Apollo是一个代理服务器，在amq的基础上发展，支持stomp，amqp，mqtt，openwire，ssl，websockets，Apollo提供后台管理页面，方便开发者管理和调试

emq 百万级mqtt消息服务器，支持大规模链接和分布式集群，发布订阅模式的开源mqtt消息服务器。

HIveMQ：一个企业级的MQTT代理。。。。。

Mosquitto：实现了消息推送协议mqttv3.1的开源消息代理软件，提供轻量级、支持可发布订阅的消息推送模式。

----------------

Apollo使用完全不同的线程和消息调度框架来实现。

理解：Apollo是一个框架，需要创建是指的MQTT服务器

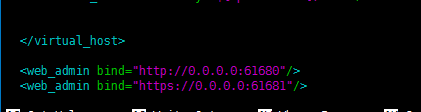
1.创建服务器 ./Apollo create （MQTT服务器）

在当前目录下创建一个服务器，

2.创建服务器后需要启动服务器

命令：./服务器名称 /bin/apollo-broker run 运行创建的服务器

会显示mqtt要连接的端口和管理页面端口



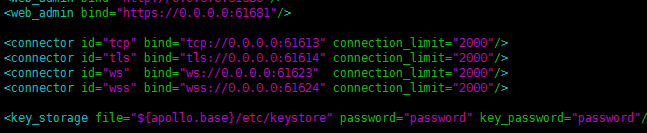
设置此处修改ip的访问

http://127.0.0.1:61680/

https://127.0.0.1:61681/

用户名密码：admin/password

代码的链接地址



java代码链接mqtt

创建mqttclient对象

MqttClient client=new MqttClient(“apollo对外暴露的地址”，“随机的客户端id”);

设置mqtt的链接属性

client.connect(链接属性对象);

MqttconnectionOptions options =new MqttconnectionOptions();

设置链接的地址

options.setServerUris(new String[]{服务器的集群}); 自己理解

设置是否清空session

options.setcleansession(true) 清空session 则会清楚客户端的连接记录

设置连接的用户名和密码

options.setUserName(); options.setPassword(password.toCharArray)

设置连接超时时间

options.setConnectionTimeout();

设置会话心跳时间, 每隔多长时间向客户端发送消息,判断是否在线,但并没有重连机制.

发送消息

client.publish("发送消息的topic",MQTTMessage);topic是可以在java端创建的,

上述是生产者

============================

下面是消费者

连接上mqtt之apollo服务器,将mq中的消息读取并存储到MongoDB库中