***极智平台架构设计及部署技术文档***

***v1.0***

***提供方：北京众策科技咨询有限公司***

***联系人：白克宇***

***联系电话：18601240956***

***邮箱：baikeyu@grouptactic.com***

**目录**

1. 平台介绍 1

1.1 主要功能列表 1

2. 技术设计 2

2.1 架构设计介绍 2

2.2 应用架构图 3

2.3 基础服务模块列表 3

2.4 公共模块列表 4

2.5 业务服务模块列表 4

2.6 应用模块列表 4

3. 系统部署 5

3.1 硬件列表（供参考） 5

3.2 部署操作 5

3.2.1 安装依赖环境 5

3.2.2 参数配置 6

3.2.3 加载并启动基础服务镜像 6

3.2.4 构建应用镜像 7

3.2.5 运行应用 8

# 平台介绍

极智平台是基于微信第三方开放平台，提供企业服务号的营销，管理及运营的一站式SaaS平台。

* 一键接入，即插即用，开发零成本，微信营销热启动
* 接入即可拥有多种微信运营工具，操作简单易上手
* 灵活组合营销场景进行推广，马上实现快速引流增粉，三级裂变式传播
* 集会员招募、会员活动拓展、会员精准营销、会员积分营销等智能立体化模块
* 解决传统会员管理软件不能解决的来店营销和离店营销的困境帮助商家彻底的升级了营销思路和营销手段
* 树立自己品牌最直接、投入产出比最好的解决方案，真真实实帮助企业获取精准客户、促成交易、提升会员活跃度
* 增加会员粘性提升品牌忠诚度，促进会员二次消费

## 主要功能列表

|  |  |
| --- | --- |
| 内容 | 描述 |
| 微信自动回复 | 基于微信自动回复接口实现，主要包括三种模式   1. 服务号粉丝关注后自动回复 2. 粉丝在公众号内发送消息后自动回复 3. 关键词出发后自动回复 |
| 微信自定义菜单 | 配置服务号3 x 5菜单，提供默认链接及消息推送，同时支持自定义外链的配置。 |
| 模板消息 | 对系统内相应消息的推送模板。 |
| 游戏设置 | 服务号内相关游戏的配置内容，目前只支持转盘抽奖。 |
| 营销海报 | 会员传播链推荐海报的相关配置。 |
| 渠道二维码 | 渠道或者门店所使用的关注二维码配置，包括设置以及相应的统计分析内容。 |
| 会员列表 | 所用服务号粉丝以及相关信息的列表展示 |
| 会员关系 | 以关系图的形式展现会员的推荐关系 |
| 分析报表下载 | 相关生成的报表下载统一入口 |
| 公众号管理 | 管理相关企业以及注册的服务号 |
| 员工管理 | 可登陆平台账号的员工管理 |
| 账户管理 | 企业账号相关充值以及消费明细的管理 |
| 美克美家相关报表查询 | 内嵌美克美家定制化内容 |

# 技术设计

极智平台技术方向主要依托于微信开发平台机制（参考：[微信开放平台](https://open.weixin.qq.com/)），代理进行服务号的管理以及接口调用（相关微信接口参考：[公共平台开发文档](https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1445241432&token=&lang=zh_CN)）。整体架构采用微服务架构体系，共4个基础服务模块，10个业务服务模块以及前段应用。

## 架构设计介绍

整体架构采用微服务方案，核心框架为Spring Cloud，配合Docker实现DevOps的整合。

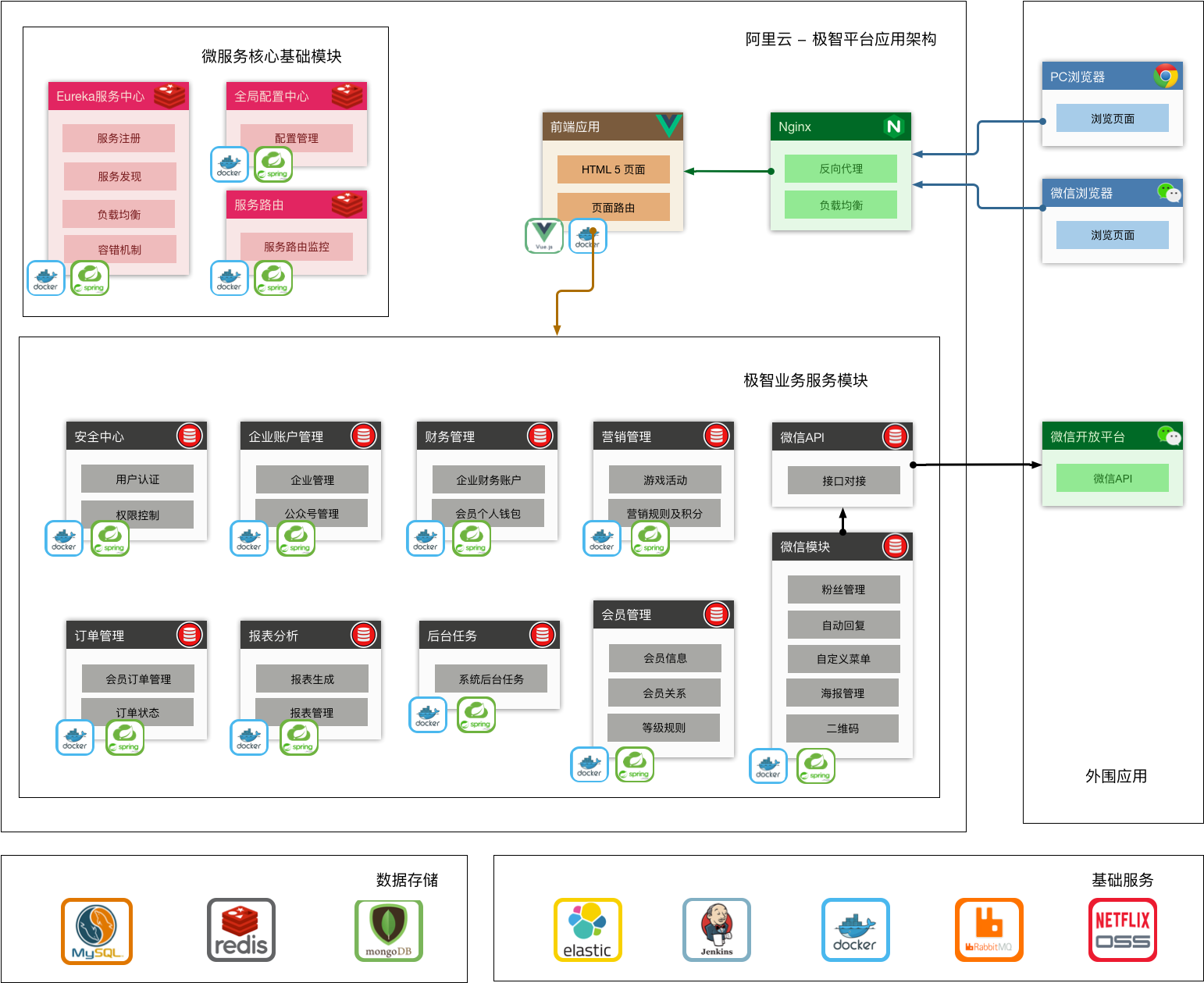
* 服务器入口使用Nginx进行HTTP的反向代理。
* 前端（PC及移动端）应用封装为可单独部署容器，同时配合服务中心Eureka进行服务接口的负载均衡，容错及接口调用。
* 服务间接口调用采用同步REST API以及异步消息队列机制，保证核心服务模块的稳定以及可扩展性。
* 在持久化层，使用MySQL作为传统业务数据库，同时配合Redis实现相应的缓存策略以及高并发的控制机制。
* 在数据聚合以及响应式计算方向，使用MongoDB为主要解决方案，以提高数据的可扩展性，响应事件的持久化，数据存储与读取速度。
* 在运维方向，使用Jenkins，docker-compose作为主要的容器部署工具。同时配合ELK（ElasticSearch，Logstash，Kibana）进行日志的收集以及监控。

技术架构主要围绕Spring技术栈为主，前端使用Vue.js进行前后端分离策略

* Spring Boot为主要应用框架
* Spring Cloud作为主要服务框架，使用了config，eureka，zipkin，feign，hystrix进行微服务架构的扩展
* Spring Security进行相关的认证及权限控制
* Spring Messaging和Spring Rabbit作为异步消息队列AMQP框架支持
* 持久层采用Hibernate JPA和Spring Data作为ORM和持久层框架
* Reactor作为部分响应式计算框架支持
* FreeMarker作为服务器端页面渲染模板
* Vue.js作为主要的双向数据绑定，单页面路由前端框架，保证前后端分离架构
* 数据分析，报表类视图通过接口调用数据，并使用echarts.js进行渲染。

整体部署方案以阿里云为主，同时与其他外围系统（微信，商城）对接。

## 应用架构图



## 基础服务模块列表

|  |  |
| --- | --- |
| 模块名称 | 描述 |
| grouptactic-admin | 基于Spring Boot Admin配置的服务模块状态监控应用（参考[Spring Boot Admin Reference Guide](http://codecentric.github.io/spring-boot-admin/1.4.6/)） |
| grouptactic-config | 基于Spring Cloud Config配置的系统全局配置中心，所有微服务模块通过此入口获取相应环境的配置参数（参考[Spring Cloud Config](https://cloud.spring.io/spring-cloud-config/)）。 |
| grouptacitc-docker | 相关基础服务镜像（ELK，RabbitMQ，Redis）Dockerfile和docker-compose.yml配置文件 |
| grouptactic-eureka | 基于Spring Netflix OSS Eureka配置，主要完成服务注册及服务解析工作（参考[Spring Cloud Netflix](https://cloud.spring.io/spring-cloud-netflix/)） |
| grouptactic-repository | 相关配置文件 |
| grouptacic-zipkin | 基于Spring Cloud Sleuth配置，基于Span机制追踪服务模块内的请求路由情况（参考[Spring Cloud Sleuth](https://cloud.spring.io/spring-cloud-sleuth/)） |

## 公共模块列表

|  |  |
| --- | --- |
| 模块名称 | 描述 |
| grouptactic-commons | 项目共享包，包括请求回复，异常，分页等共享类，常量，枚举以及日期，路径，JSON等常用工具类 |
| grouptactic-facade | UI应用的façade封装模块，进行相应的数据聚合及过滤 |
| grouptactic-echarts-api | 将服务器端数据进行聚合，封装成为前端echarts所需数据模型 |
| grouptacitc-wechat-api | 微信接口的二次封装 |
| grouptactic-youzan-api | 有赞接口的二次封装 |

## 业务服务模块列表

|  |  |
| --- | --- |
| 模块名称 | 描述 |
| grouptactic-account | 企业账户管理模块，包括企业的注册及信息查询，公众号与企业关联，员工信息管理等。 |
| grouptactic-distributor | 分销商管理，分销规则及返佣模块（开发中） |
| grouptacitc-finance | 企业资金账户的管理，以及会员个人钱包的管理 |
| grouptactic-marketing | 相应营销管理内容模块，包括营销游戏活动，促销管理，会员积分等 |
| grouptactic-member | 核心会员管理模块，会员信息的管理，会员等级及相关规则配置，会员关系的建立与追踪 |
| grouptactic-order | 订单中心，负责会员订单信息的管理以及状态变更 |
| grouptactic-reporting | 报表分析模块，包括前台实时数据分析以及Excel报表的生成 |
| grouptactic-security | 安全中心，PC端用户认证以及权限管理 |
| grouptactic-task | 系统全局后台任务管理模块 |
| grouptactic-wechat | 微信模块，进行相关公众号菜单和消息的管理，同时负责接收微信推送事件，收集公众号粉丝信息 |
| grouptactic-youzan | 有赞商城的店铺管理和商品管理 |

## 应用模块列表

|  |  |
| --- | --- |
| 模块名称 | 描述 |
| grouptactic-ui | PC端以及移动端相关路由和页面 |

# 系统部署

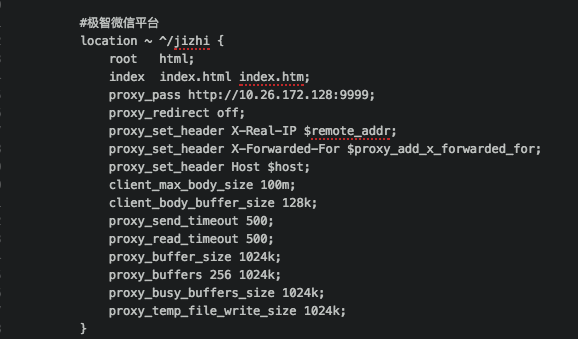
## 硬件列表（供参考）

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 描述 | 配置 | 价格 |
| 阿里云ECS服务器 | 生产应用服务器，进行主要服务模块的部署 | CPU：8核/内存：16G/硬盘：40G/带宽：5M/操作系统：CentOS 7.2 | 9333/年 |
| 阿里云ECS服务器 | 运维服务器，包括相关CI工具（Jenkins，Maven，GIT），Docker容器管理以及ELK环境 | CPU：8核/内存：16G/硬盘：40G/带宽：5M/操作系统：CentOS 7.2 | 9333/年 |
| 阿里云RDS数据库服务器 | 数据库服务器 | CPU：2核/内存：4G/硬盘：100G | 3468/年 |
| 阿里云MongoDB数据库服务器 | 数据库服务器 | CPU：2核/内存：4G/硬盘：100G | 8404/年 |

## 部署操作

### 安装依赖环境

1. 安装并配置JDK 8，参考官方文档
2. 安装并配置Maven，参考官方文档：<https://maven.apache.org/>
3. 安装并配置Git，参考官方文档
4. 安装并配置Docker及docker-compose，参考官方文档：https://www.docker.com/
5. 安装并配置Nginx，加入以下配置内容

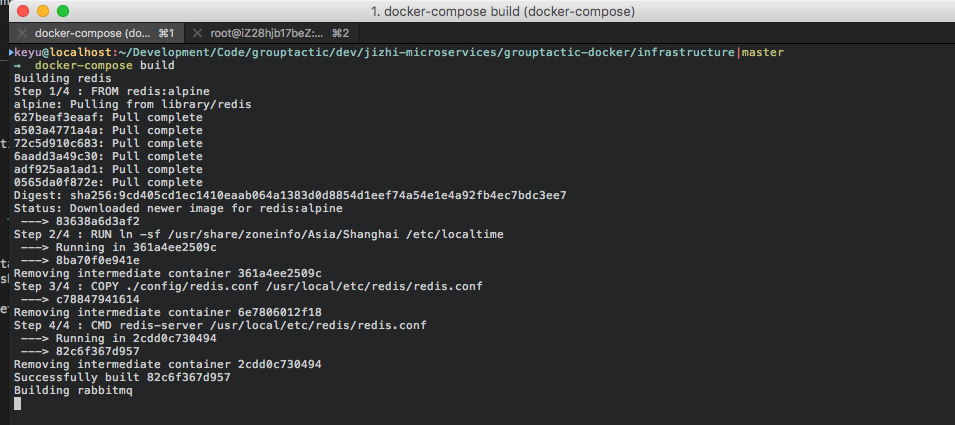


### 参数配置

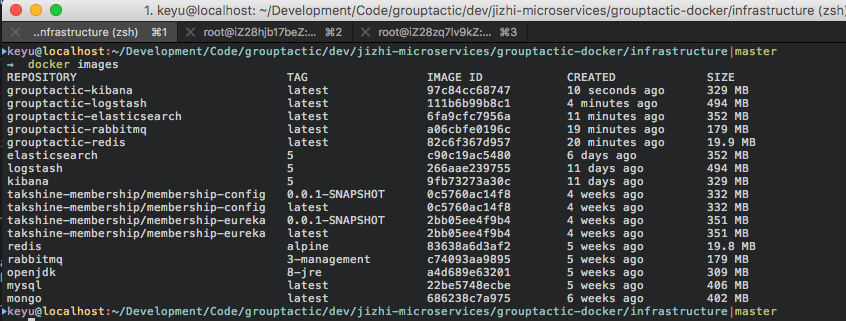
按需修改相关配置文件grouptactic-repository/application.yml，以及对应服务模块下的resources/application.yml和resources/bootstrap.yml

### 加载并启动基础服务镜像

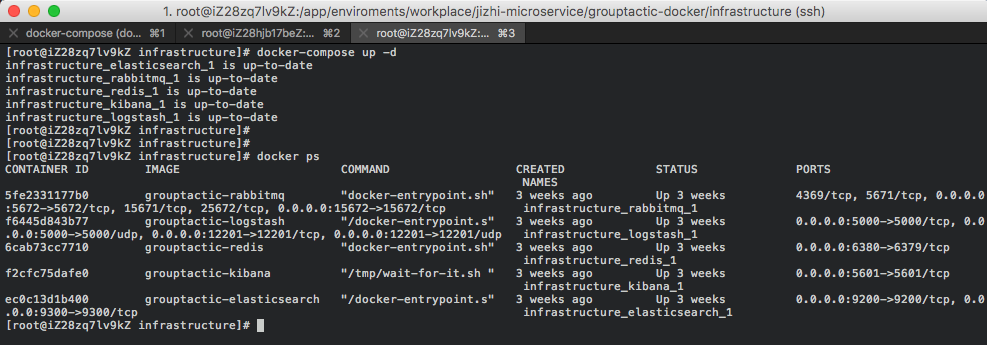
在grouptactic-docker/infrastructure目录下运行docker-compose build命令，开始加载并构建镜像



镜像构建完毕后使用docker images查看已构建完成的镜像

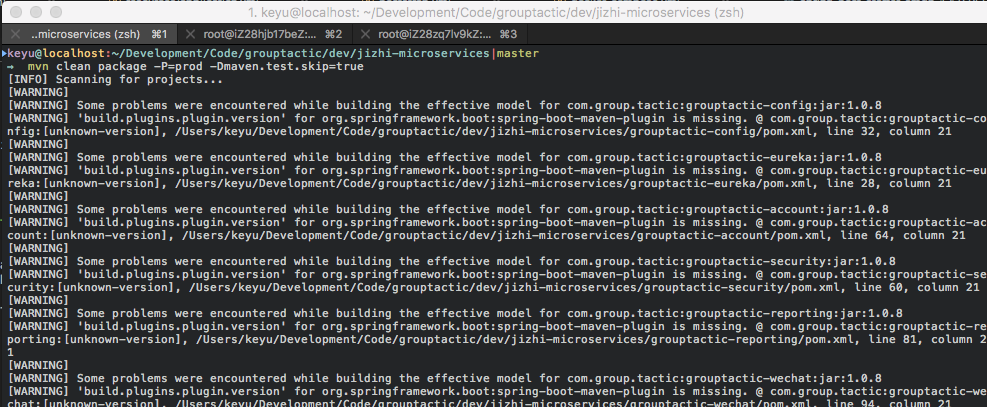


运行docker-compose up –d启动所有所需容器，之后运行docker ps查看已启动的容器

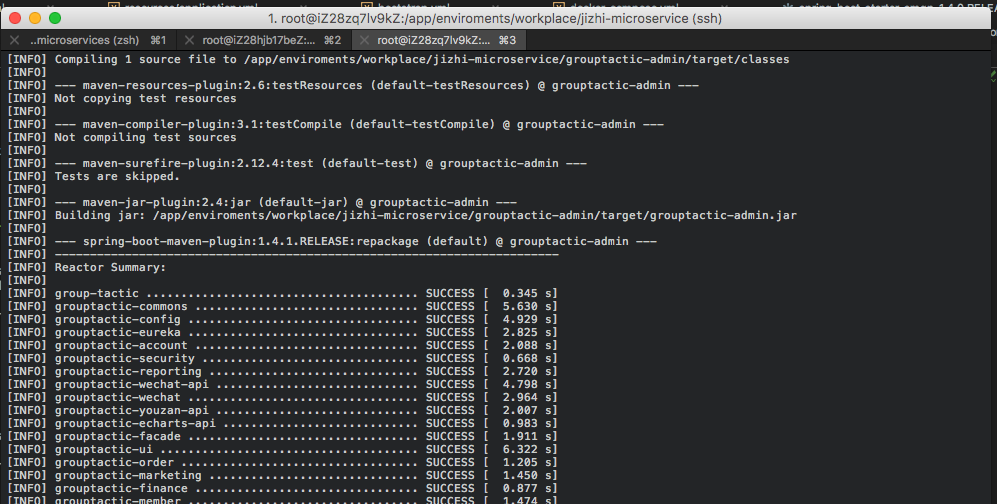


### 构建应用镜像

进入项目主目录，用Maven进行打包，mvn clean package –P=prod –Dmaven.test.skip=true。打包之后针对每个模块将生成独立的jar文件，这些jar包已经可以在没有docker容器封装的情况下独立运行

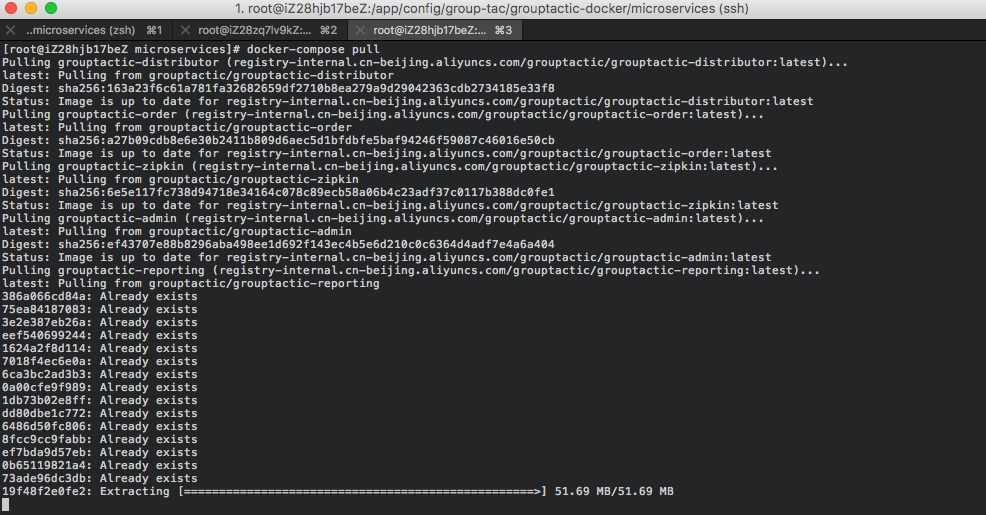


如需打包为docker镜像，需输入一下命令：mvn docker:build -P=prod -DpushImage -pl grouptactic-wechat。此命令将grouptactic-wechat模块打包为docker镜像，同时推送至镜像服务器。如果需要打包多个镜像，需要-pl参数后跟随多个模块名称，同时用逗号进行分隔。

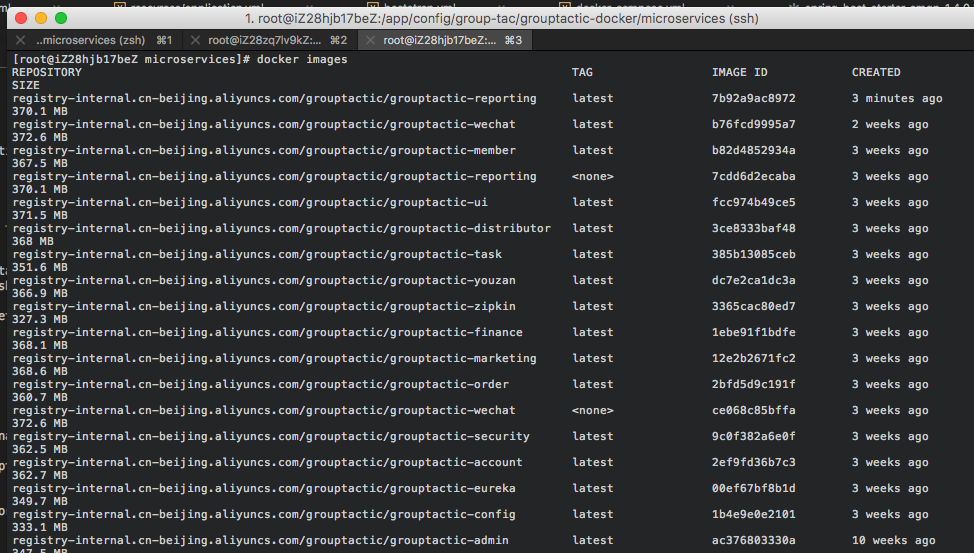


### 运行应用

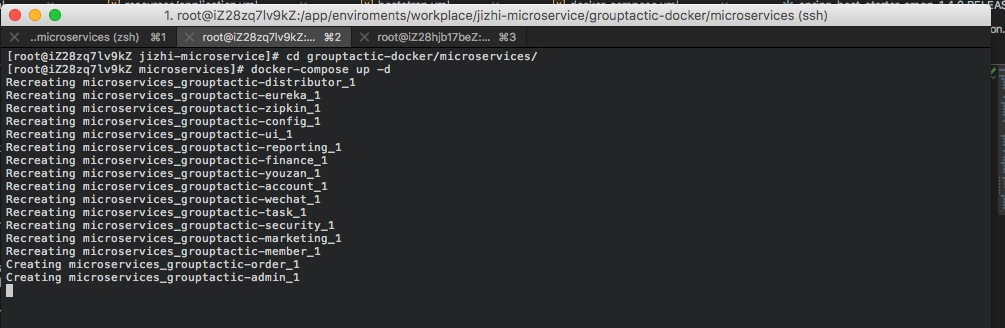
在应用服务器上，在grouptactic-docker/microservices目录下运行docker-compose pull从镜像服务器拉取最新的版本的镜像文件



使用docker images命令查看镜像版本



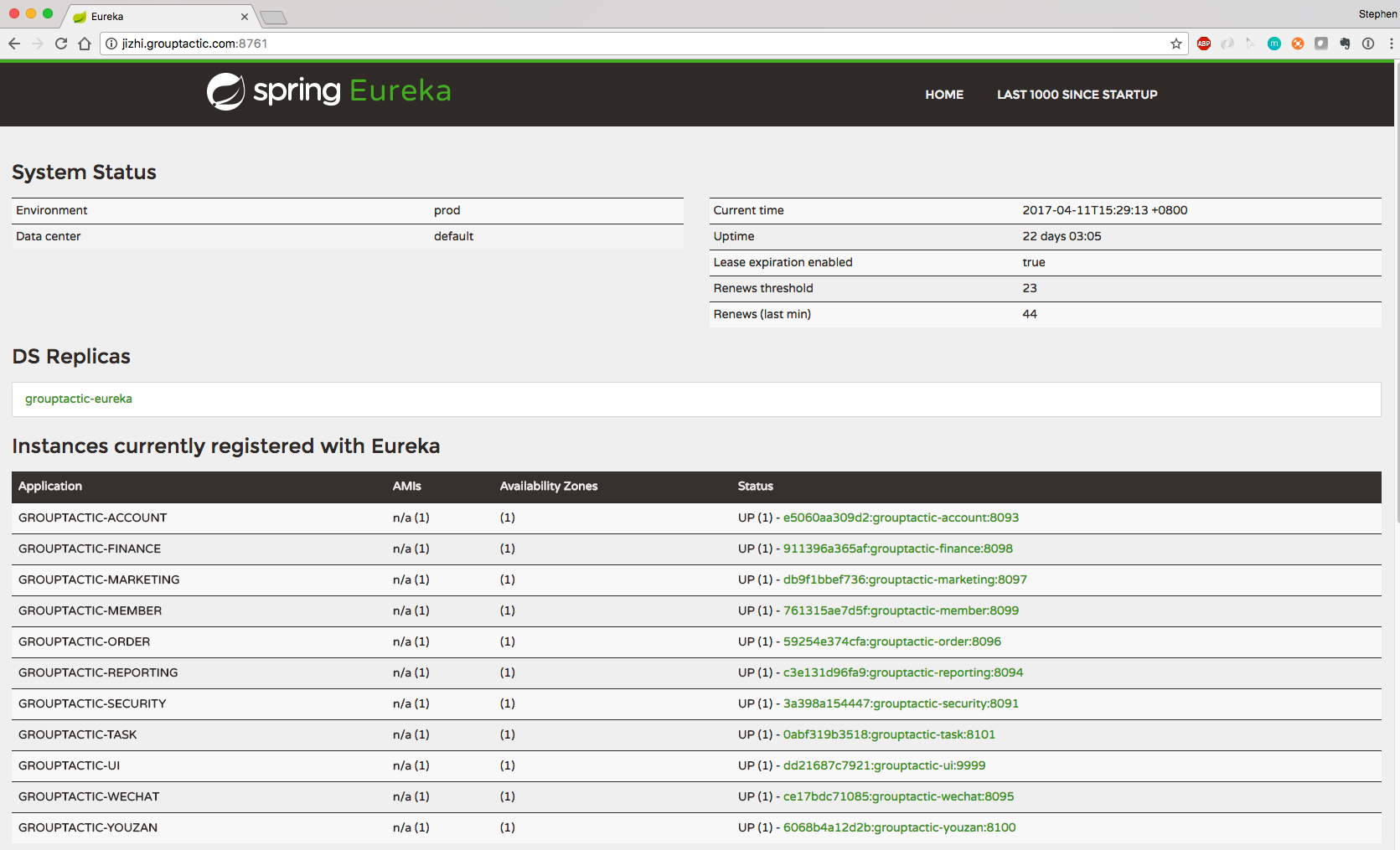
使用docker-compose up –d启动所有服务模块（docker-compose down 为停止所有服务模块）



进入Kibana页面，查看应用启动情况



同时进入Eureka管理页面，查看服务注册情况



这样服务整体启动完毕