swarm-client 使用文档

前言：目前该版本属于内部试用alpha版本，后续main.go调用的包会成为go的静态库或动态库，调用方式可能会有些许的变化，所以请将消息处理函数封装于自己的逻辑块中，避免后续版本更新的移植问题。

1：下载swarmclient 项目

git clone <http://10.121.140.202/cloud/swarmclient.git>

2：向士杰申请证书，证书文件名为config，该证书用于接入消息系统。

3：修改项目 swarmclient目录下的/cmd/main,go，该文件是一个简单的使用demo，命名客户端的clientID（clientID的命名规则：必须包含字符 - or \_）和在相应的位置填写消息处理逻辑。

\*\*\*消息系统基本概念和demo程序的介绍

概念介绍：

clientID（客户端的唯一标识）:每启动一个客户端就需要分配一个clientid，该标识是客户端的唯一标识。用于记录消息的状态。

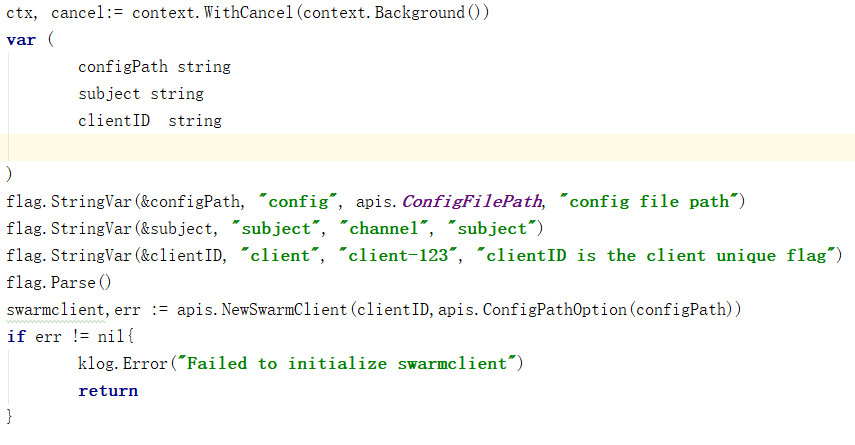
pub（publish）： 向主题发布消息

sub(subscribe)：接收主题的消息，pub与sub成对出现，且必须订阅同意主题。

Subject(主题)：pub与sub的作用对象都是主题。

demo示例讲解：

一：初始化



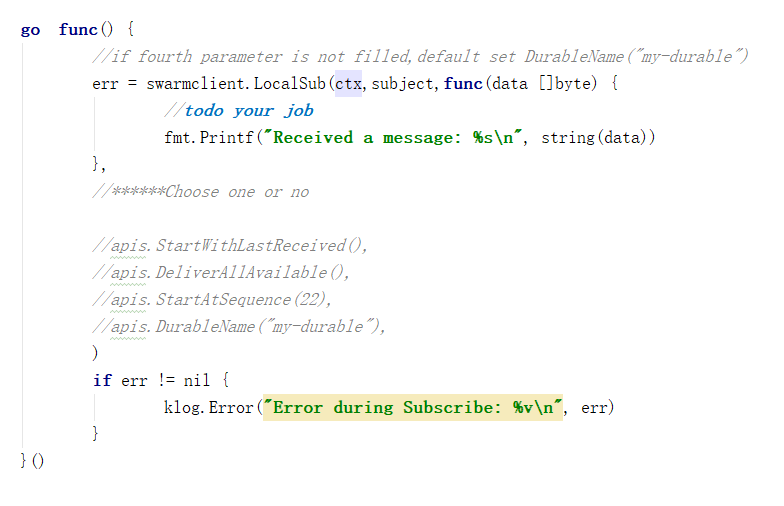
ctx, cancel:= context.WithCancel(context.Background())

ctx用于传入sub中，用户接收用户停止订阅。

cancel调用该函数通知订阅停止。

swarmclient,err := apis.NewSwarmClient(clientID,apis.ConfigPathOption(configPath))

NewSwarmClient 至少传递一个clientid参数，可以选择传入config文件的路径。

二：订阅

err = swarmclient.LocalSub(ctx,subject,**func**(data []byte) {  
 *//****todo your job*** fmt.Printf(**"Received a message: %s\n"**, string(data))  
},

订阅函数有local和cloud开头的两种函数，local边缘本地通信使用该类函数，数据会持久化在本地。cloud云边通信和网络不互通的边边通信使用该类函数，数据会持久化在云端。

LocalSub：本地订阅函数

该函数是一个阻塞函数所以需要用goroutine来运行。

参数说明

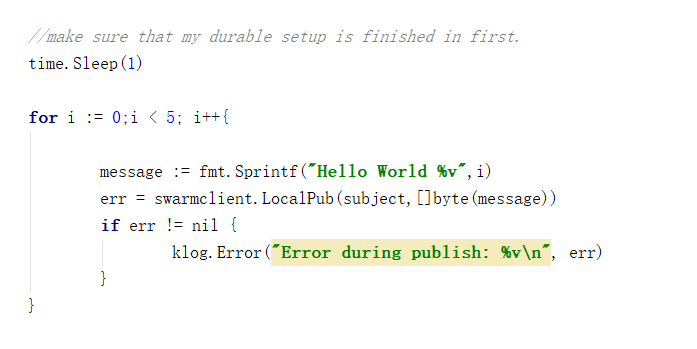
1：ctx第一步实例化的context对象，用户通知sub停止订阅

2：Subject 订阅的主题(必须与pub的主题一致)

**3：func**(data []byte) {  
 *//****todo your job***  
} 消息处理函数，data代表获取到的消息内容。在这个函数添加你的接收消息逻辑即可。

4：可选参数（不填默认会设置*apis.DurableName("my-durable")*）关于该参数的说明可以查看readme

三：发布消息



time.Sleep(1)

demo程序中的sub第四个参数每天默认使用了durable，该参数有个问题需要注意，就是只有服务端对这个durable创建完成以后才能记录主题的消息状态，也就意味着sub只能收取到durable建立完成以后消息，之前发送的消息无法接收。这里等待时间是为了确保durable建立完成以后再pub消息。

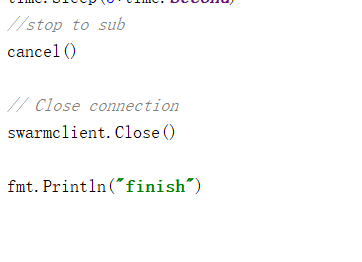
err = swarmclient.LocalPub(subject,[]byte(message))

参数说明：

subject（主题）：与通信的sub保持一致

message： pub的消息内容

四：停止订阅与关闭连接



Cancle():第一步实例化context对象的第二个参数，用于停止订阅

swarmclient.Close()：关闭tcp连接。

4：将士杰分发的证书放置于/etc/swarm目录下

5：如果消息系统与启动程序不在同一个主机（如若在同一主机跳过该步骤），需要在启动程序所在主机配置dns解析，配置内容为 消息系统所在主机Ip（这个ip需要士杰来提供） 对应dns为edge.local

6：启动程序即可