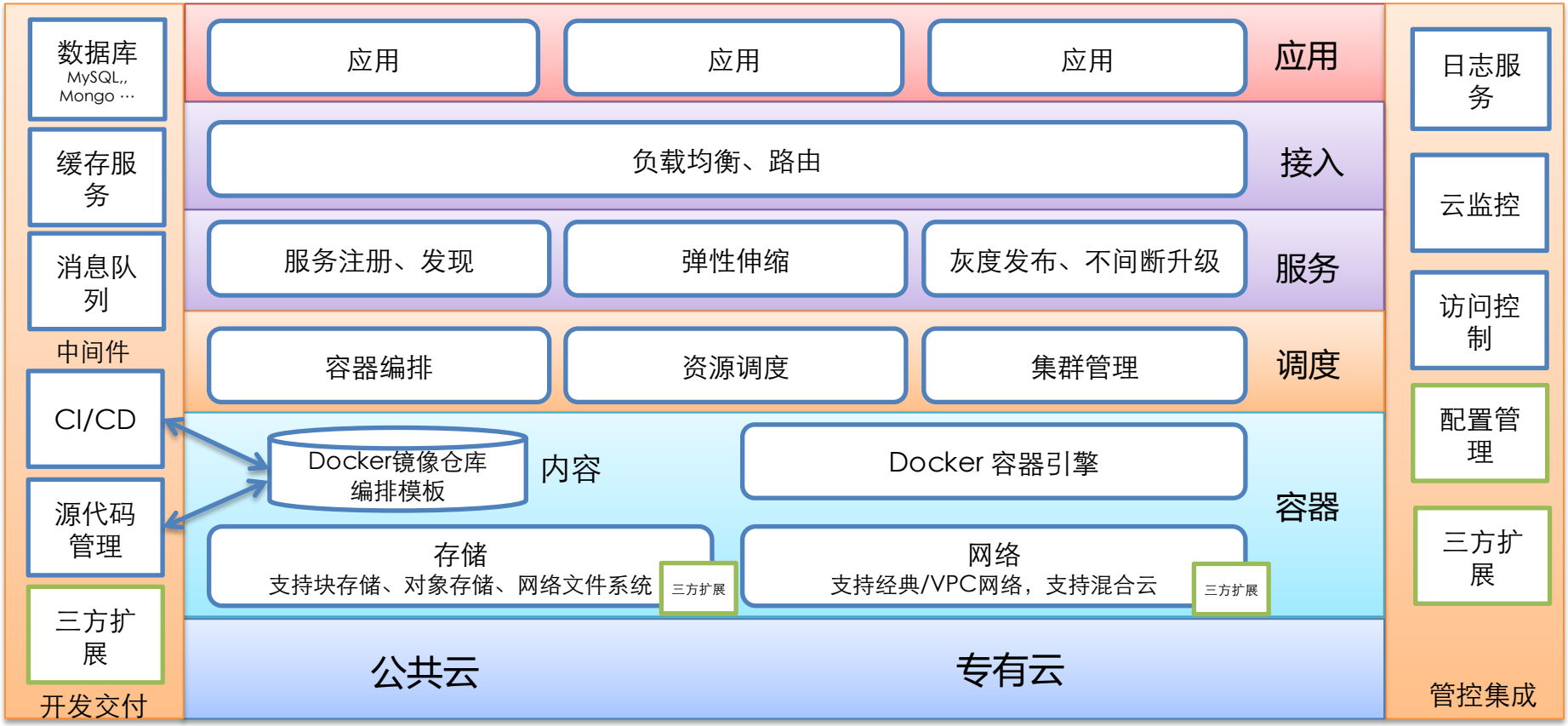
```
version: '2'
services:
  wordpress:
    image: wordpress:4.5
    restart: always
    links:
      - 'db:mysql'
    volumes:
      - 'wp_upload:/var/www/html/wp-content/uploads'
    environment:
      - WORDPRESS_DB_USER=blog
      - WORDPRESS_DB_PASSWORD=xxxxxx
      - WORDPRESS_DB_NAME=wordpress
      - availability:az==2
    labels:
      aliyun.probe.url: http://container
      aliyun.routing.port_80: http://wordpress
      aliyun.scale: '3'
      aliyun.log_store_wordpress: stdout
  db:
    external:
      host: rdsxxxxxx.mysql.rds.aliyuncs.com
      ports:
        - 3306
volumes:
  wp_upload:
    driver: ossfs
    driver_opts:
      bucket: acs-sample-wordpress
```

- 完全兼容 Docker Compose/Swarm
- 声明的方式支持容器及云服务
- 支持微服务架构

一键部署到云上

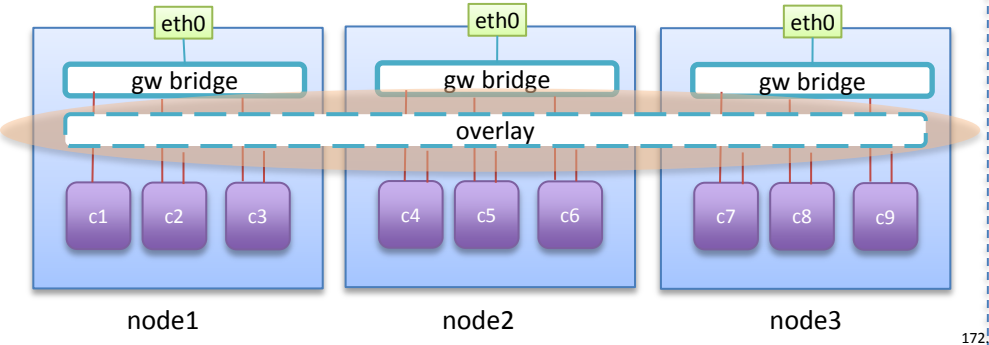


阿里云容器服务

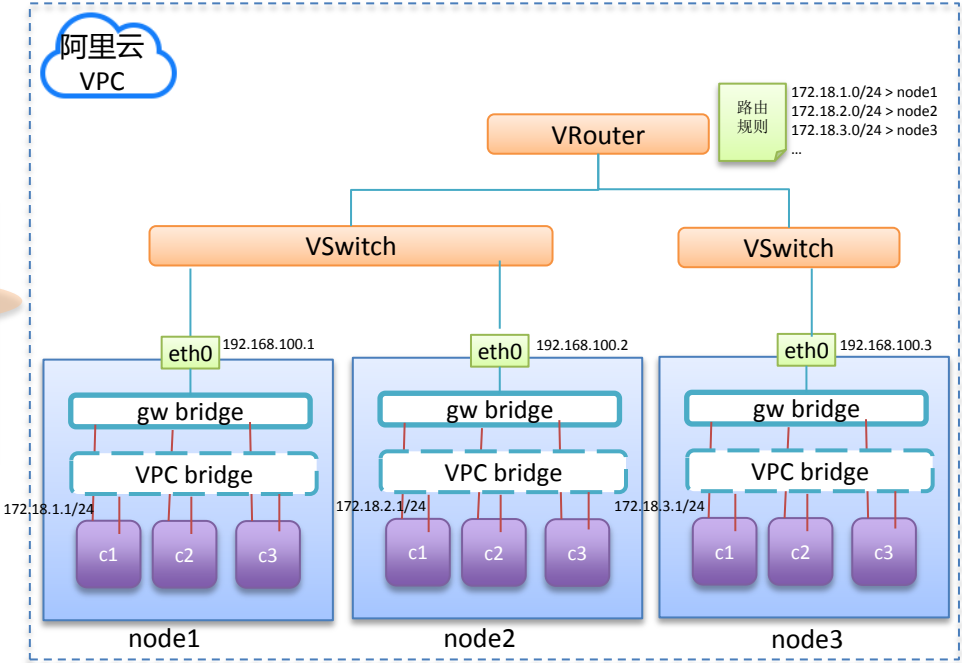

数据库
MySQL,,
Mongo …
缓存服务
消息队列
中间件
CI/CD
源代码管理
三方扩展
开发交付
应用
应用
应用
应用
负载均衡、路由
接入
服务注册、发现
弹性伸缩
灰度发布、不间断升级
服务
容器编排
资源调度
集群管理
调度
Docker镜像仓库
编排模板
内容
Docker 容器引擎
存储
支持块存储、对象存储、网络文件系统
三方扩展
网络
支持经典/VPC网络，支持混合云
三方扩展
容器
公共云
专有云
日志服务
云监控
访问控制
配置管理
三方扩展
管控集成

跨主机容器网络

- 每个容器一个独立IP
- 容器跨宿主机直接通信
- 容器网络可以通过DNS解析容器地址



Docker 原生Overlay网络（VXLAN）



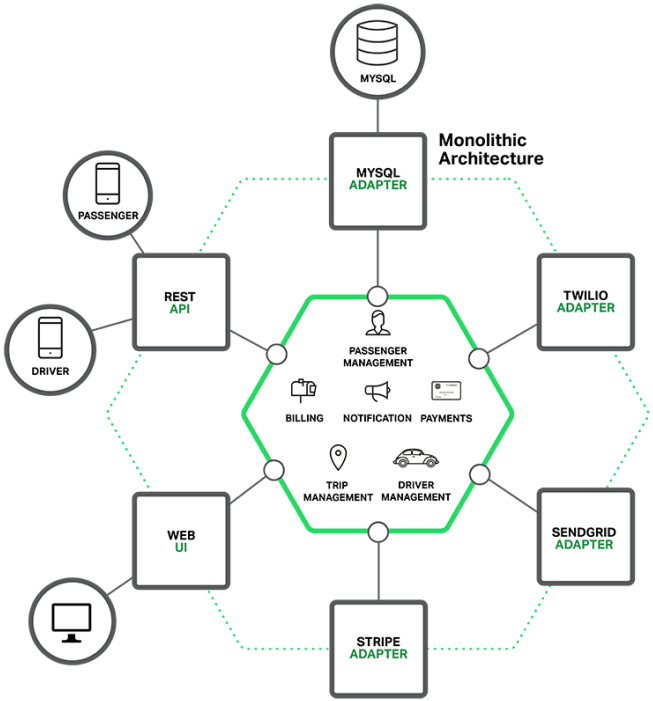
利用VPC网络方案

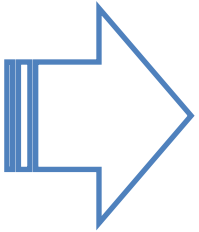
日程

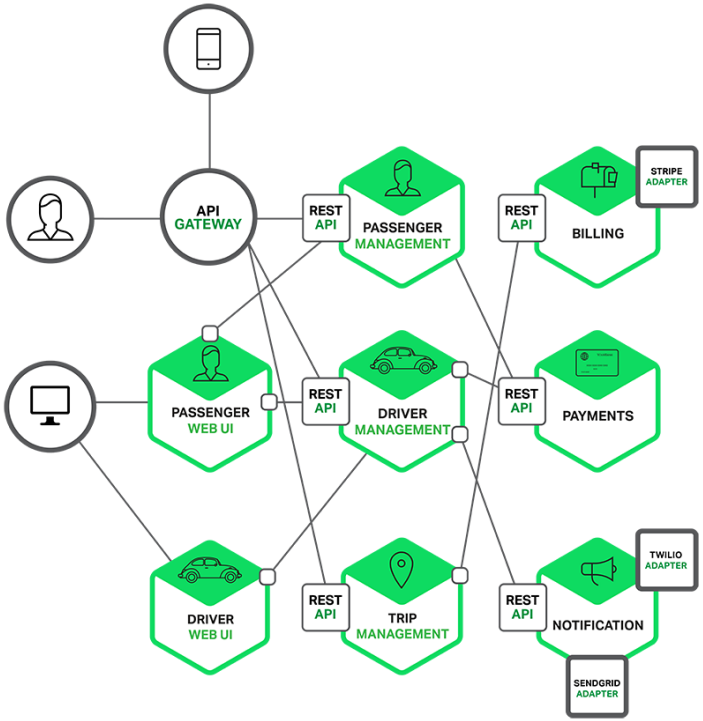
- Docker编排技术概述
- 容器即服务(Container as a Service)
  - 微服务支持
  - DevOps
- 未来发展趋势

微服务架构

图片来自：https://www.nginx.com/blog/introduction-to-microservices/



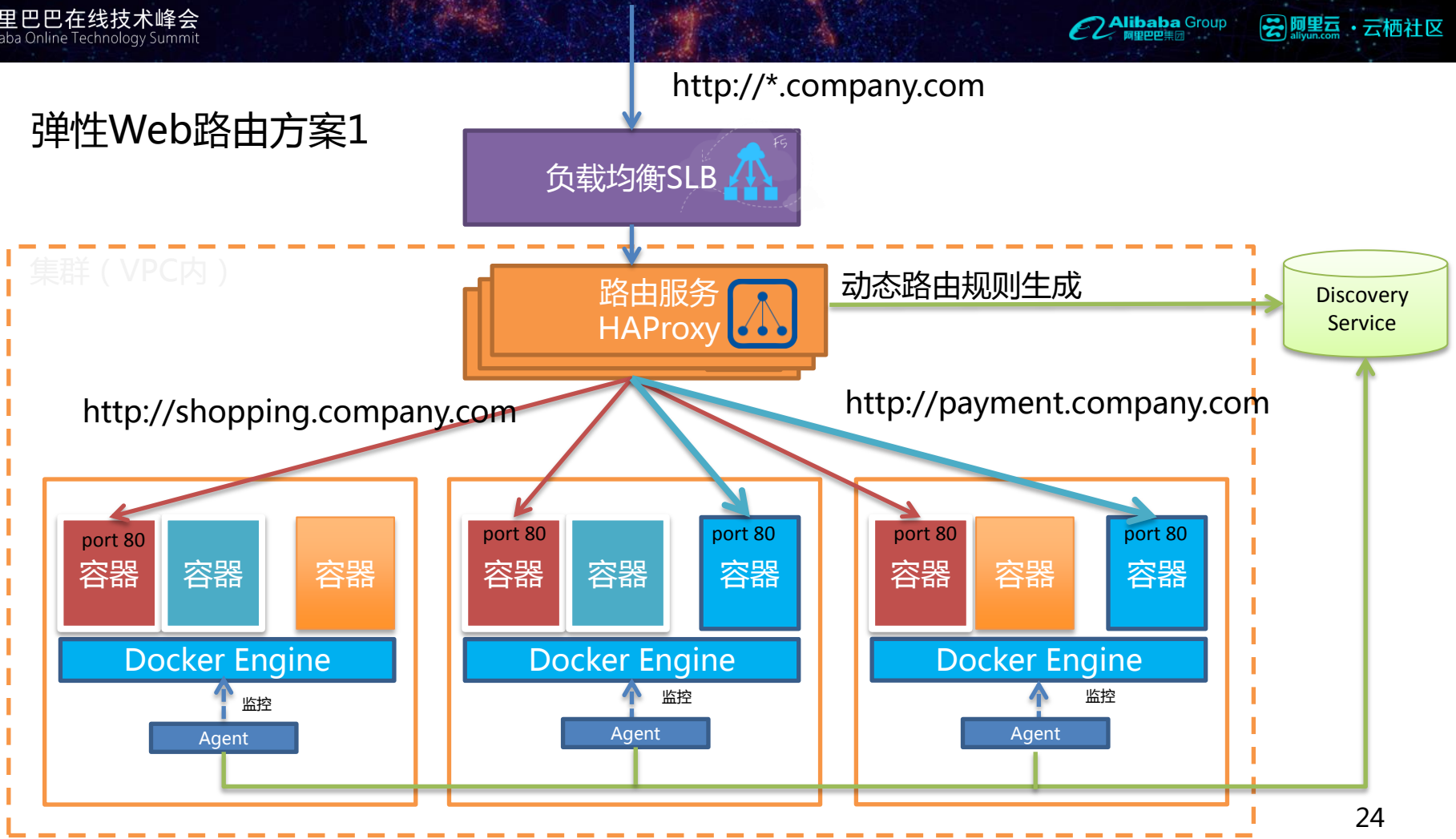




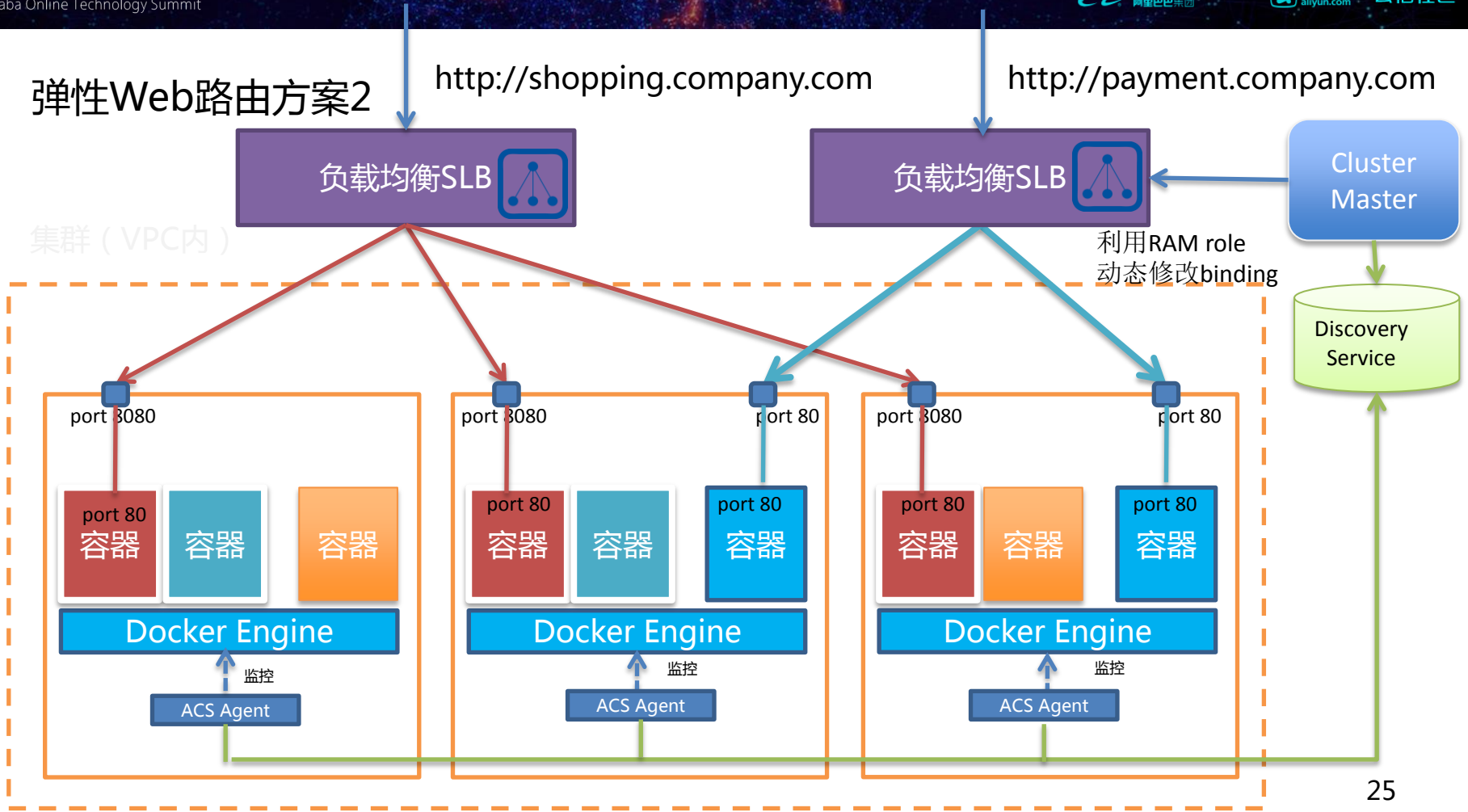
一个大而全的单体应用

一组解耦的、自治的、协同工作的服务

弹性Web路由方案1

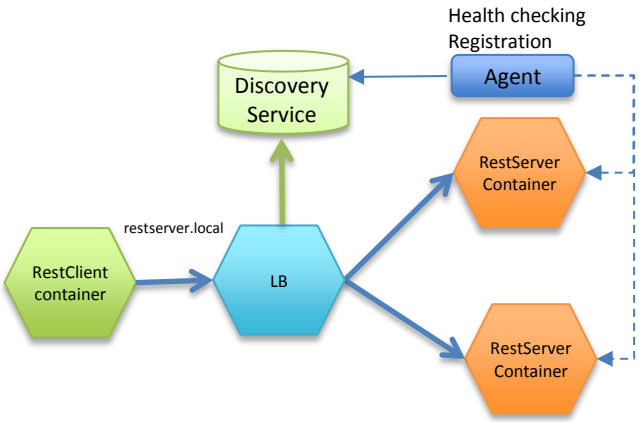

http://*.company.com
负载均衡SLB
集群（VPC内）
路由服务
HAProxy
动态路由规则生成
Discovery Service
http://shopping.company.com
http://payment.company.com
port 80
容器
容器
容器
Docker Engine
监控
Agent
port 80
容器
容器
port 80
容器
Docker Engine
监控
Agent
port 80
容器
容器
port 80
容器
Docker Engine
监控
Agent

弹性Web路由方案2


http://shopping.company.com
http://payment.company.com
负载均衡SLB
负载均衡SLB
Cluster Master
利用RAM role
动态修改binding
Discovery Service
集群（VPC内）
port 3080
port 80
port 80
port 80
port 3080
port 80
port 80
port 80
port 3080
port 80
port 80
port 80
容器
容器
容器
容器
容器
容器
容器
容器
容器
Docker Engine
Docker Engine
Docker Engine
监控
监控
监控
ACS Agent
ACS Agent
ACS Agent

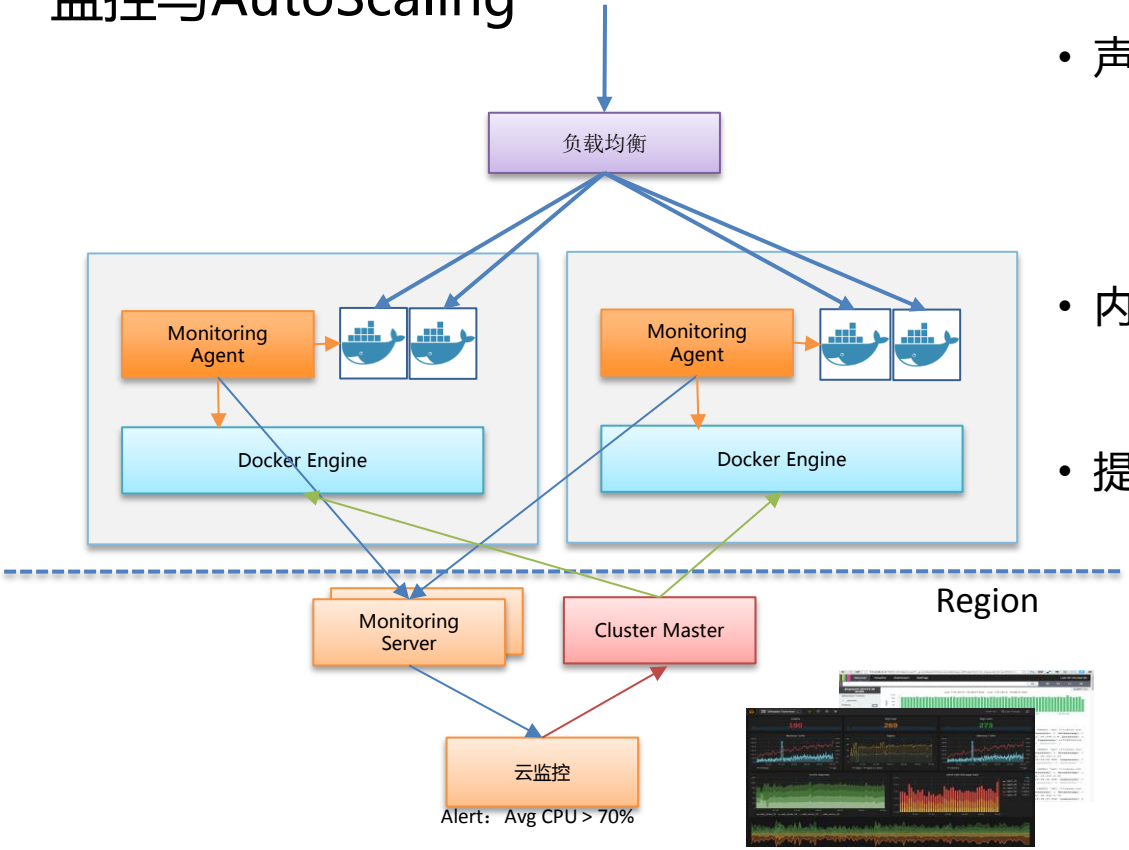
实现无关的服务发现与负载均衡

```
server:
  image: nginx
  labels:
    aliyun.routing.port_80: restserver.local
    aliyun.scale: "2"
    aliyun.probe.url: http://container:80
    aliyun.probe.initial_delay_seconds: "2"
    aliyun.probe.timeout_seconds: "2"
client:
  image: test_app
  external_links:
    - "restserver.local"
```



- 和DNS服务发现相比
  - 支持灵活的负载均衡策略
  - 避免TTL问题
  - 支持健康检查

监控与AutoScaling



- 声明式方式定义弹性伸缩策略

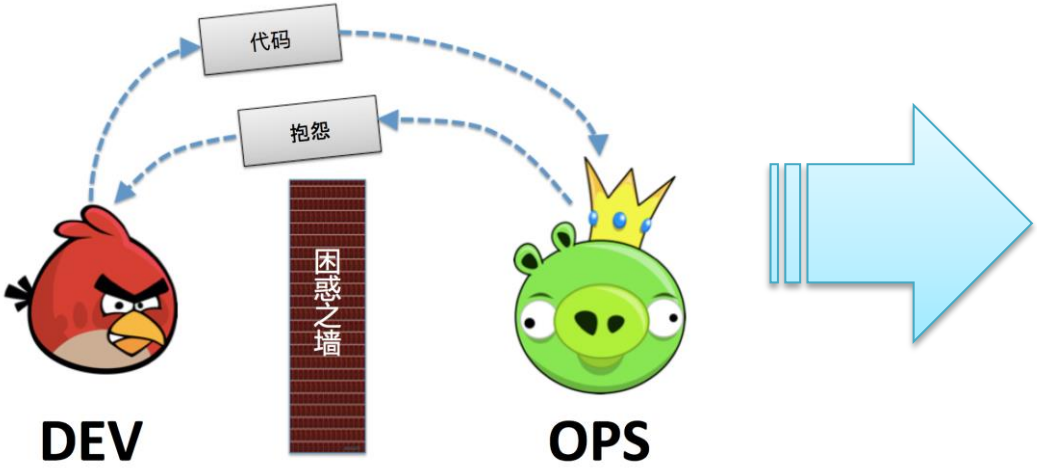
```
aliyun.auto_scaling.max_cpu: 70
aliyun.auto_scaling.step: 2
```

- 内置云监控集成
- 提供插件机制支持开源、三方监控集成
  - Input: nagios, apache, docker, UDP, ...
  - Output: Influxdb, prometheus, kafka ...

日程

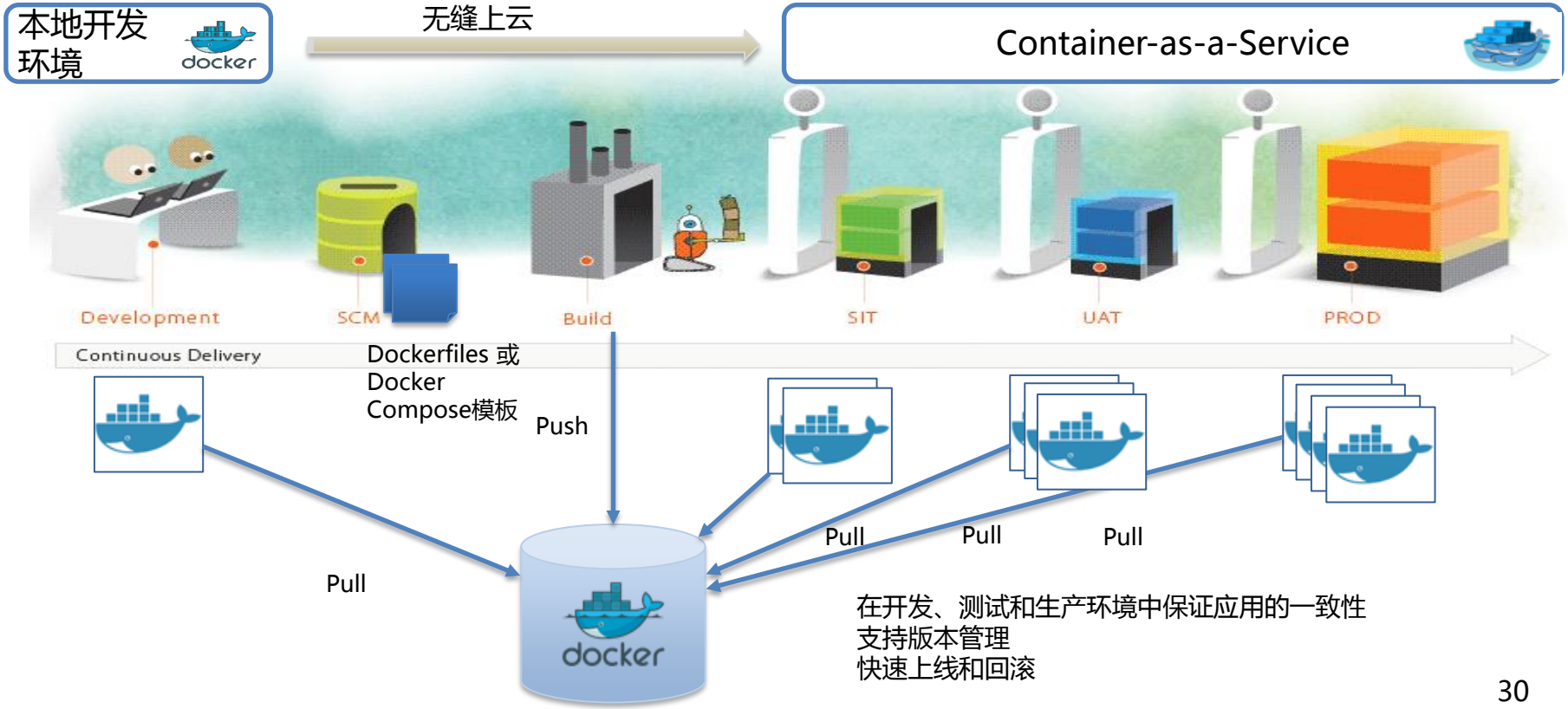
- Docker编排技术概述
- 容器即服务(Container as a Service)
  - 微服务支持
  - DevOps
- 未来发展趋势

从Dev vs. Ops到DevOps

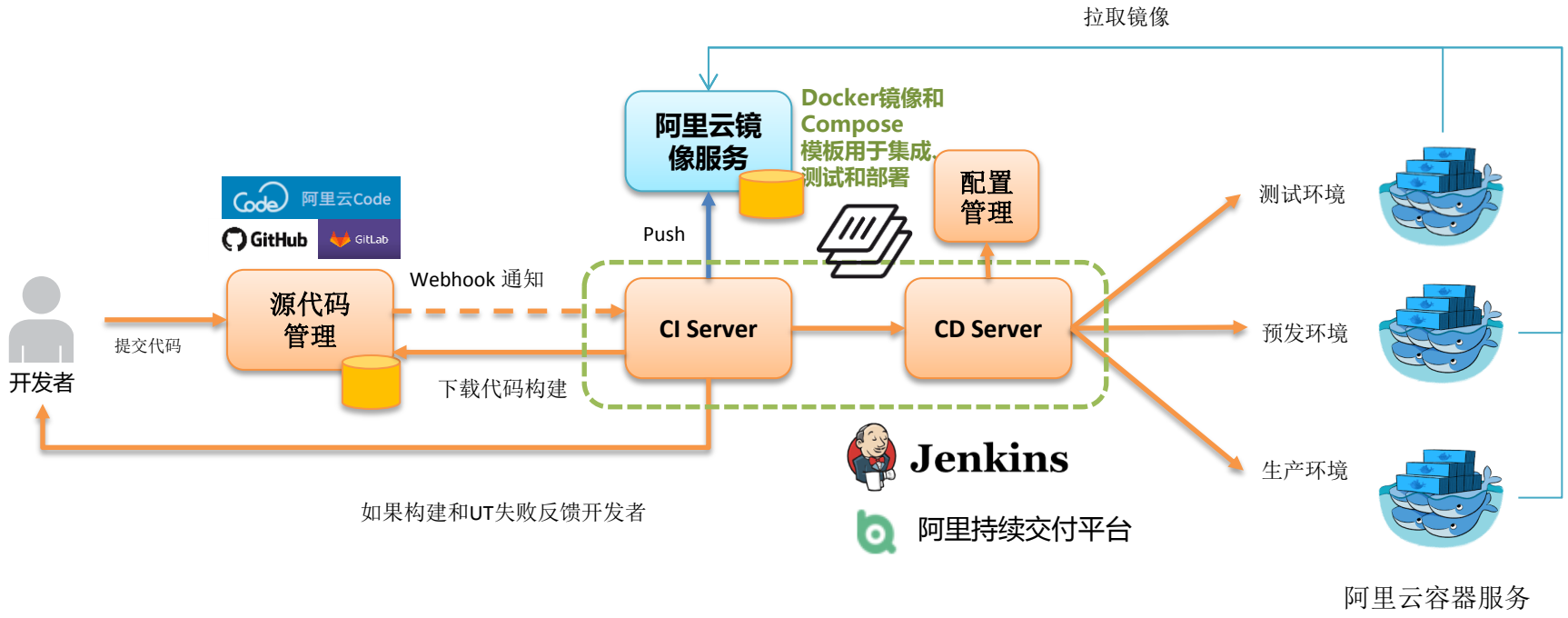


图片来自：http://servicevirtualization.com/devops-design-poll/

利用容器实现持续集成和交付 Build Once and Deploy Everywhere



完整的容器化持续交付流程

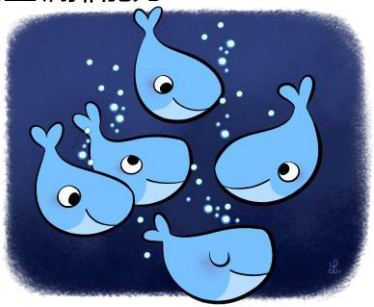


日程

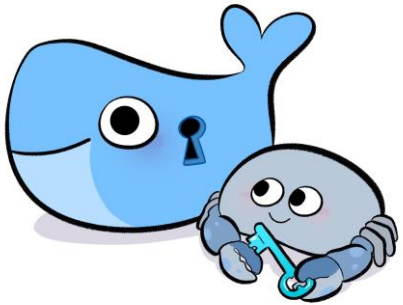
- Docker编排技术概述
- 容器即服务(Container as a Service)
  - 微服务支持
  - DevOps
- 未来发展趋势

The Best Way to Orchestrate Docker is Docker

Docker 1.12 已经内置编排能力

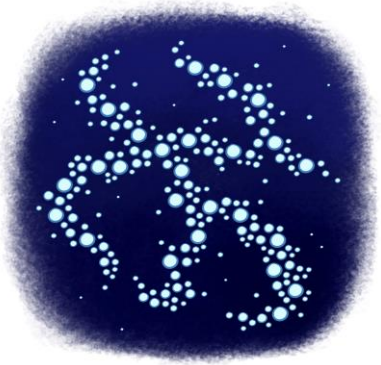


Swarm mode



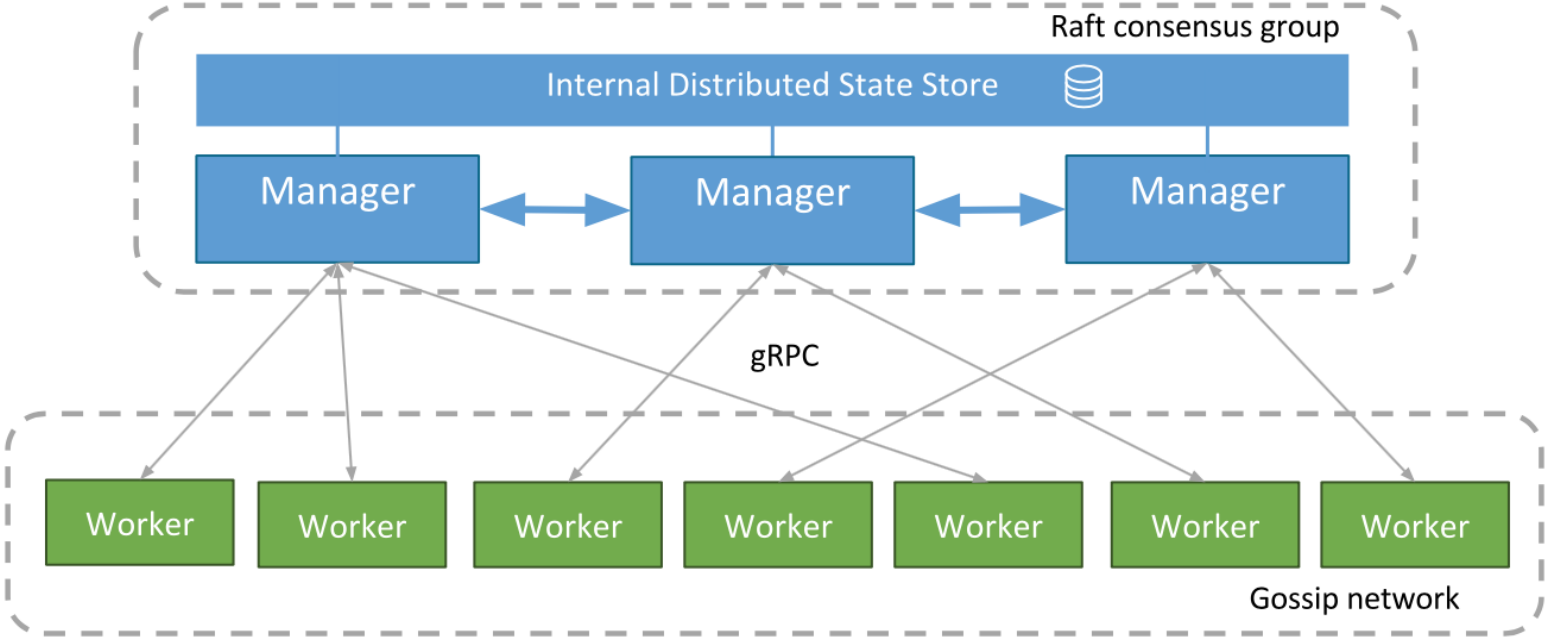
Cryptographic node identity

Service API



Built-in routing mesh

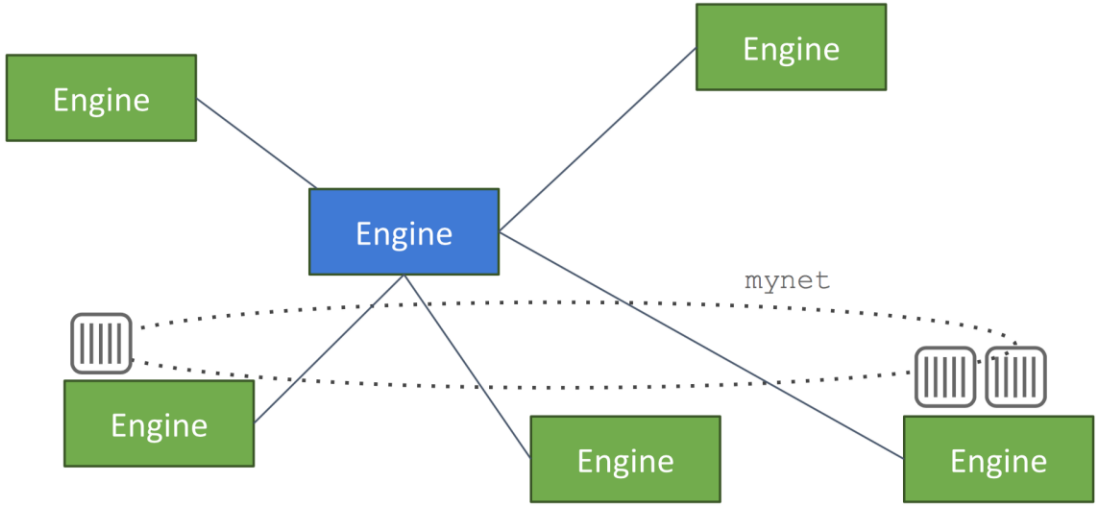
Docker Swarm 模式




```
docker swarm init
docker swarm join <MASTER_IP>:2377
```

参见：在阿里云上体验Docker 1.12内置的编排能力

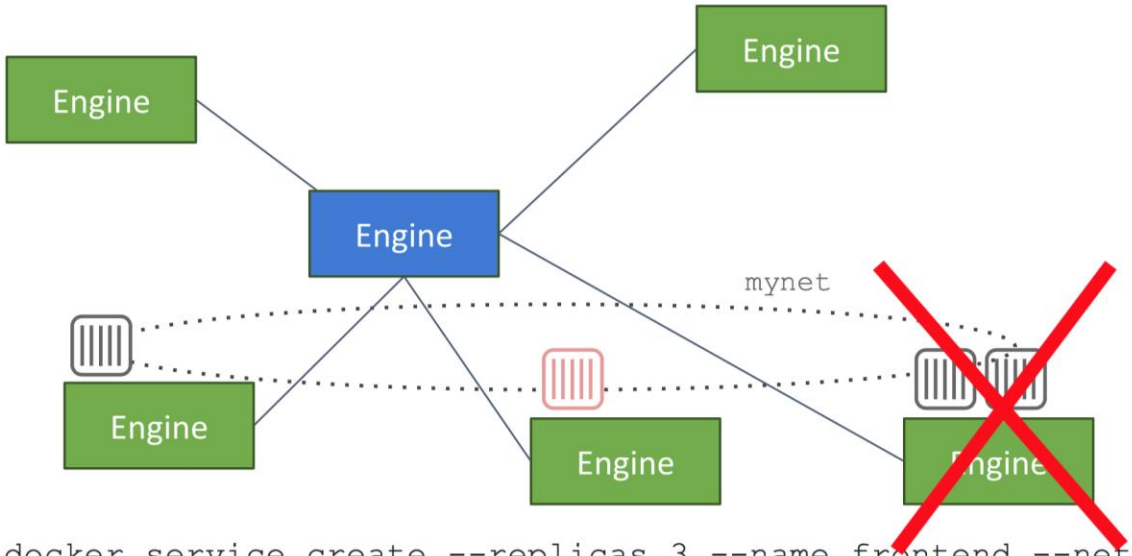
服务 Services




```
$ docker service create --replicas 3 --name frontend --network mynet
 --publish 80:80/tcp frontend_image:latest
```

```
$ docker service create --name redis --network mynet redis:latest
```

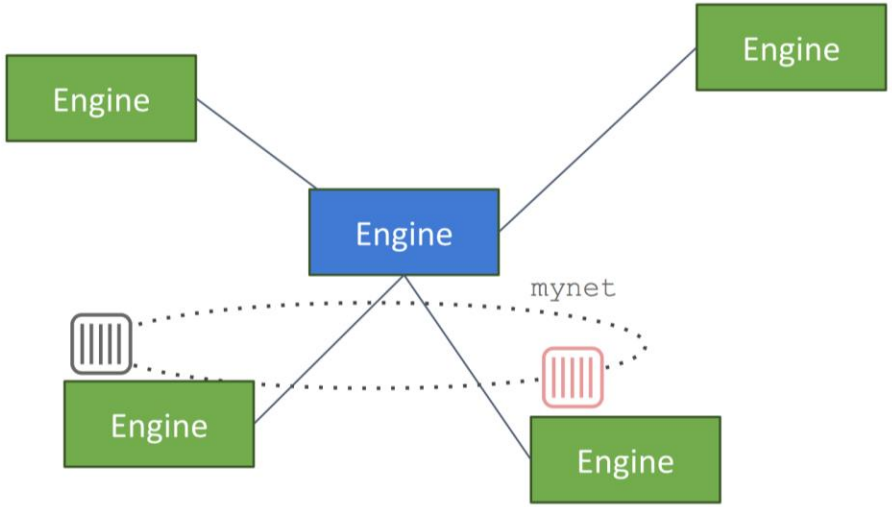
节点失败




```
$ docker service create --replicas 3 --name frontend --network mynet
 --publish 80:80/tcp frontend_image:latest
```

```
$ docker service create --name redis --network mynet redis:latest
```

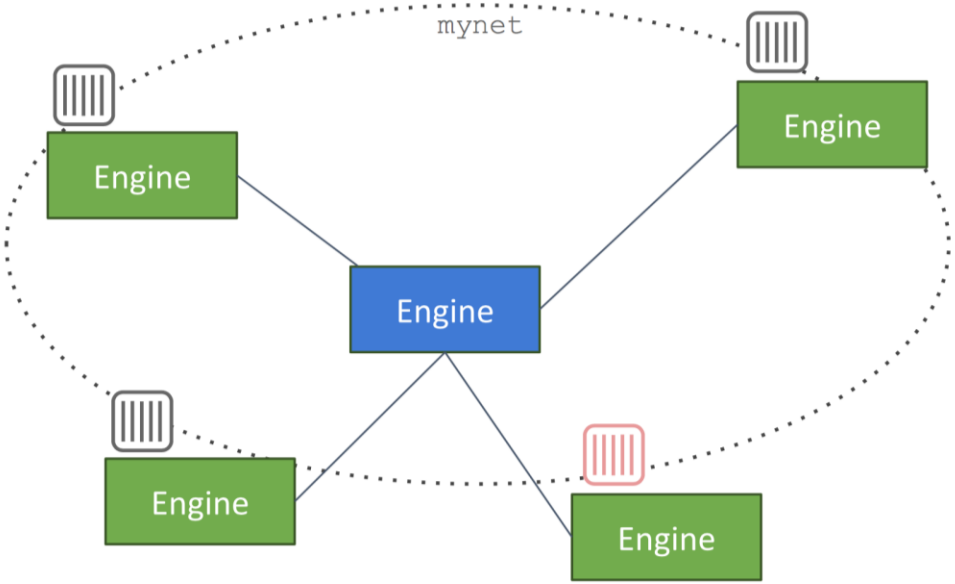
期望状态≠实际状态




```
$ docker service create --replicas 3 --name frontend --network mynet
 --publish 80:80/tcp frontend_image:latest
```

```
$ docker service create --name redis --network mynet redis:latest
```

自动恢复

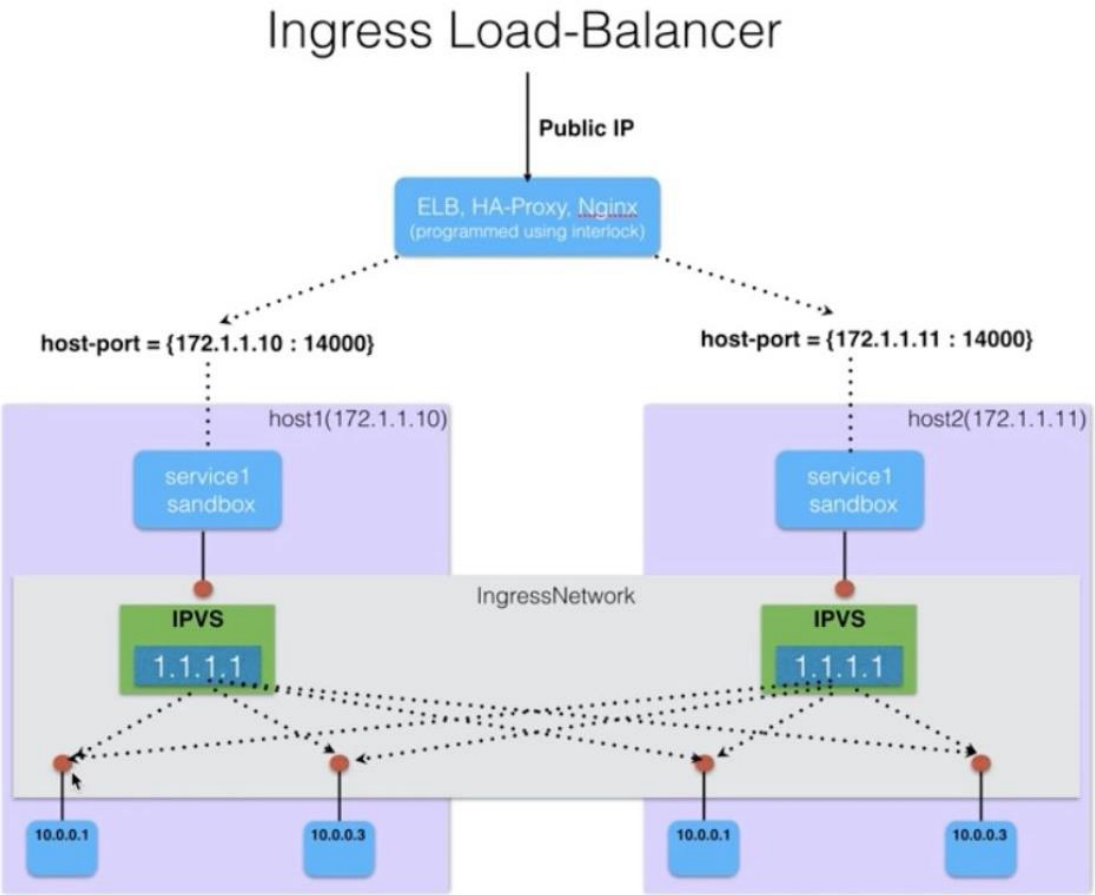



```
$ docker service create --replicas 3 --name frontend --network mynet
 --publish 80:80/tcp frontend_image:latest
```

```
$ docker service create --name redis --network mynet redis:latest
```

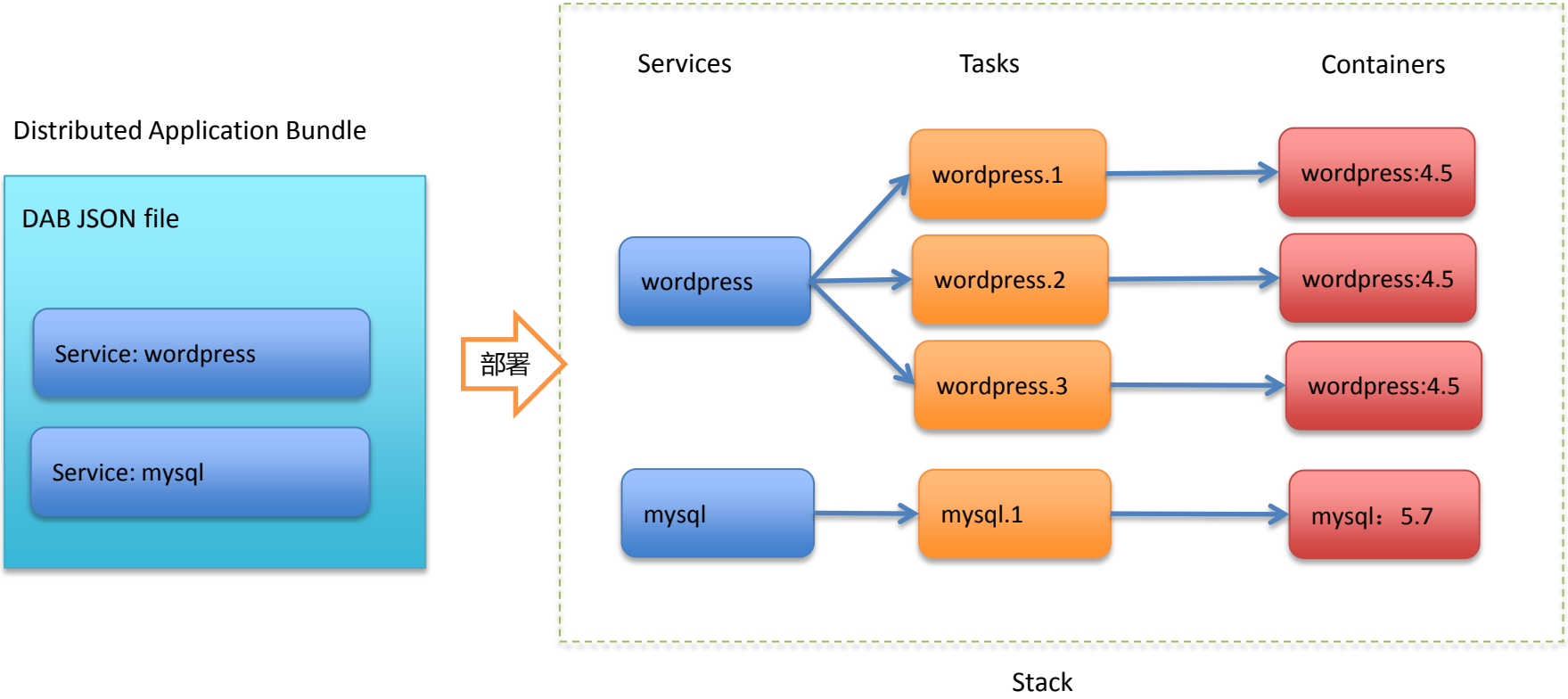
Routing Mesh

- 每个服务一个VIP
- IPVS实现负载均衡
- 动态/手工分配PublishedPort
- 每个worker参与路由



参见：在阿里云上体验Docker 1.12的路由能力和容器应用分发部署

Distributed Application Bundle 与 Stack



复用已有Docker Compose

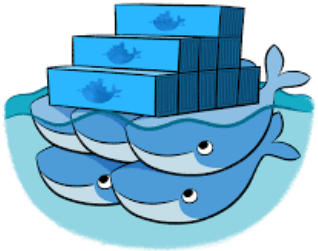
docker-compose bundle　　docker deploy

Container Orchestration War

kubernetes
by Google

三国杀™

Docker Swarm Mode



MESOSPHERE

Thank you !

阿里云容器团队博客
https://yq.aliyun.com/teams/11

容器服务钉钉群

Docker云端漫步

Docker Machine

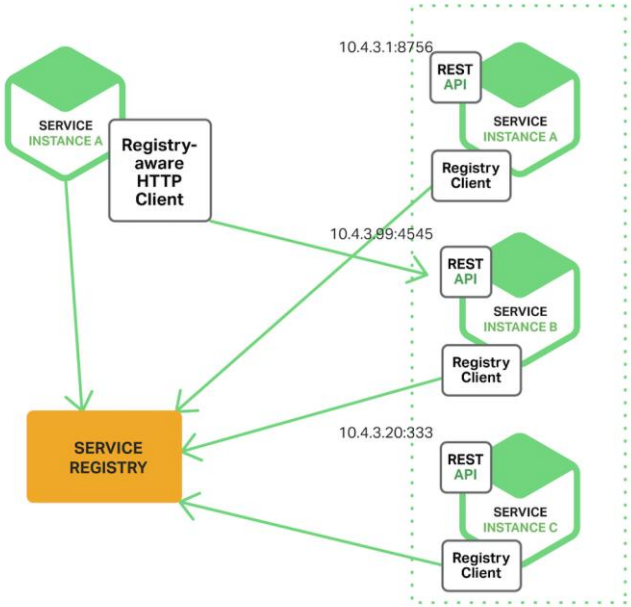
- 配置安装
  - 安装Docker Toolkit
  - 安装云驱动
    - ECS driver for Docker Machine
    - AWS, GCE, 等等.
- 在阿里云创建Docker运行环境

```
export ECS_ACCESS_KEY_ID=xxxxxx
export ECS_ACCESS_KEY_SECRET=xxxxxx
docker-machine create --driver aliyunecs mytest
eval "$(docker-machine env mytest)"
docker run -d nginx
```

https://yq.aliyun.com/articles/6809

在动态环境中的服务发现模式

客户端发现模式



服务端发现模式