



上汽集团基于容器技术的 尝试实践

信息系统部
上海汽车集团股份有限公司

目录

- 背景介绍
- 需求分析
- 上汽在容器领域的实践
- 经验体会

上汽集团与上汽IT

安亭基地



自主品牌上海研发基地

南京基地



自主品牌A级车平台



金桥基地

上汽集团云数据中心

英国基地



自主品牌英国研发基地

转型当中

- **多业务需求方**
面向客户：下属企业，包括整车、金融、汽车后市场、IOT；
- **多元化应用架构**
多应用架构:面向互联网、面向企业、面向制造、面向研发；
- **统一资源服务**
平台级服务：资源提供，标准服务目录。

Docker 技术背景



Image:
Docker Image

Runtime:
Docker Engine



改变与挑战

- 容器技术带来的改变

容器镜像标准化了应用交付
构建统一的IT支持平台
自动化、可编程化的标准流程
应用模块化（**SaaS**基础）

- 容器面临的挑战

介乎开发和基础设施之间
涉及应用架构
传统应用的改造

案例分享

上汽产品互联网推广活动

系统挑战

- ❑ 互联网系统高并发大流量
- ❑ 业务动态伸缩弹性需求
- ❑ 面向服务架构设计
- ❑ 分布式团队协作、快速迭代
- ❑ 标准化运维模式



人气挑战 云翼犒赏

有你喜爱车队么？赶紧来支持！
云翼犒赏之无人机奖给同样志在探索的你！
单价**2999**元，每天**1**台

* 奖品以实物为准

活动规则

云翼犒赏幸运者

DAY1 186****3552	DAY2 137****8966	DAY3 138****8657
DAY4 185****5098	DAY5 ?	

YunOS战队 ❤️ 364

爱卡汽车战队 ❤️ 581

魅蓝E队 ❤️ 180

小辣椒超龄少年战队 ❤️ 554

飞亚达战队 ❤️ 113

萝卜车队 ❤️ 406

系统设计思路

□ 微服务架构

- 前后端解耦，服务组件独立部署

□ 分布式集群化设计

- 系统高可用随时在线，消除单点故障

□ 全容器化交付

- 资源高效管理，应用动态伸缩

□ DevOps 和持续交付流水线

- 应用快速迭代

技术组件选型

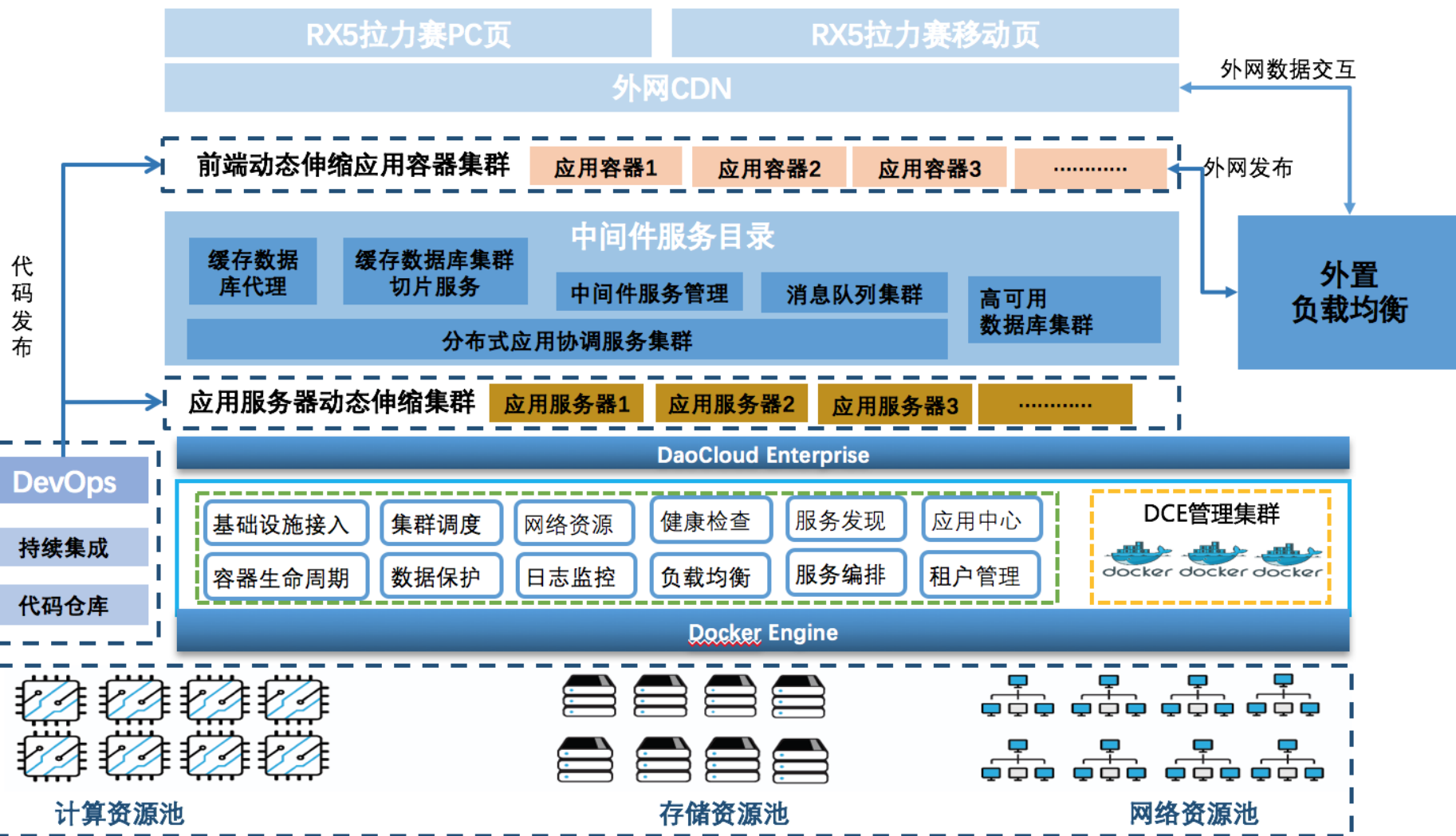
❑ 平台：DaoCloud Enterprise 应用云平台

❑ 开源中间件

- 缓存数据库：Codis 分组集群，支持分片操作
- 消息队列：Kafka
- 数据库：MySQL 高可用主从半同步数据库
- 分布式协调服务框架：ZooKeeper
- 应用服务器：Tomcat

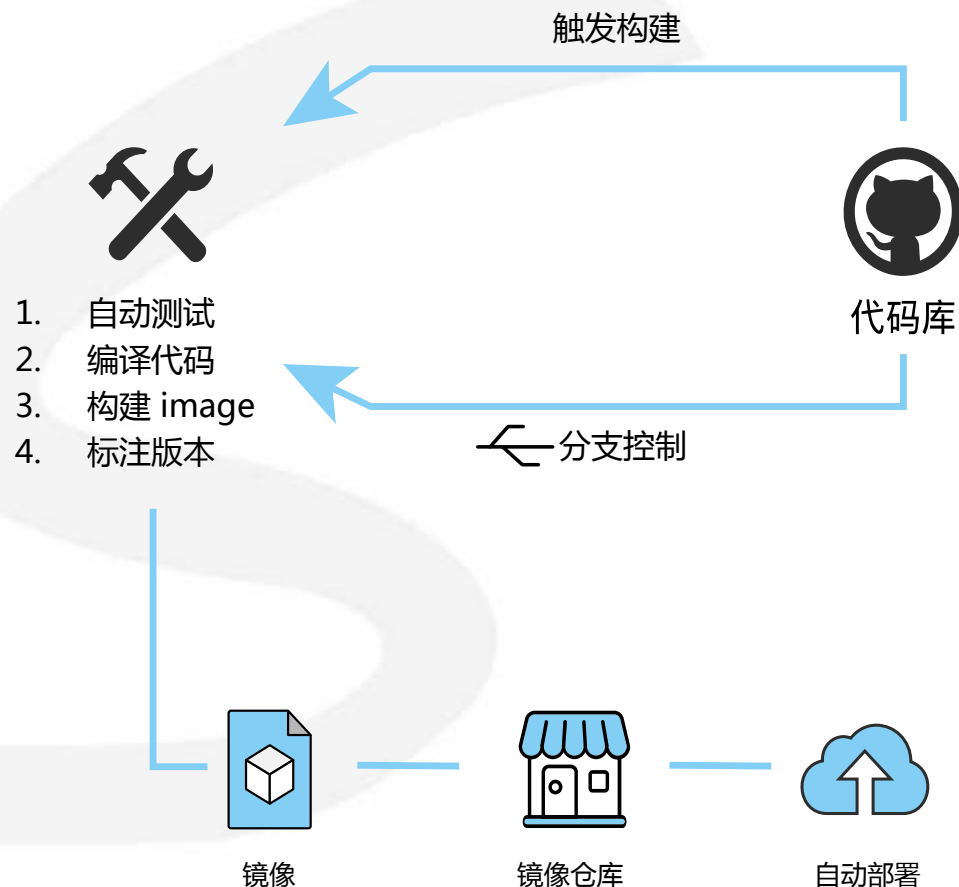
❑ 前端业务逻辑：NodeJS

系统架构图








快速迭代

- 全容器化交付流水线
- 一键自动部署
- 应用多版本共存
- 应用自动化持续发布
- 快速伸缩升级和回滚
- 标准化运维（监控、日志、告警等）
- 多环境配置：开发、测试、预生产
- 跨地域团队协作



系统容器化落地部署



 应用
  容器
  存储
  网络

全部 11 | 运行中 11 | 系统 1

部署

共 11 个应用，已选中 0 个应用

<input type="checkbox"/>	RX5-activity-kafka-manager 普通应用 不是最新	1 服务 kafka-manager	0 存储卷
<input type="checkbox"/>	RX5-activity-saiconode 普通应用 最新	5 服务 saiconode5、saiconode4、...	0 存储卷
<input type="checkbox"/>	RX5-activity-praise-1 普通应用 最新	10 服务 praise-9、praise-8、pr...	0 存储卷
<input type="checkbox"/>	RX5-activity-Kafka-Cluster 普通应用 不是最新	6 服务 kafka1、kafka2、kafka3...	0 存储卷
<input type="checkbox"/>	RX5-activity-Codis 普通应用 最新	16 服务 codis-fe、zookeeper、c...	0 存储卷

容器 | 配置 | 日志

环境变量

键	值
BROKER_ID	2
constraint:node	=docker-PRD-NTNX-*
ZOOKEEPER_CONNECT	zookeeper1:2181,zookeeper2:2181,zookeeper3:2181
ADVERTISED_LISTENERS	kafka2:9092

添加环境变量

容器 | 配置 | 日志 | 监控

```

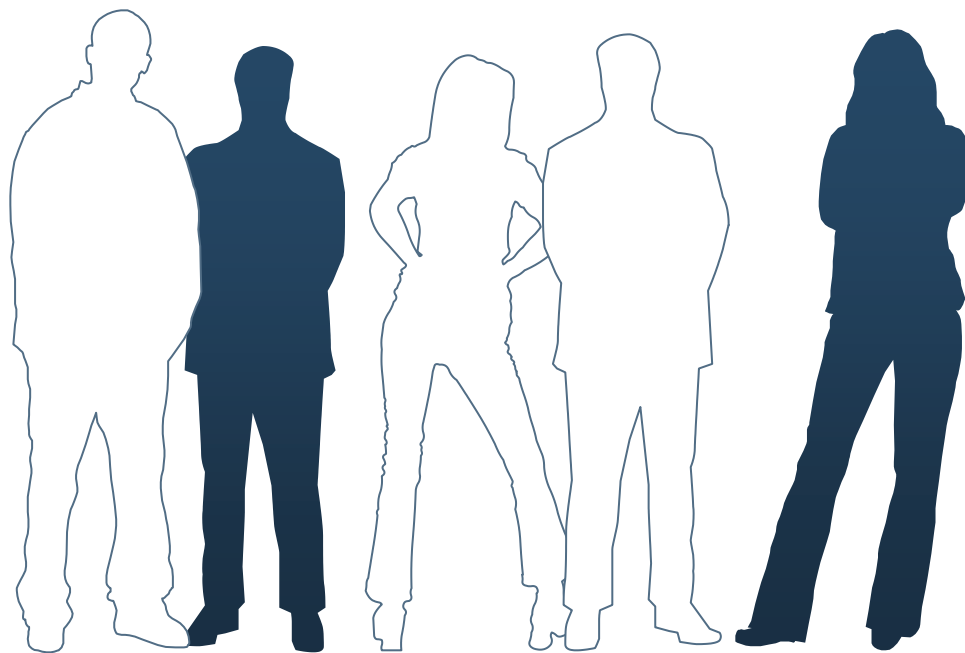
2016-10-19 11:58:03 rx5activitykafkacluster_kafka2_1 [2016-10-19 03:58:03,651] INFO [Group Metadata Manager on Broker 2]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
2016-10-19 12:08:03 rx5activitykafkacluster_kafka2_1 [2016-10-19 04:08:03,651] INFO [Group Metadata Manager on Broker 2]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
2016-10-19 12:18:03 rx5activitykafkacluster_kafka2_1 [2016-10-19 04:18:03,651] INFO [Group Metadata Manager on Broker 2]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
  
```

经验分享

- ❑ 使用 Docker 原生的 Overlay 网络，通过服务名进行内部访问，解决了配置文件繁琐的问题。
- ❑ 交付流程一切通过容器化，代码的编译构建都在容器中进行，实现了更高的迭代速度。
- ❑ 通过代码实现了 HAProxy 的后端服务自发现，解决了容器横向扩缩的负载配置问题。
- ❑ 使用 Dockerbeat、ElasticSearch、Kibana 进行容器日志收集，统一管理。

体会





Thank You