GOPS 2020
Shenzhen

GOPS 全球运维大会 2020

-AIOps 风向标

深圳站

中国·深圳

指导单位：

主办单位：

GOPS 2020
shenzhen

基于TARS的云原生Devops技术架构演进

利开园　　腾讯高级工程师

工作经历

腾讯Docker容器平台开发

TARS服务的基础运行环境和运营平台

腾讯TARSGo框架开发

TARS的GO语言实现

腾讯云开发业务后台开发

使用TARSGo，实现TARS+K8S的方案

GOPS 2020
shenzhen

目录

CONTENTS

01

TARS的DevOps解决方案

DevOps的核心挑战

高质运维

支持海量微服务的维护，保障服务质量，减少运维工作和复杂度

高效开发

快速实现产品能力，支持版本的高效升级

节省成本

支持弹性调度，节省服务器、网络等基础设施成本

并发能力

支持海量用户并发访问，支撑复杂的业务逻辑

TARS微服务架构主要能力

高效率开发

IDL　代码自动生成　高性能RPC　API网关

多语言　CI/CD　名字服务　测试工具

高质量运维

服务可视化　无损变更　配置管理

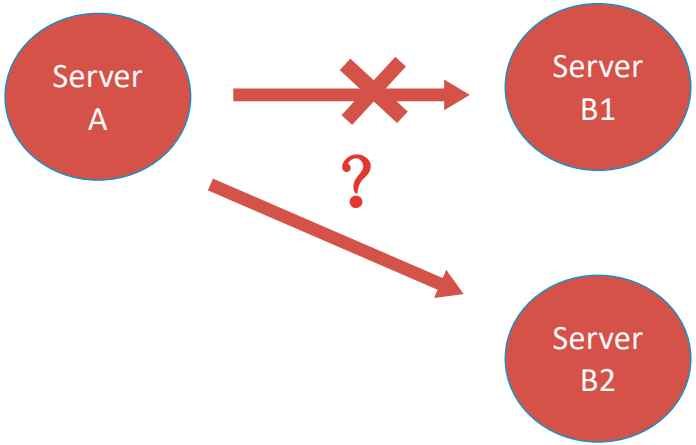
日志　监控　调用链　Set管理

过载保护　容灾容错　弹性伸缩

TARS与主流微服务方案比较

对比项	Tars	GRPC	Dubbo	备注
网络协议	udp/tcp/http2/自定义	http2	tcp/http/自定义	Tars默认提供udp/tcp支持，dubbo支持tcp/http等，grpc只支持http2
IDL	tars/pb/自定义	pb	无	IDL本身思路差不多，tars可以设置字段默认值
多语言	支持	支持	不支持	Grpc/tars都支持多语言开发，dubbo最主要支持java
RPC性能	非常高	一般	高	经过测试，tars性能高grpc 5倍，且高于dubbo
服务治理	提供整套体系	无	提供服务治理	Tars与dubbo提供整个服务治理生态体系，grpc更多的是提供思路，由其它社区或者自己构建生态。
应用规模	已大规模应用11年	社区广泛应用	社区广泛应用	Tars已经大规模应用11年，grpc和dubbo社区用得更多，时间不长。

TARS负载均衡与容灾容错



- 负载均衡
  - 轮询
  - 取模HASH
  - 一致性HASH
- 熔断
  - 屏蔽：超时/无法连接
  - 恢复：尝试连接与请求
- 过载保护
  - 限制并发数

TARS名字服务与Set模型

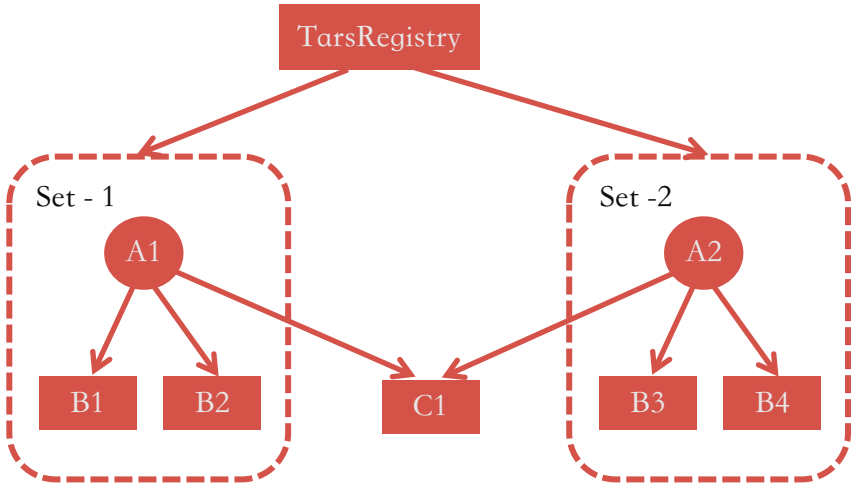
写死IP

IP配置化

服务发现

Set模型+服务发现

运维复杂度



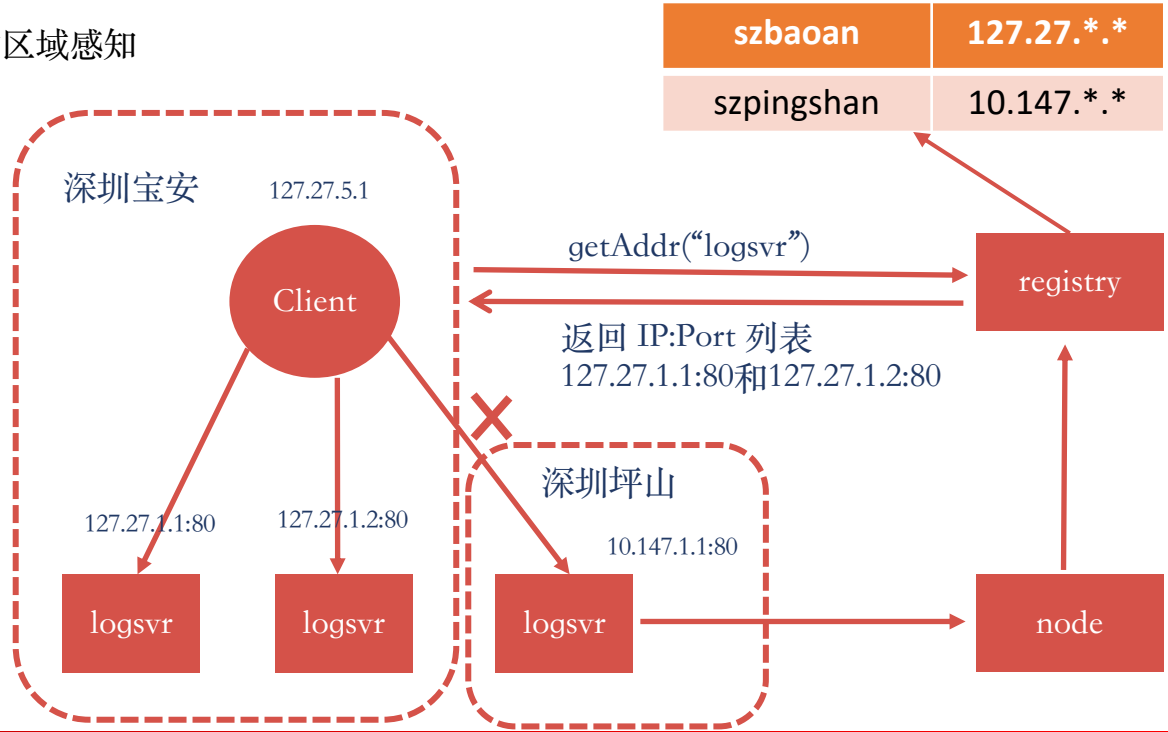

```go
func main(){
    comm := tars.NewCommunicator()
    client := &Test.Server{}
    comm.StringToProxy( servant: "Test.Server.MainObj" client)

    ret, _ := client.Sum( A: 1, B: 1 )
    _ = ret
}
```

机房容灾方案-IDC分组

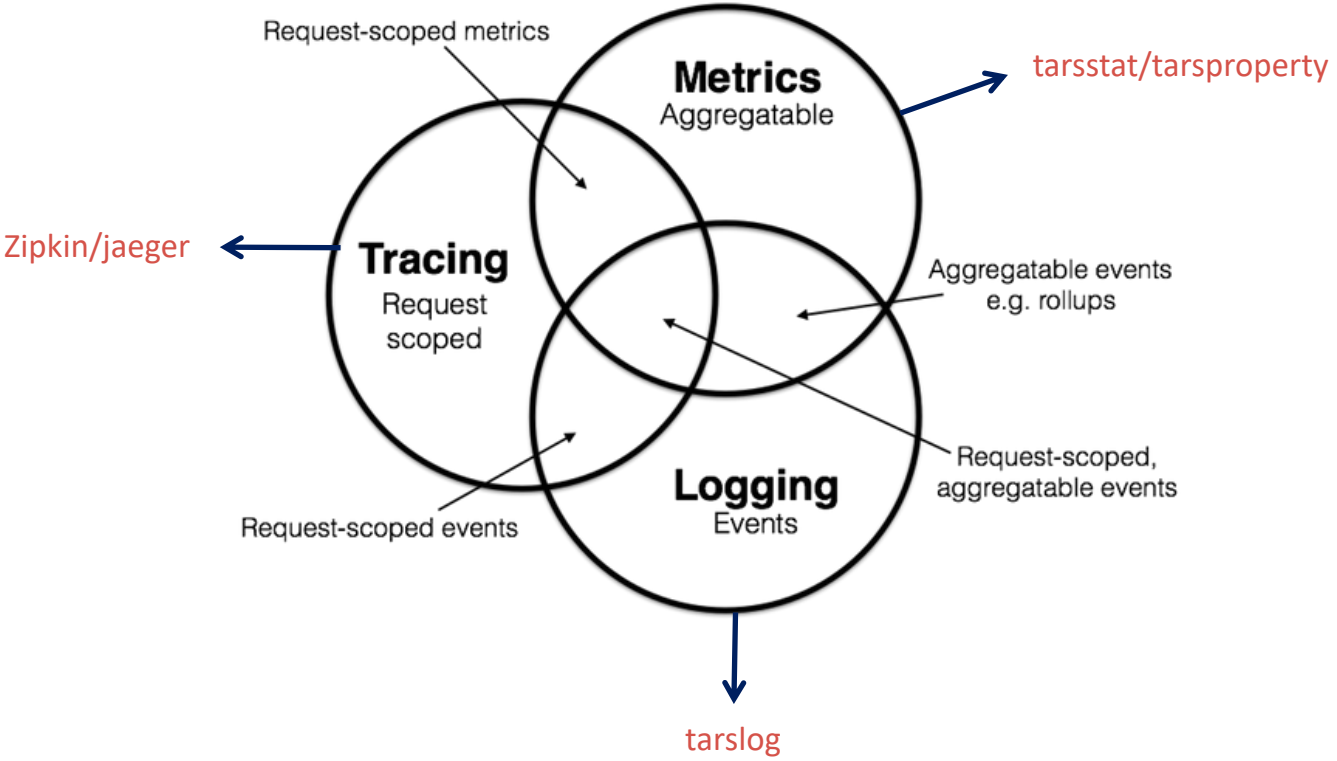
- 常规方案：网络延时高 OR 配置复杂
- TARS方案：名字服务实现自动区域感知

优势

- 运维简单
- 降低延时减少带宽消耗
- 更强的容灾能力



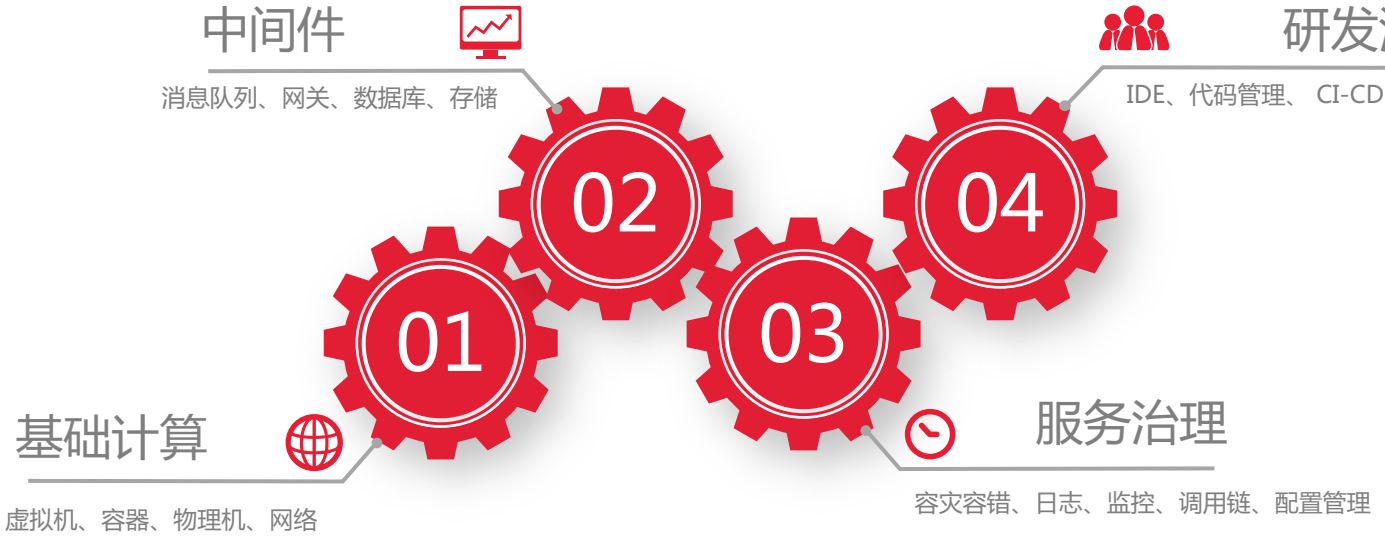
Logging/Metrics/Tracing



02

云原生趋势与TARS架构演进

云原生基础架构的演进趋势



TARS云原生方案的思考

TARS + K8S

完善服务治理能力

高效开发

高质运营

云原生服务部署能力

低成本

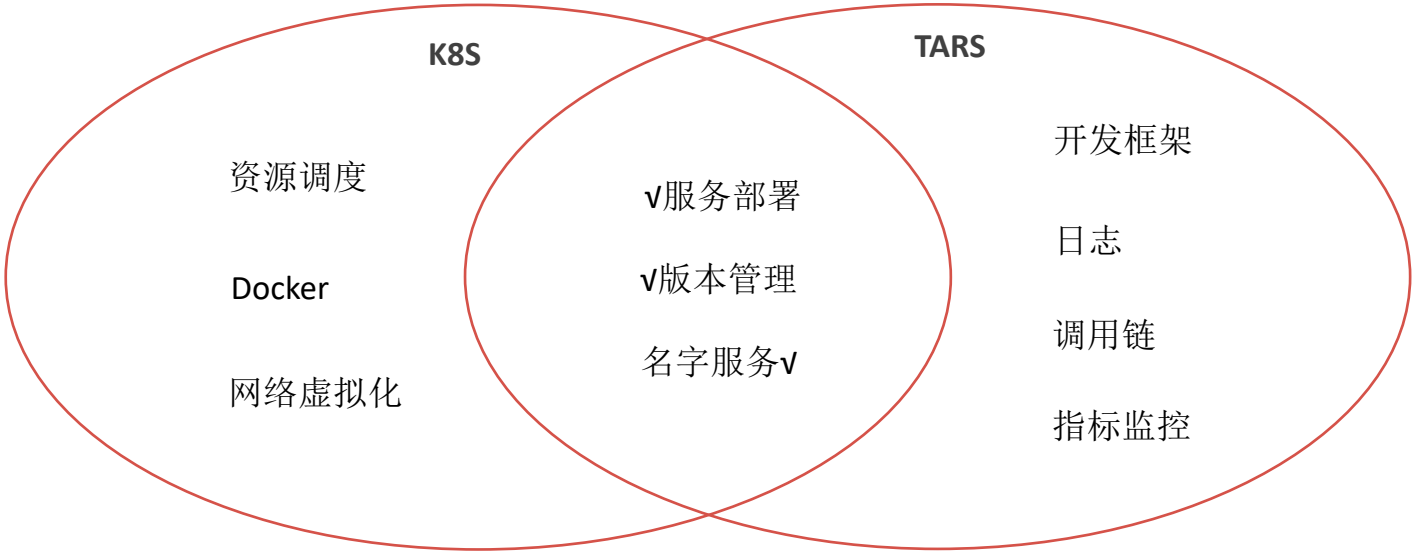
避免厂商绑定

没用使用Istio原因：

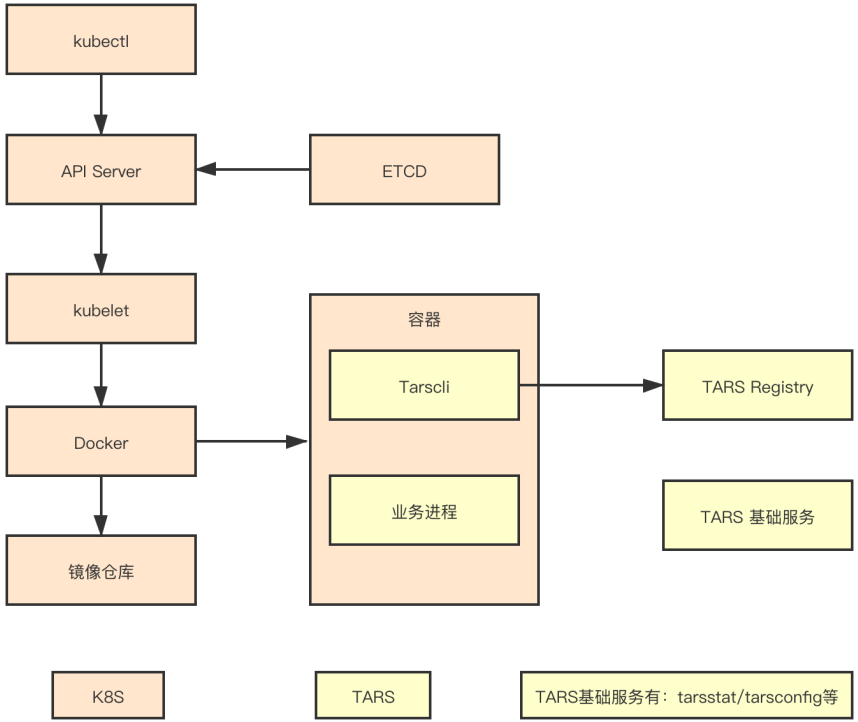
- 存量服务迁移成本
- 长连接（高性能）支持
- 客户端容错能力
- 支持set模型/IDC分组的名字服务
- 服务端口管理
- 版本稳定性

TARS+K8S整合方案

- K8S: 原生资源管理能力
- TARS: 保留开发架构及服务治理能力



TARS+K8S方案及名字服务整合



方案架构图

Pod生命周期

Tarscli: 保证TARS名字服务与容器列表的一致性

启动	Tarscli supervisor 启动TARS服务
就绪	Tarscli supervisor 将TARS服务设为active
运行	Tarscli supervisor 监控TARS服务和上报心跳
删除前	Tarscli prestop 下线TARS服务
删除	Tars Registry通过心跳保证最终一致

TARS服务部署示例及可视化管理

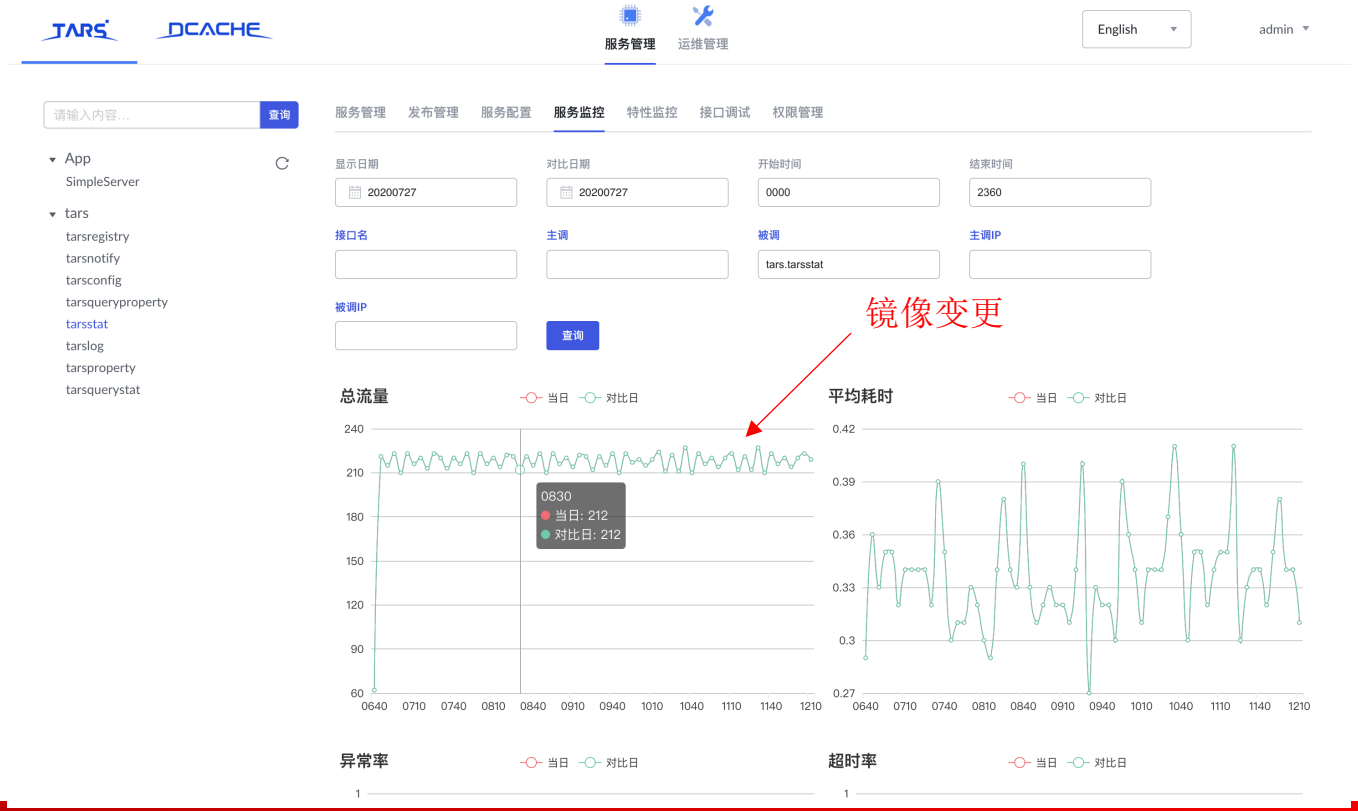
基础服务

kubectl apply -f baseserver/yaml/db_all_in_one.yaml
kubectl apply -f baseserver/yaml/registry.yaml
kubectl apply -f baseserver/yaml/tarsweb.yaml
kubectl apply –f baseserver/yaml/tarsnotify.yaml

业务服务示例

kubectl apply -f examples/simple/simpleserver.yaml
kubectl apply -f examples/TestApp/HelloGo/simpleserver.yaml

监控页面（变更过程无损）


TARS
DCACHE
服务管理
运维管理
English
admin
请输入内容...
查询
App
SimpleServer
tars
tarsregistry
tarsnotify
tarsconfig
tarsqueryproperty
tarsstat
tarslog
tarsproperty
tarsquerystat
服务管理
发布管理
服务配置
服务监控
特性监控
接口调试
权限管理
显示日期
20200727
对比日期
20200727
开始时间
0000
结束时间
2360
接口名
主调
被调
tars.tarsstat
主调IP
被调IP
查询
镜像变更
总流量
当日
对比日
240
210
180
150
120
90
60
0830
当日: 212
对比日: 212
0640 0710 0740 0810 0840 0910 0940 1010 1040 1110 1140 1210
平均耗时
当日
对比日
0.42
0.39
0.36
0.33
0.3
0.27
0640 0710 0740 0810 0840 0910 0940 1010 1040 1110 1140 1210
异常率
当日
对比日
1
超时率
当日
对比日
1

TARS未来规划

- 日志、监控、调用链与开源方案的整合
  - ES/Grafana/Jaeger
- TARS+K8S的深度整合
  - TarsWeb对接K8S，屏蔽k8s底层实现
- TARS + Service Mesh的探索
  - 将TARS核心能力整合到Sidecar中

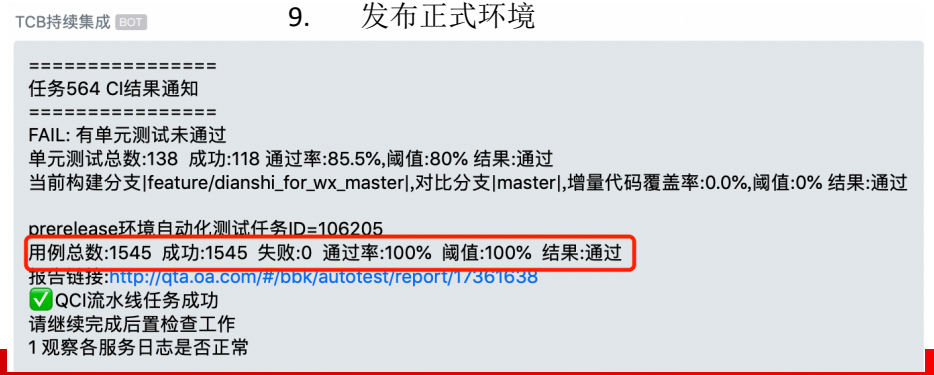
03

基于TARS的应用实践

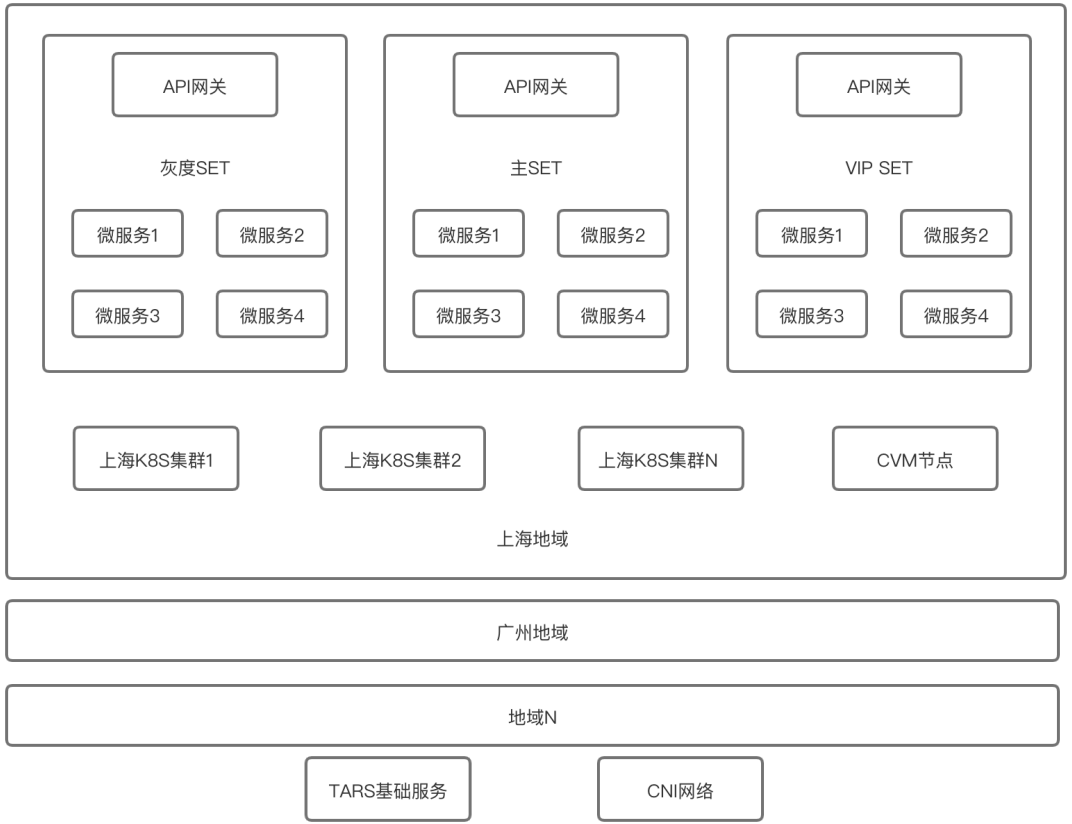
CI/CD应用实践

- CI/CD流程
  1. 提MR请求
  2. 单元测试
  3. 人工Review
  4. 合并代码
  5. 发布正式环境

- 演进方案
  1. 企业微信机器人发布
  2. 自动提MR
  3. 单元测试
  4. 模拟合并
  5. 自动发布体验、预发环境
  6. 自动化测试
  7. 人工Review
  8. 合并代码—自动生成版本
  9. 发布正式环境



基于TARS + K8S多集群混合部署架构

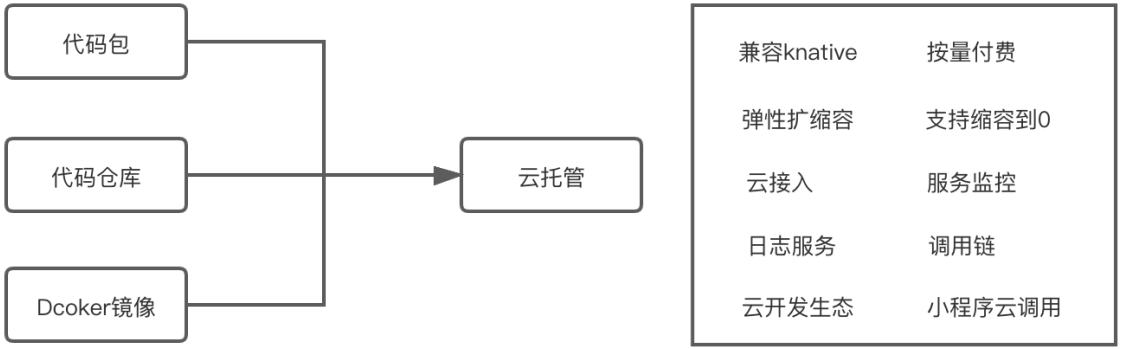


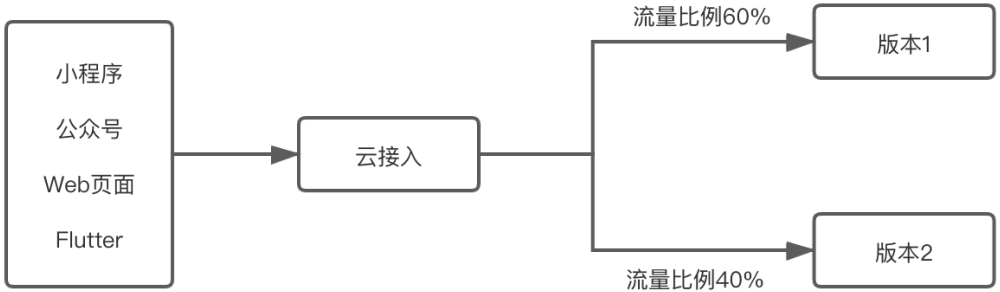
- 多Set功能
  - 隔离故障域
  - 容灾备份切换
- 部署架构
  - 多K8S集群容量水平扩容
  - 容灾
  - 支持CVM手工部署
- TARS基础服务
  - 复用同一套TARS系统
  - 统一的管理视图
- CNI网络模式
  - 实现多地域网络互通

TARS丰富的应用案例

- Tars在腾讯内部使用超过十年，并于2017年4月10日开源，开源后与业界众多企业交流，同时也得到了广泛应用。
- 开源地址： https://github.com/TarsCloud
- 2018年6月加入Linux 基金会

云托管：新一代云原生应用引擎





https://cloud.tencent.com/product/tcbr

Thanks

高效运维社区

开放运维联盟

荣誉出品

GOPS 2020
shenzhen

想第一时间看到高效运维社区的新动态吗？