





后台测试参考

张彦

2017-11-01



提纲

- 名词
- 环境搭建
- Meizu 公共组件
- 后台测试方法




名词

- 客户端
 - 即手机上的运行的应用，通过 http 协议调用后台开发的接口
- 后台开发的程序主要包括以下几种
 - Api：后台接口，供客户端调用
 - Admin：运营管理后台，用浏览器来访问
 - Task：定时任务，周期性重复执行的程序

名词

• 灰度验证

- 灰度验证是发布流程的一个步骤，发布  api（客户端接口）时该步骤是必须滴
- 实质是将线上一台机器部署新版本，并通过设置使外网用户无法访问，仅能通过内网ip访问
- 灰度验证通过，正式发布
 - 修改配置，将灰度的那台机器设置为外网用户也可以访问
 - 在其他机器上部署新版本

• 注意事项

- 灰度验证期间，做灰度的机器外网用户是无法访问的，这样剩余机器的负载就会加大，如果不能及时验证通过，当请求量比较大时就会导致外网处理能力不足，引发故障

环境搭建

- nginx
- 应用程序及其运行容器
- Mysql 数据库
- Redis 缓存



环境搭建：nginx

- Nginx是个 web 服务器
 - 提供静态资源的访问，需要配置静态资源的路径
 - 通过反向代理访问动态资源，并实现负载均衡
 - 为了更好的模拟生产环境，测试环境应该部署多个 api（客户端接口）并用 nginx 反向代理来进行路由，这样才能模拟到灰度发布中遇到的一些问题
- Nginx的安装和配置：
 - <https://wiki.meizu.com/index.php?title=Nginx%E5%AE%89%E8%A3%85%E5%92%8C%E9%85%8D%E7%BD%AE>

环境搭建-应用程序类型及运行容器

- 应用程序类型

- Api：客户端接口，运行在 jetty 上
- Admin：运营管理后台，运行在 jetty 上
- Task：定时任务及 mq 消费者，运行在 launcher 上
- Kiev：基于 kiev 的服务化接口，目前没有开发这类程序

- 一些资料

- Jetty 安装及应用程序部署

https://wiki.meizu.com/index.php?title=FlymeTV_manage%E9%83%A8%E7%BD%B2

- Task 环境搭建及部署：

https://wiki.meizu.com/index.php?title=FlymeTV_task%E9%83%A8%E7%BD%B2

环境搭建：mysql 数据库

- 安装 mysql

https://wiki.meizu.com/index.php?title=Mysql_%E5%AE%89%E8%A3%85

环境搭建：redis 缓存

- 搭建高可用的 redis 集群

[https://wiki.meizu.com/index.php?title=Redis %E9%9B%86%E7%BE%A4%E6%90%AD%E5%BB%BA](https://wiki.meizu.com/index.php?title=Redis_%E9%9B%86%E7%BE%A4%E6%90%AD%E5%BB%BA)

- 查看 redis 数据
 - RedisDesktopManager

Meizu 公共组件

- 配置中心
- zookeeper
- 分布式调度平台
- 图片服务器
- 分布式文件系统 FastDFS
- Metaq 消息中间件



公共组件：配置中心

- 访问地址 config.meizu.com
- 修改了配置中心的配置项一般需要重启应用程序

公共组件：zookeeper

- Zookeeper 简单理解，就是一个存放配置的数据库
 - Zk 服务器一般是集群部署，会有多个ip地址和端口
 - 数据在 zk 里是树形存放的，需要正确指定数据存放的路径，例如 /redis/proxy/hotapps
 - Redis/metaq/调度平台/kiev 需要配置对应的 zk
- 各个环境 zk 使用说明
<https://wiki.meizu.com/index.php?title=%E5%90%84%E4%B8%AA%E7%8E%AF%E5%A2%83zookeeper%E4%BD%BF%E7%94%A8%E8%AF%B4%E6%98%8E>

公共组件：分布式调度平台

- 调度平台管理定时任务
- Meizu 有多个调度平台，目前我们用的是 ElasticJob1.0.8
 - <https://wiki.meizu.com/index.php?title=%E6%96%B0%E7%9A%84%E5%88%86%E5%B8%83%E5%BC%8F%E8%B0%83%E5%BA%A6%E5%B9%B3%E5%8F%B0elastic-job%E4%BD%BF%E7%94%A8%E8%AF%B4%E6%98%8E-1.0.8>

公共组件：图片服务器

- 保存图片文件，并提供一个 url 来访问该图片
- 配置项
 - 业务名
 - 访问密码

公共组件：分布式文件系统 FastDFS

- 可以保存各种小文件，不限于图片
- 配置项
 - 不需要做任何配置



公共组件：Metaq 消息中间件

- 参考资料

https://wiki.meizu.com/index.php?title=Metaq_%E5%AE%9E%E6%88%98%E7%BB%8F%E9%AA%8C

- 配置项

- Zk 服务器地址
- 命名空间

- 测试要点

- 对于广播消息要测试是否所有消费者都接收到消息

后台测试方法

- Api 的测试
- Task 的测试
- Consumer 的测试



后台测试方法：api

- 根据接口文档，构造访问要测试的接口的url
- 访问该url
- 分析返回的结果
 - 应答码（一般是 value）是否 200
 - 若应答码是 200，返回的其他字段是否符合接口文档的定义
- 查看后台日志，有没有记录异常
 - 日志位置：/data/log/jetty，异常记录在xxx-error.log文件

后台测试方法：task

- Task 没有可供调用的接口，也没有界面
- 要测试 task 是否正常运行
 - 了解 task 的功能，一般 task 是对数据库或缓存进行操作，可以在 task 执行前清理目标数据库/缓存，然后在 task 执行后看数据库或缓存是否被更新
 - 查看日志
 - 日志位置 /data/log/task
 - xxx-server.log，一般会在 task 执行时输出 task 启动的日志
 - Xxx-error.log，在 task 执行是若记录了异常，那表示 task 执行出错

后台测试方法：consumer

- Consumer，是消息的消费者，接收到消息后会执行特定的任务，api，admin，task 里都可能有consumer
- 测试申请里应该对消费者的功能进行说明
- 一般消息都有一个生产者
- 示例
 - 在运营后台点击【刷新缓存】的按钮，运营后台就可能会发送一个消息
 - 该消息有可能发送给多个机器，甚至是多个机房
 - 每个收到该消息的消费者都会执行相同的功能，在本例子里就是刷新缓存