

# 服务器端运营手册

## 服务器需求

2核4G内存以上机型即可

## 部署方案

### 依赖组件：

标准服务依赖于一下组件

1. Node.js: v0.10.x
2. redis: v2.8.x
3. python: >2.6
4. unzip
5. g++, gcc

## 部署操作

1. 首先确保机器上已经正确安装了Node.js, redis, unzip
2. 解压文件，运行安装包中的install.py, 例如 `./install.py $PARENT_PATH`，会在 PARENT\_PATH之下生成解压当前的Chameleon Backend服务器
3. 如果redis安装在另外的机器上，那么请修改\$PARENT\_PATH/config/svr.conf里面 pendingOrderStoreCfg的配置
4. 到安装目录之下，运行 `./chameleon.js start`。验证是否启动完成，可以使用 `./chameleon.js active` 来验证是否启动正常，正常启动的话会打印running

## 对外开放端口

Chameleon的backend会启动三个http服务，包括渠道服务器回调HTTP服务，响应鉴权HTTP服务，管理端HTTP服务，分别对应端口 80（生产环境必须），8081（可配），8083（可配）。80端口必须对外开放到外网。8081和8083必须不能开放到外网，因为Chameleon backend设计时候假定是内网安全环境。

## 维护手册

Chameleon的backend端使用chameleon.js这个工具来管理，它的位置在backend的安装目录之下。以下说明假设Chameleon Backend的安装目录的父目录是 CHAMELEON\_DIR

### 添加或者更新产品

由客户端工具可以生成对应产品的配置包，假设为ucloud.zip，将这个压缩包上传到服务器之后，运行 `./chameleon.js up-product /path/to/ucloud.zip` 即可为当前的服务器上的Chameleon添加一个产品。

## 添加SDK的plugin module

在获取到plugin module的压缩包之后，假设为anzhi.zip, 将这个压缩包上传到服务器之后，运行 `"/code>./chameleon.js add-plugin file:///path/to/anzhi.zip'` 即可为当前的服务器上的Chameleon添加新的SDK服务插件。

## 停止服务

使用 `./chameleon.js stop` 来停止本机的服务

## 事件监测

Chameleon Backend本身会统计10分钟内的登录和充值数据，利用chameleon.js工具，可以用来获取这个信息 `./chameleon.js monitor event`

## 服务器状况监控

Chameleon Backend有两种方式监控服务状态

1. 使用 `./chameleon.js active` 来监测服务器是否正常运行，如果正常的话，stdout会输出running，否则会输出dead。也可以通过-e来设置一旦服务宕掉之后的响应脚本，例如 `./chameleon.js active -e 'YOUR_SCRIPT'`
2. 使用 `./chameleon.js monitor status` ,可以监控各个敏感部件的情况

## 高可用

Chameleon Backend自身设计的是无状态的，因此可以随意的多份服务热备。程序core掉之后也可以即时拉起，不会造成状态的异常。但是Chameleon Backend依赖于外部的存储

（redis）来维护状态，因此如果需要高可用，可以对redis进行高可用的主从热备，这个业内已经有成熟的方案。或者使用高可用的，兼容redis协议的kv云存储方案。