**消息推送使用说明文档**

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | AMessage | 项目代号 |  |
| 文档名称 | 消息推送说明文档 | 文档版本 | 1.0 |
| 起草 | 高明亮 | 起草日期 | 2019.3.1 |
| 审批 |  | 审批日期 |  |

**版权声明：**

本文档的版权归北赞同科技有限公司所有，未经授权不得复制和引用。本文档若有改动，恕不另行通知。

**本书约定：**

**1、通用格式约定**

宋体：正文采用小四宋体。

楷体：警告、说明等提示等内容一律用楷体，并且在内容前后增加线条与

正文隔离。

屏幕显示描述：对于来自屏幕的显示信息，采用如下格式：英文Courier New ，中文宋体，文字大小为小四号；信息中夹杂的用户从终端输入的信息，使用加粗字体表示。

**2、各类标志**

本书还采用各种醒目标志来表示在操作过程中应该特别注意的地方，这些标志意义如下：

：警告、危险、提醒操作中应注意的事项。

：说明、注意、提醒操作中应注意的事项。

**变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 修改内容 | 页号 | 修改人 | 完成日期 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |

注：对该文件内容增加、删除或修改均需填写此变更记录，详细记载变更信息，以保证其可追溯性。

**目录**

[1 消息推送流程说明 5](#_Toc2352131)

[1.1 架构说明 5](#_Toc2352132)

[2 目录结构 5](#_Toc2352133)

[3 在ACaas中部署步骤 6](#_Toc2352134)

[3.1 准备资源 6](#_Toc2352135)

[3.1.1 准备amessage.tar.gz需要的资源 6](#_Toc2352136)

[3.1.2 将amessage文件夹打包成tar.gz 6](#_Toc2352137)

[3.2 登录ACaaS管理端 6](#_Toc2352138)

[3.2.1 新增模板 6](#_Toc2352139)

[3.3 镜像构建 6](#_Toc2352140)

[3.4 部署服务 7](#_Toc2352141)

[4 物理机部署步骤 7](#_Toc2352142)

[4.1 准备amessage需要的资源 7](#_Toc2352143)

[4.2 修改启动脚本 7](#_Toc2352144)

[4.3 执行启动脚本即可 7](#_Toc2352145)

[5 配置文件参数相关说明 7](#_Toc2352146)

[5.1 application.yml 7](#_Toc2352147)

[5.2 log4j2-spring.xml 8](#_Toc2352148)

[6消息发送方式和消息体说明 9](#_Toc2352149)

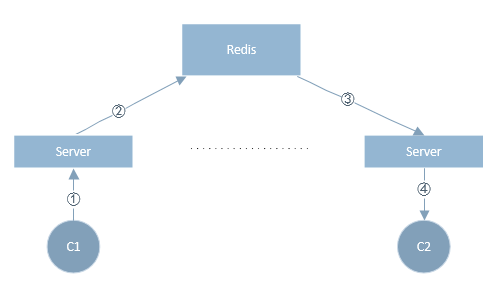
[客户端注册 9](#_Toc2352150)

[客户端主动推送消息 9](#_Toc2352151)

[服务端主动推送信息 9](#_Toc2352152)

[服务端清除客户端信息 10](#_Toc2352153)

# 消息推送流程说明



## 架构说明

* 方案描述

C1向C2发送消息

1、在C1和C2都在线的情况下，C1通过websocket向server推送消息。

2、服务端接收到消息后，将消息发布到redis上。

3、订阅者（server）接到消息。

4、Server将消息发给C2。

# 目录结构

|  |  |
| --- | --- |
| 目录 | 目录含义 |
| config | 消息推送服务器使用到的配置 |
| log | 程序运行的日志 |
| micro\_product\_message-xxx.jar | 程序主jar包 |
| startup.sh | 程序启动的sh脚本 |

# 在ACaas中部署步骤

## 准备资源

### 准备amessage.tar.gz需要的资源

将目录结构中的相关文件及资源放到amessage文件夹下，根据实际情况修改config下的application.yml和log4j2-spring.xml

### 将amessage文件夹打包成tar.gz

## 登录ACaaS管理端

### 新增模板

点击技术栈—技术栈模板，选择增加模板

模板内容为：

|  |
| --- |
| FROM 119.23.20.22:31000/wangyang/os-jdk-sftp-ssh:7.0.2  ADD amessage.tar.gz /  ENV JAVA\_HOME /usr/java/jdk1.8.0\_181  ENV PATH $JAVA\_HOME/bin:$PATH  ENV CLASSPATH .:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar  ENV LANG en\_US.UTF-8  WORKDIR /amessage  EXPOSE 8080  CMD sh startup.sh |

简单说明：（具体含义百度搜索dockfile命令介绍）

* FROM 为基础镜像的地址，根据实际情况修改
* ADD后为实际的tar.gz名称
* WORKDIR后一般为上述tar.gz中的文件夹名字

EXPOSE端口号为amessage.tar.gz中的yml中定义的server.port端口号。

## 镜像构建

* 点击技术栈—镜像构建，选择“上传镜像源文件”，选择1.2中准备的tar.gz文件
* 选择2.1中的镜像模板
* 输入镜像名和版本号
* 点击确认构建镜像

## 部署服务

点击管控中心—部署应用，选择私有仓库中的对应的镜像名和版本号，点击下一步，填入相应的内容即可。

# 物理机部署步骤

## 准备amessage需要的资源

将目录结构中的相关文件及资源放到服务器实际文件夹下，确保使用用户对该文件夹有读写及执行权限，根据实际情况修改config下的application.yml和log4j2-spring.xml。

## 修改启动脚本

修改启动脚本为：

java -jar micro\_product\_message-1.0.0.jar &

可根据实际情况添加jvm相关参数信息，如-Xms、-Xmx等

## 执行启动脚本即可

sh startup.sh

# 配置文件参数相关说明

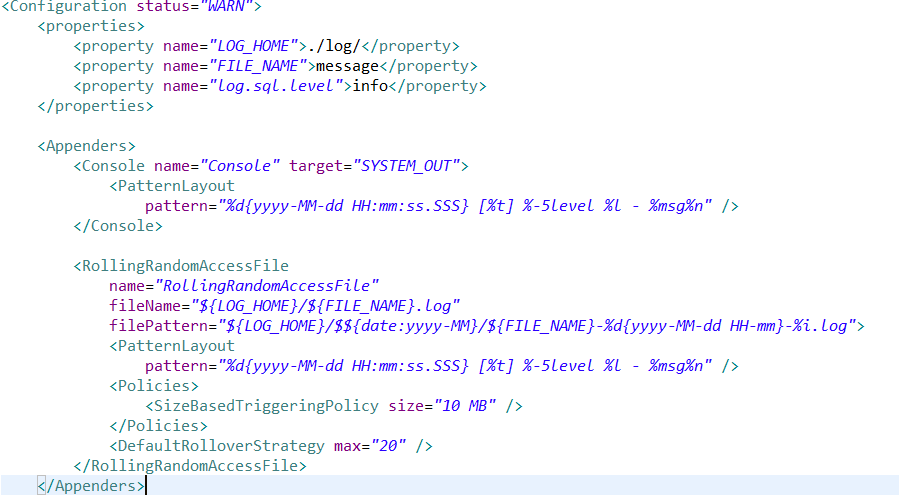
## application.yml

|  |  |
| --- | --- |
| 配置 | 配置说明 |
| server.port | 服务启动监听的端口，默认8080 |
| Spring.redis.mode | Redis模式（stand-alone 、 master-slave 、 cluster） |
| Spring.redis.sentinel | Redis哨兵配置 |
| Spring.redis.sentinel.master | 主节点名称 |
| Spring.redis.sentinel.nodes | 哨兵节点信息 |
| Spring.redis.redisAddr | 单例Redis服务地址 |
| Spring.redis.redisAuth | Redis密码 |
| Spring.redis.maxWaitMills | 最长等待时间 |
| Spring.redis.cluster | Redis集群配置 |
| Spring.redis.cluster.nodes | Redis 集群节点信息 |
| Jedis.pool.max-active | 连接池最大连接数（使用负值表示没有限制） |
| Jedis.pool.max-idle | 连接池中的最大空闲连接 |
| Jedis.pool.min-idle | 连接池中的最小空闲连接 |
| message.channel | 消息频道 |
| message.overtime | 消息推送超时时间 |
| logging.config | log4j2配置文件所在的文件目录 |

## log4j2-spring.xml

log4j2的详细配置说明参考官网：<http://logging.apache.org/log4j/2.x/>

当前log4j2-spring.xml配置文件用到的配置常量如下：



表示的含义分别为：

|  |  |
| --- | --- |
| 配置常量名 | 含义 |
| LOG\_HOME | 日志文件输出的目录（相对运行jar） |
| FILE\_NAME | 日志文件名称 |
| log.sql.level | 日志输出级别 |
| SizeBasedTriggeringPolicy size | 每个日志文件最大大小 |
| DefaultRolloverStrategy max="20" | 最多存在的日志文件数量 |
| PatternLayout pattern | 日志输出格式 |

# 6消息发送方式和消息体说明

**推送方式**：单点发送、网点群发、全行群发。

6.1客户端注册

向服务端发送自身信息存储到内存，固定四个参数，四个参数都必须传值：

**例 {"id":"oid1","type":"init","data":{"app":"qq","corporation":"001" ,"branch":"boss1"}}**

* **Id** : 本身oid；
* **Type** : init，代表要注册信息；
* **data** : 存放跟消息推送方式有关的数据app，branch。

**branch**： 网点名称；

**corporation：**法人

**app** ： 应用类型。

6.2服务端主动推送信息

**发送方法**：http://localhost:8080/pushMsgToChannel

调用pushMsgToChannel接口进行推送，需要传递5个入参：

* **type** ：消息类型
* **scope** ：消息推送的类型，即范围，有3种取值，0代表单点推送，1代表网点群发，2代表全行群发；
* **destId** ：推送的目的oid；
* **branch**：网点；
* **corporation：**法人
* **app** ：应用。
* **content**：推送的消息内容。

单点推送时5个入参必须都传值，网点群发时目的destId可不传，全行群发destId和branch可不传。

**例**：

{

"type":"wspush",

"scope":"0",

"destId":"oid1",

"branch":"boss1",

"corporation": "001",

"app":["qq"],

"content":"1send2"

}

6.3客户端主动推送消息

需要指定应用（app），然后再分为单点发送、网点群发、全行群发的方式。固定两个参数，两个参数必须传值：

* **type** ：wspush，代表要发送信息；
* **data** ：存放跟消息推送方式有关的数据，data中单点发送的时候包括5个参数：

**scope** ：消息推送的类型，即范围，有3种取值，0代表单点推送，1代表网点群发，2代表全行群发；

**destId** ：推送的目的

**branch**：网点；

**corporation:** 法人

**app** ：应用，传递的是要推送到的目的应用类型；

**content**：推送的消息内容。

**例**：

**{"type":"wspush","data":{"scope":"0","destId":"oid1","corporation":"001" ,"branch":"boss1","app":"qq","content":"1send2"}}**

1.单点传送时data中5个参数必须都传值。

2.网点群发时data中目的destId可不传。

3.全行群发data中destId和branch可不传。

6.4服务端清除客户端信息

若客户端断开，在回调方法中把客户端信息从redis中清除，关闭session连接，避免后续消息推送报错。