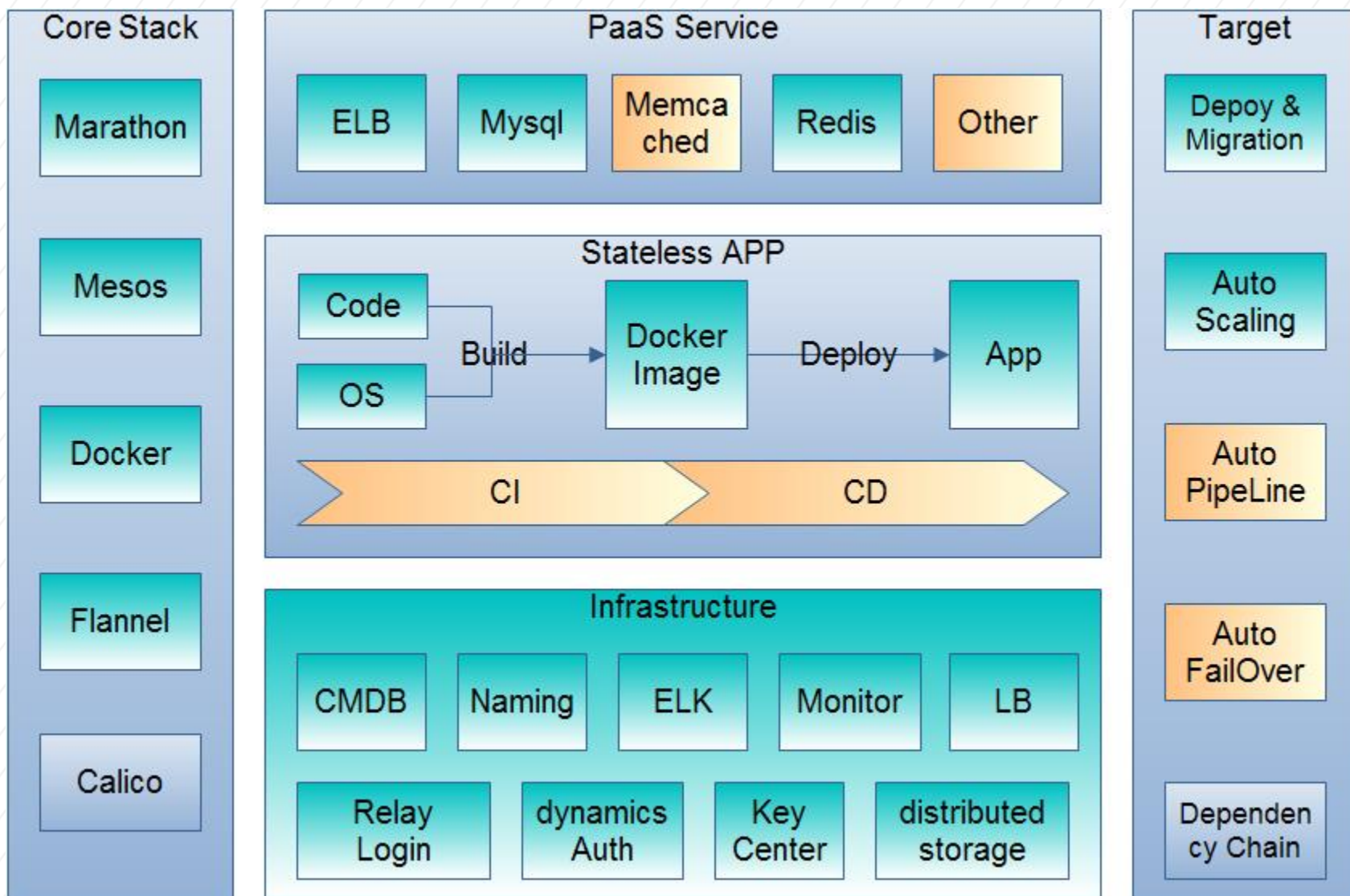




小米弹性调度平台 Ocean实战

孙寅，现在小米负责运维基础设施、基础平台

1 体系概览

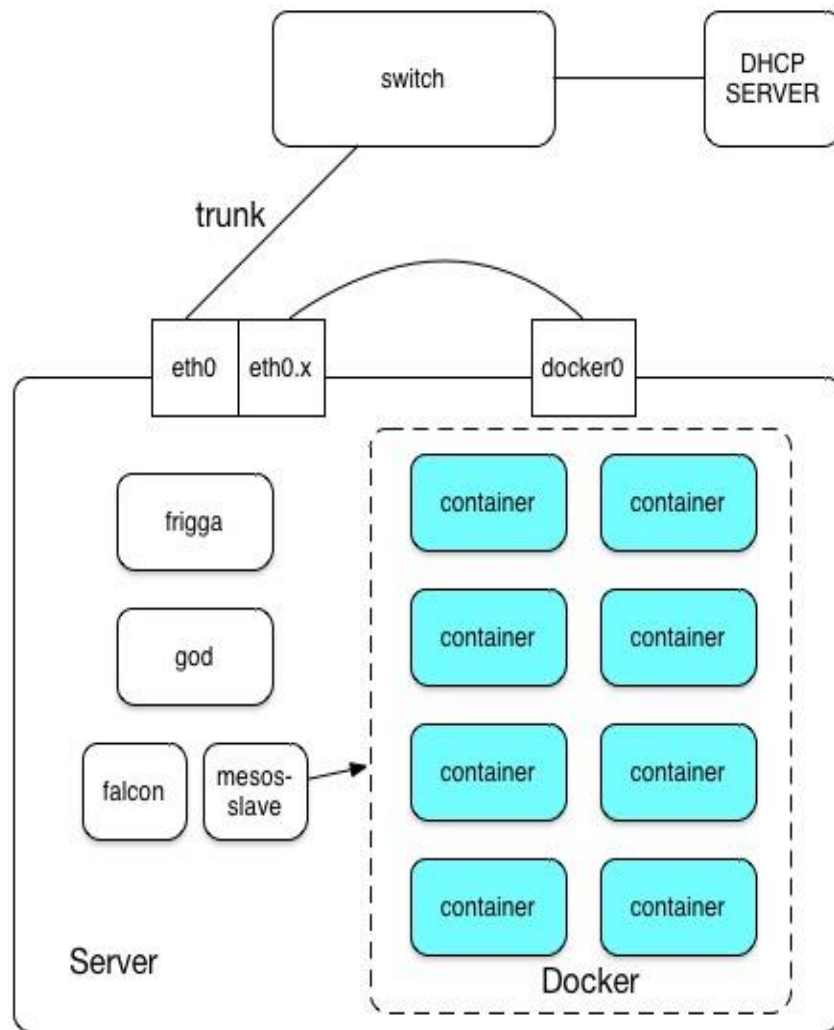




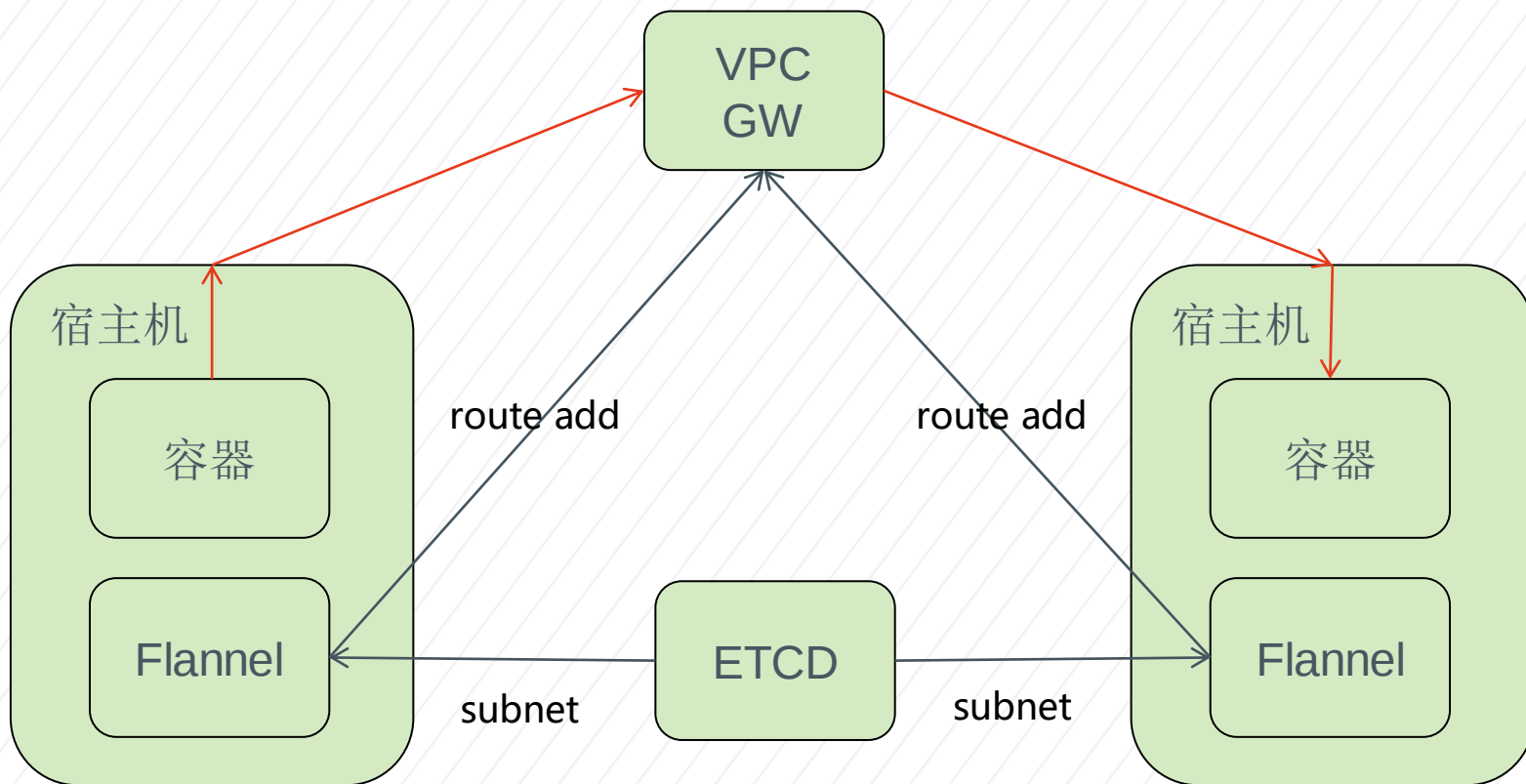
Docker落地

1 物理机房容器网络

- 真实内网IP，便于标识和定位
- 与原有物理网络天然互通
- 吞吐、延迟与纯物理网络几无差别

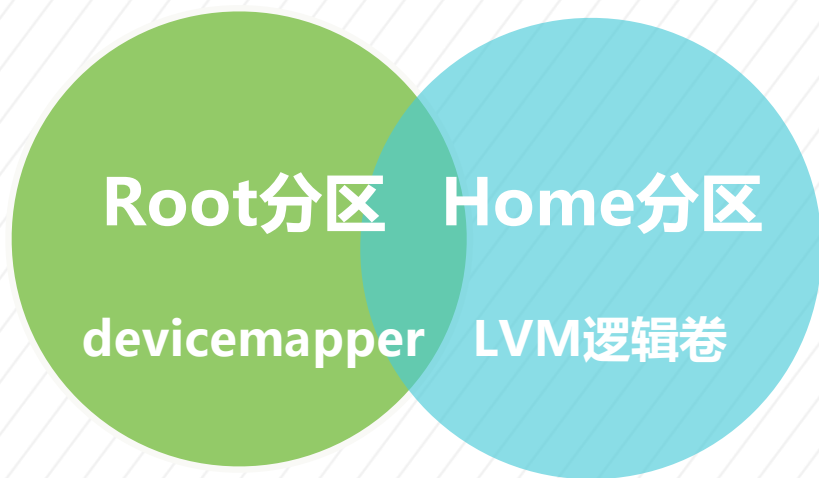


1 云机房容器网络



2

文件系统



目标

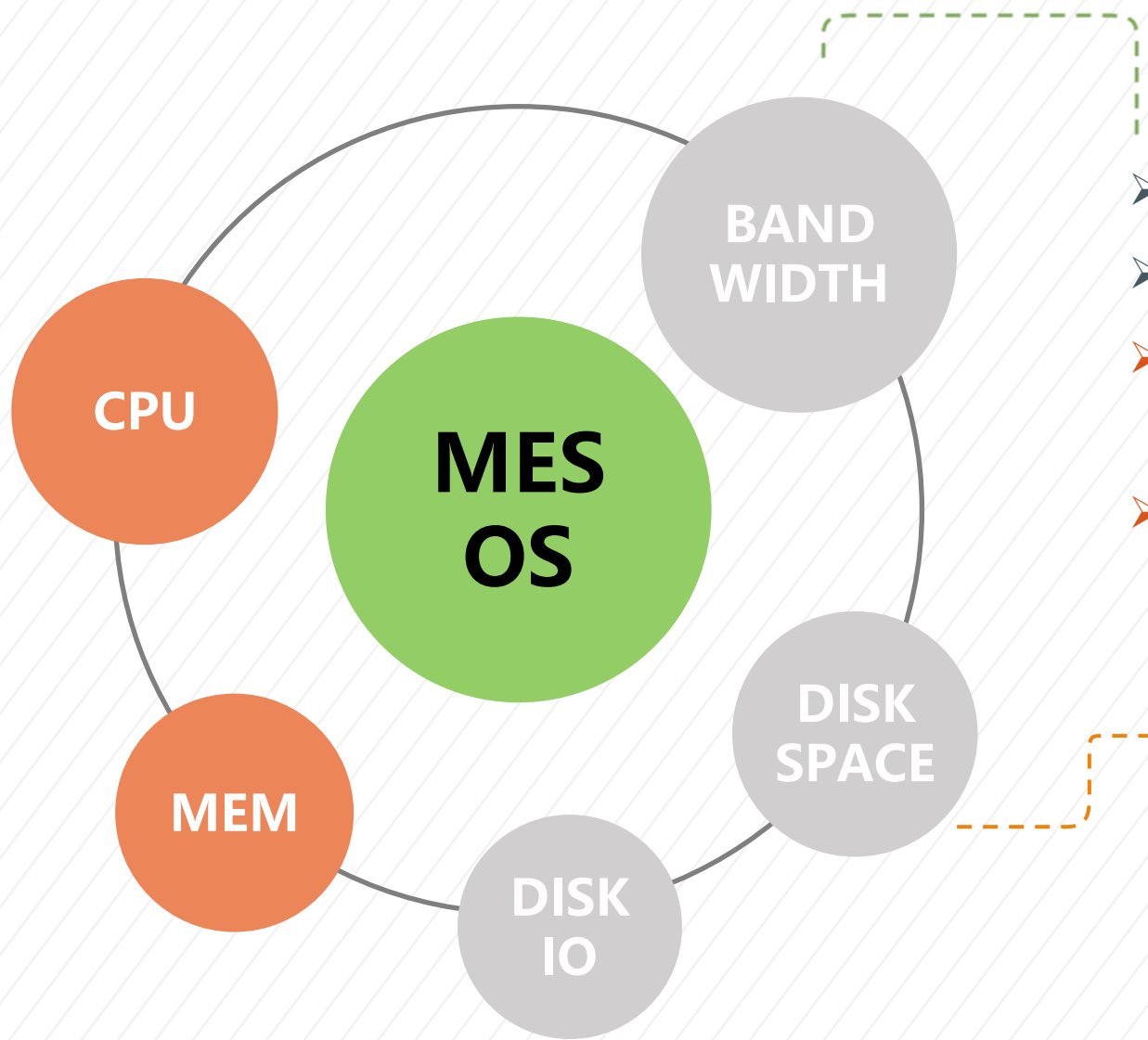
- 容器磁盘空间大小可分配
- 保证磁盘IO性能不下降

技术考量

- aufs、btrfs等等存储方式都会损耗IO性能
- 一个宿主机devicemapper大小固定
- Home分区引入LVM动态设置空间大小

3

资源隔离资源汇报



- 扩展Mesos Resource
- 环境变量传递资源大小
- Docker中嵌入TC限速模拟网络带宽隔离
- DM+LVM隔离磁盘空间, mesos-slave hook创建和清理LVM

4

build

封装规范Dockerfile

➤ FROM + Dockerfile 分片

➤ CMD

➤ PORT

服务信息

require* :?	centos	请选择
+ port* :?		-
+ entry* :?		-
+ cron:?		-
+ add:?	源文件	目标路径
+ env:?	变量名	值
stopentry:?		-
build_require:?		请选择

- DHCP
- 内置服务管理
- 进程组
- 进程HealthCheck
- 进程启动退出钩子
- 回收僵尸进程
- DEBUG模式
- 与内部基础设施对接
 - Marathon HealthCheck
 - Falcon
 - DoorGod (安全免密登陆)
 - Mysql Auth



服务发现

2

服务发现

RPC框架

Mesos-DNS

Zookeeper

Mysql Proxy

Nginx Module

Redis Cluster(Gossip)

Other

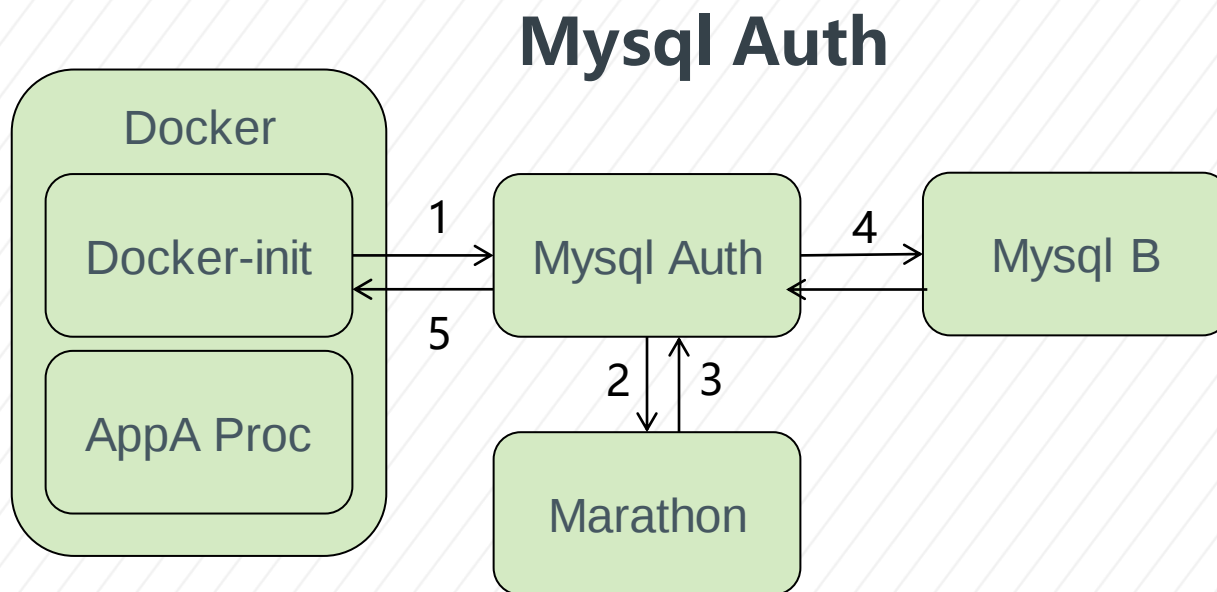




动态安全

3

动态安全



1. 声明属于AppA，申请对MysqlB的认证授权
2. 询问第三方Marathon，源IP是否属于AppA
3. 返回AppA的instance信息
4. 如果认证成功，且Mysql Auth中有MysqlB对AppA的授权，则去MysqlB去做实际授权操作
5. 同步返回Docker-init，任何失败都会让Docker-init退出，即容器退出



平台主要功能

3

基本功能

服务编配

- 发布
- 缩扩容
- 克隆迁移

容器开发机

- 持久化卷

job.data-game-xiaomi-com_service.data-game-xiaomi-com_cluster.c3_pdl.com_owt.game_cop.xiaomi

发布和回滚

实例详情

当前状态

当前实例数: 75 当前版本: e9df126a266 主机模板: 均衡型1-2

刷新

当前状态: 取消完成

发布

回滚

分支: master

版本: ec3643732ef Time:2017-05-05T09:34:29+08:00 Commit:update ports info [手工输入](#)

调试模式: ☐

发起操作

部署日志

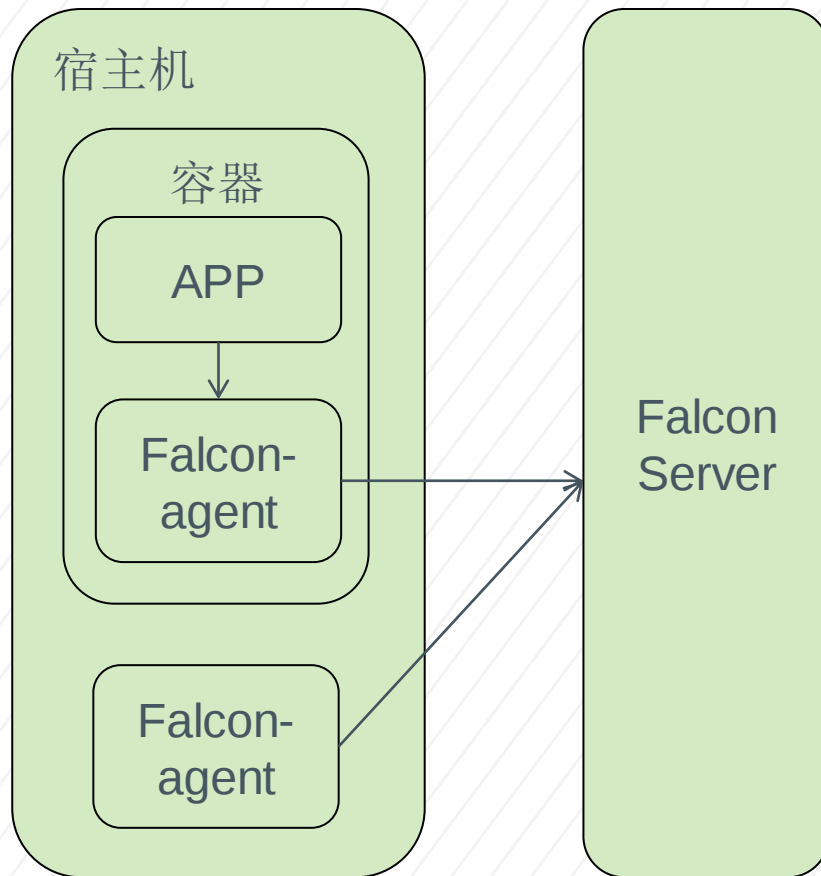
部署日志

编译日志

3

容器监控

- 宿主机和容器都有监控agent
- 默认监控策略
- 集群监控策略，触发自动伸缩



1

自动伸缩

➤ Falcon自动采集

CPU IDLE

MEM FREE

PROC QPS

PROC DELAY

➤ Hook回调

Marathon API

➤ 最短5s触发伸缩

动态调度部署系统

job.ocean-monitor-job_service.ocean-monitor-job_cluster.production-lg_pdl.ocean_owt.inf_cop.xiaomi

任务配置

运行时配置

自动伸缩

高级选项

取消

保存

伸缩指标 ?

CPU平均使用率

伸缩配置

实例数上限: 5

实例数下限: 1

实例增减步长: 1

是否通知: ☒

短信通知组 (UIC): sre_ocean

触发条件 ?

连续次数

加实例: 3

>

80

减实例: 3

<

60

2

定时触发伸缩

➤ 创建Chronos任务，调用Marathon API

动态调度部署系统

🔍 ↗ 🔔 👤 🗨 📄

主导航

仪表盘

🔌 宿主机管理

🕒 产品线仪表盘

☰ 任务管理

🗄 数据库管理

🗄 XRedis

📄 模板管理

🚩 ELB

job.ocean-monitor-job_service.ocean-monitor-job_cluster.production-lg_pdl.ocean_owt.inf_cop.xiaomi

任务配置

运行时配置

自动伸缩

高级选项

取消

保存

伸缩指标 ?

定时伸缩 ▼

定时伸缩配置 ?

+

起始时间	时间间隔	实例数	
2016-10-01 00	年 月 日 02 00	2	-
2016-10-01 00	年 月 日 07 00	4	-

Pipeline

Step	Condition	Action	Object
Step1	Branch staging change	Action1: build & ut test	Staging job
		Action2: deploy	Staging job
		Action3: bvt test	Staging job
		Action4: merge	Preview branch
Step2	Branch preview change	Action1: deploy	Preview job
		Action2: merge	Production branch
Step3	Branch production change	Action1: deploy	Production C1 job
		Action2: bvt test	Production C1 job
		Action3: deploy	Production C2 job
		Action4: bvt test	Production C2 job



服务化组件

3

ELB

➤ 自动配置域名，
按运营商自动划分
线路

➤ docker-init和旁
路模块都会动态更
新LVS配置，可重
入

动态调度部署系统

主导航

仪表盘

宿主机管理

产品线仪表盘

任务管理

数据库管理

XRedis

模板管理

ELB

新建ELB

环境

机房: c3

内外网: 外网

类型: 4层

L4配置

JOB平台: ocean

JOB绑定: job.ocean-monitor-job_

添加端口

VS: 80

RS: 8080

提交

主导航

- 仪表盘
- 🔗 宿主机管理
- 🔗 产品线仪表盘
- ☰ 任务管理
- ☰ RDS
- ☰ ElastiCache-Redis
- 🚩 ELB
- 📄 模板管理

新建数据库

取消

保存

配置

数据库名:

key

集群:

staging ▼

产品线:

browser

部门:

miui

数据库全名:

/miui/browser/key

数据库模板:

mysql-general ▼

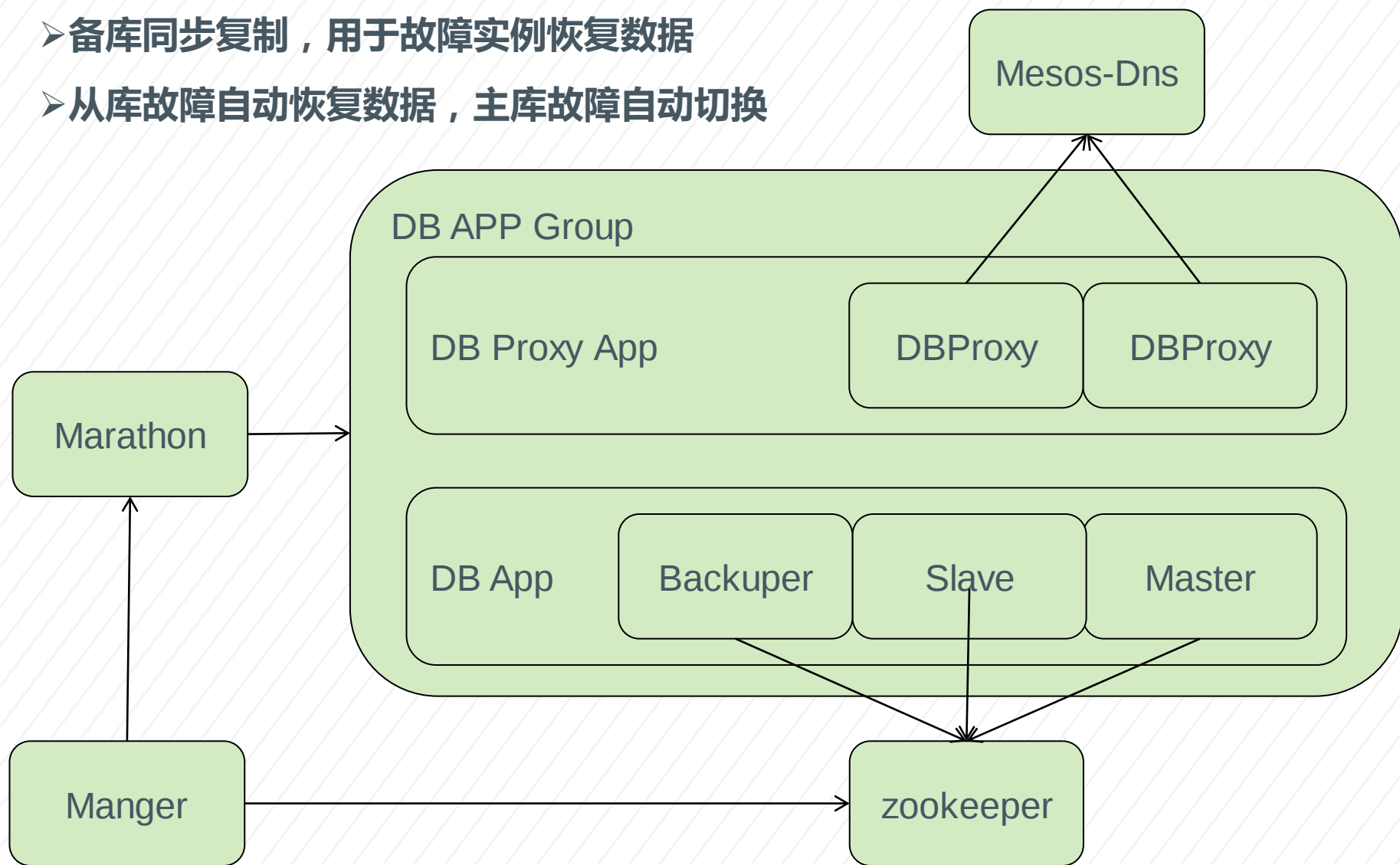
期望实例数:

3

5

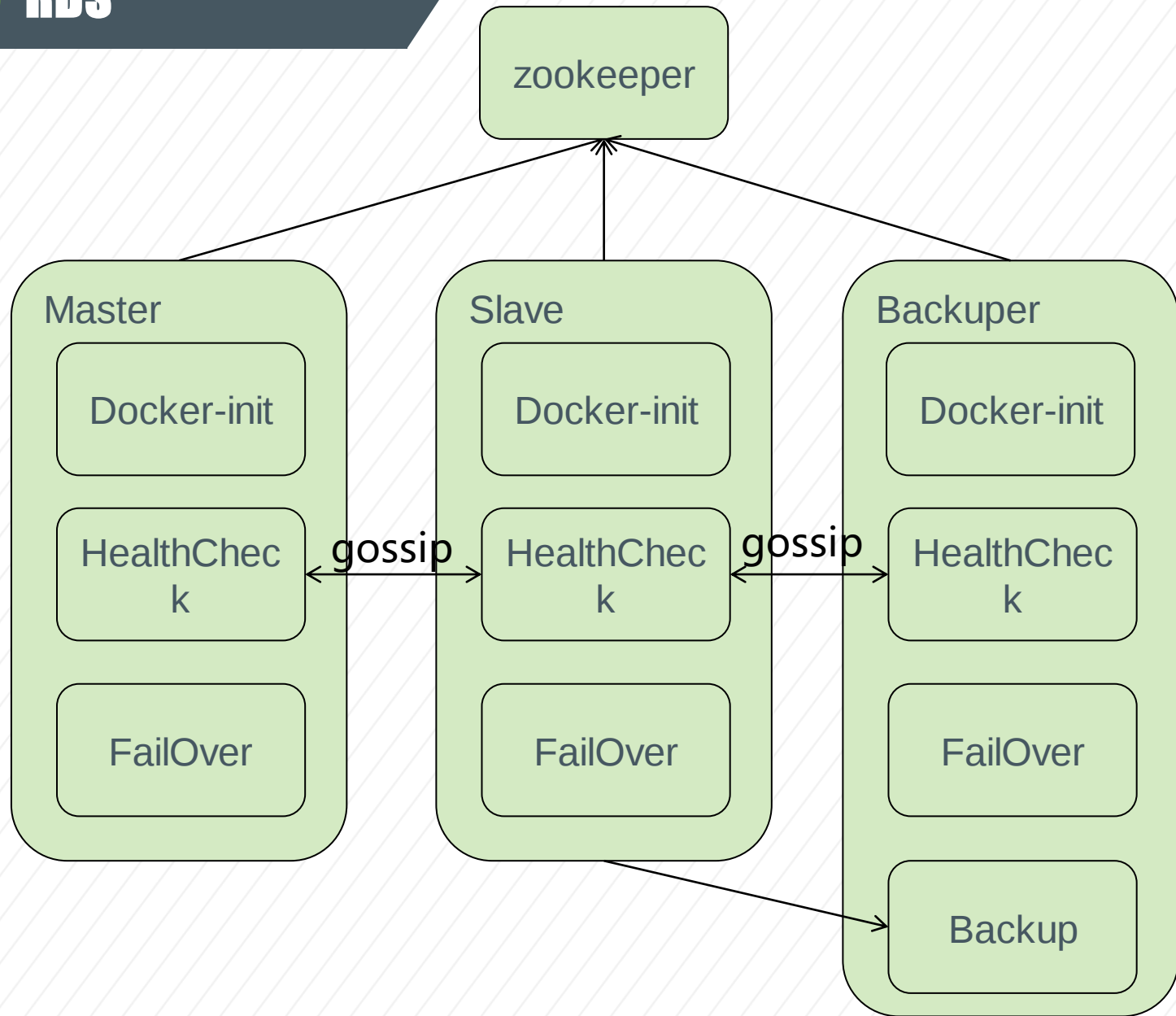
RDS

- 备库同步复制，用于故障实例恢复数据
- 从库故障自动恢复数据，主库故障自动切换



5

RDS



6

ElastiCache

➤ Redis和Memcached

➤ Redis cluster

➤ 集成对应监控插件

弹性调度平台(Ocean)

ElastiCache

主导航

- 仪表盘
- 宿主机管理
- 产品线仪表盘
- 任务管理
- RDS
- ElastiCache-Redis
- ELB
- 模板管理

申请Redis 我的申请单 待处理工单

产品线: pdl.browser_owt.miui_cop.x...

搜索 每页 10 条记录

ServiceName/cluster	实例数	内存	关联Job	集群版本	状态	运行数目	操作
_dba_aupi_micoapi	6	1536	pdl.browser_owt.miui_cop.xiaomi	2017-03-16T02:22:55.030Z	成功	6	扩容 删除 备份 重建 查看详情
_dba_redis_appstore-test	6	1536	pdl.browser_owt.miui_cop.xiaomi	2017-03-10T07:01:45.847Z	成功	6	扩容 删除 备份 重建 查看详情
_dba_redis_zhangwen-test	6	1536	pdl.browser_owt.miui_cop.xiaomi	2017-03-10T06:55:46.536Z	成功	6	扩容 删除 备份 重建 查看详情
_dba_huyu_phone-geo	10	4608	pdl.browser_owt.miui_cop.xiaomi	2017-02-28T10:51:10.604Z	成功	10	扩容 删除 备份 重建 查看详情

共 1 页 上一页 1 下一页



未来规划



未来规划

- 补充有状态组件（ZK、Kafka、Hbase）
- App间的依赖链管理和运用
- **强化故障自愈和自动容灾能力**



WECHAT

FAQ

