首先这个需求里api需要返回和接受json格式的数据

所以首先考虑使用restful 的webservice , 要使用restful的webservice 可以考虑如下几种框架

Jersey + jetty 比较轻量, 适合部署在docker容器下

Springboot + springmvc 比较重, 不过spring 支持的比较全面,如果需要使用spring其他功能可以优先考虑

测试环境下这两种方式都支持脱离tomcat之类的J2EE容器直接启动, 结合docker容器的话可以用来做微服务架构, 性能上jetty要稍微优于spring,不过spring提供并发,线程池,批处理等其他功能,在项目中要考虑实际需求

这里出于快速搭建模型的考虑(因为需求里说可以提供docker容器) , 所以我面对的需求是

轻量+快速构建+只需要提供例子 所以选择jersey不代表在实际项目中我推荐这种方式

服务可以分布在多个docker容器中, 也可以使用一个容器, docker本身不推荐使用bash 作为容器的启动进程 (会出现僵尸进程问题) , 所以这里我们遵循 一个容器, 一个服务的原则, 当然这个服务不是webservice服务, 而是一个容器一个进程

使用maven创建一个简单的jersey+jetty环境

因为数据模型只有一个mail地址代表一个user, 所以User类里只需要两个字段就可以了

1. Mail address
2. Friends ( list )

这样我们得到一个 多 对 多 的数据模型

关于订阅更新的function, 我个人意见是不推荐使用记录 “我订阅了谁” , 而应该采用回调的方式, “我被谁订阅了” , 实际上是 “观察者”模式, 当一个邮件被更新之后, 应该是这个被更新的邮件通过回调方法取得”谁订阅了我”, 然后逐个发送通知 , 而不是倒过来 当”一个用户被更新之后”, “通过遍历知道谁订阅了这个用户”

然后我们封装这个应用到容器, 这里我使用docker

[root@vagrant ~]# systemctl start docker

[root@vagrant ~]# docker run -t -i --name hfm -p 8080:8080 docker.io/centos /bin/bash

[root@83b4294f69d5 /]# yum install yum -y install java-1.8.0-openjdk\*

这里考虑是直接提供docker Makefile 还是提供ansible 的playbook

Makefile 的话不需要再安装ansible到宿主机, 也不需要先配置好docker, 如果是已经存在的docker集群的话,可以考虑直接使用playbook, 比makefile的通用性要好很多

这里为了方便我只是使用插件将依赖的jar包都拷贝出来

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-dependency-plugin</artifactId>

<executions>

<execution>

<id>copy-dependencies</id>

<phase>prepare-package</phase>

<goals>

<goal>copy-dependencies</goal>

</goals>

<configuration>

<outputDirectory>C:/hfm/jars</outputDirectory>

<overWriteReleases>false</overWriteReleases>

<overWriteSnapshots>false</overWriteSnapshots>

<overWriteIfNewer>true</overWriteIfNewer>

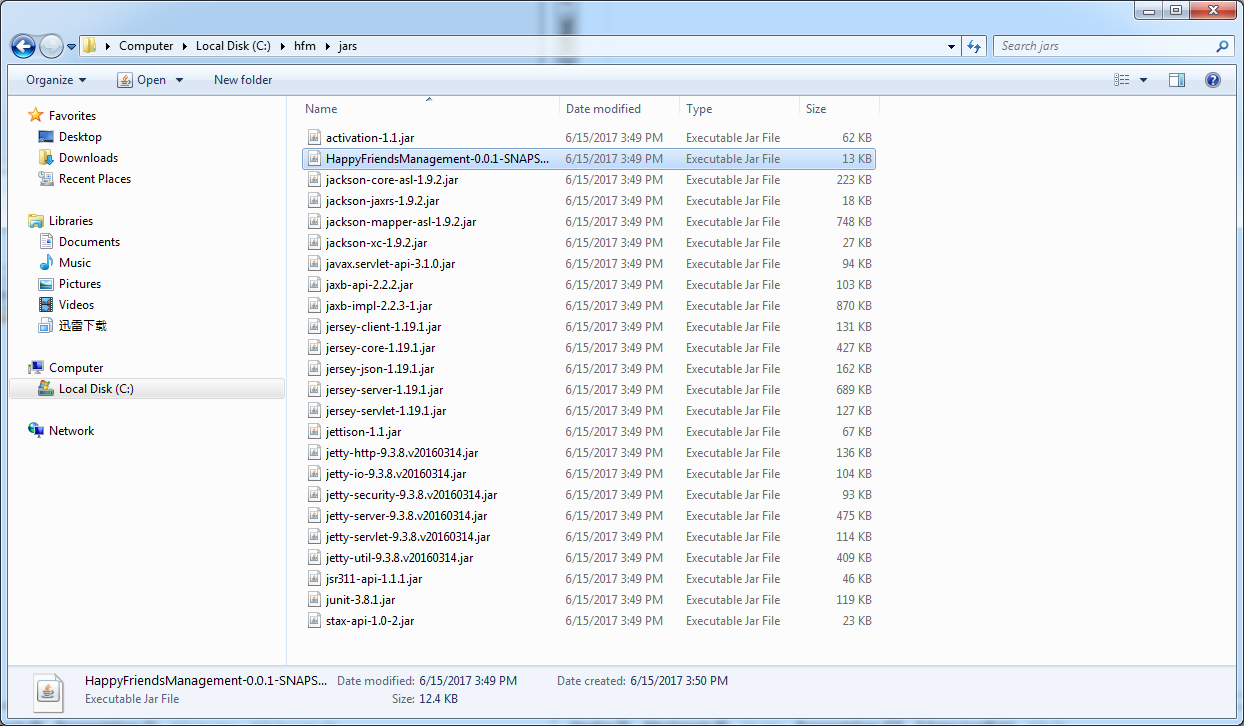
</configuration>

</execution>

</executions>

</plugin>

把项目的jar也放进去

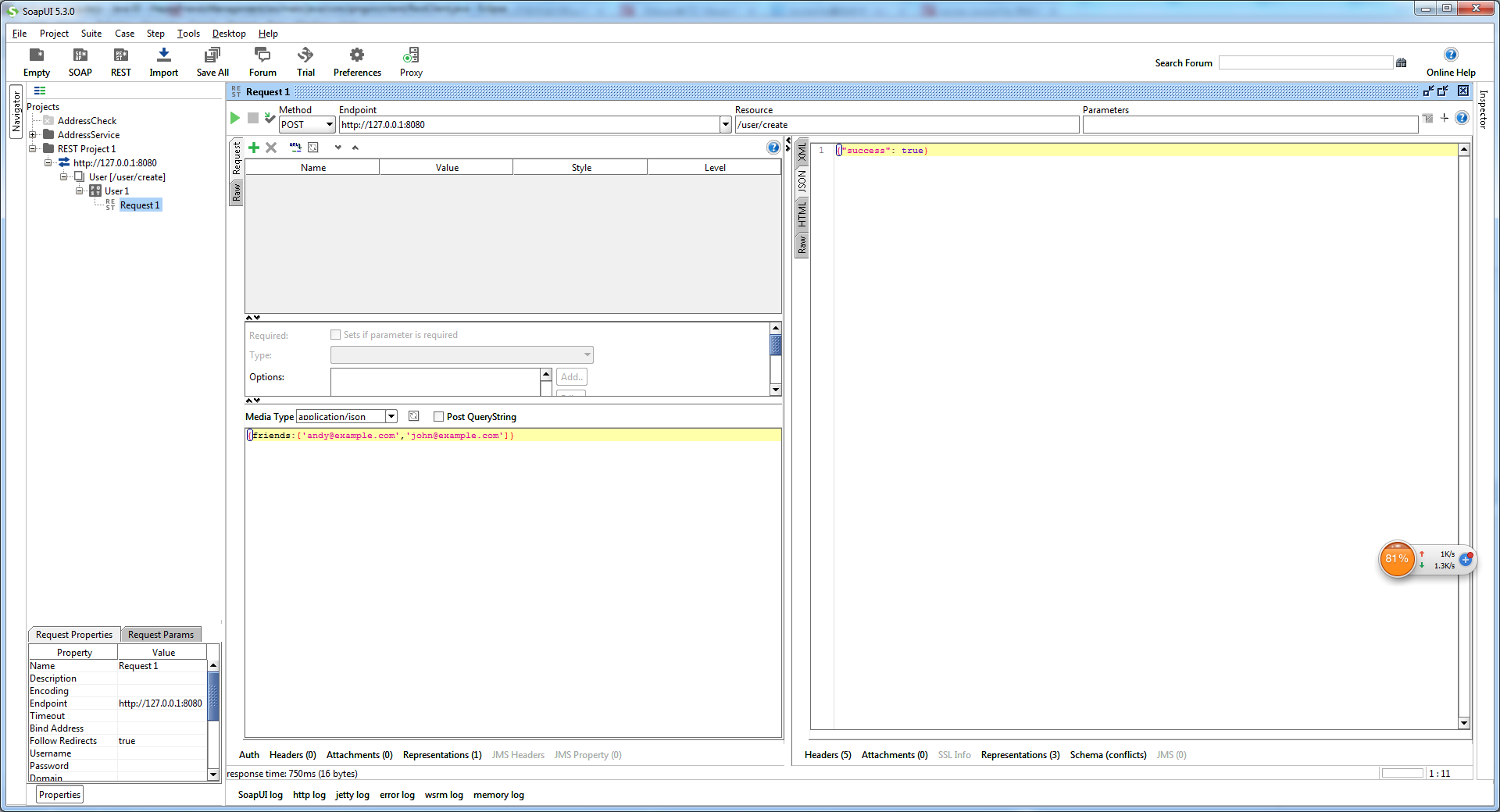


然后使用

C:\hfm\jars>java -cp ./\* com.qingxin.server.JettyServer

在windows上测试了一下

调用那个client和soapui都可以



最后为了release方便一些

我把所有需要的jar都拷贝到项目目录里的

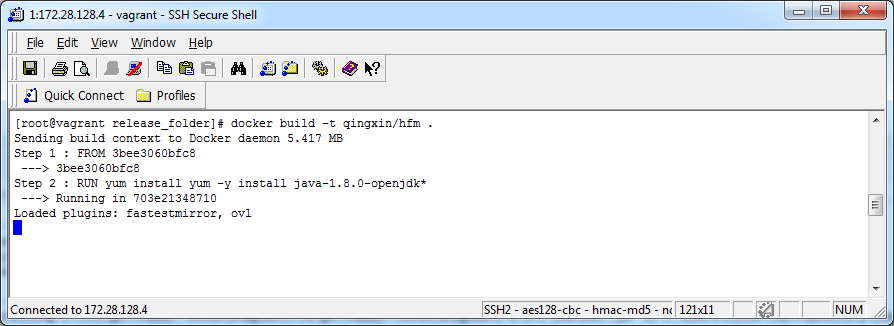
/HappyFriendsManagement/release\_folder

然后你需要把这个目录放在centos上的任意目录里

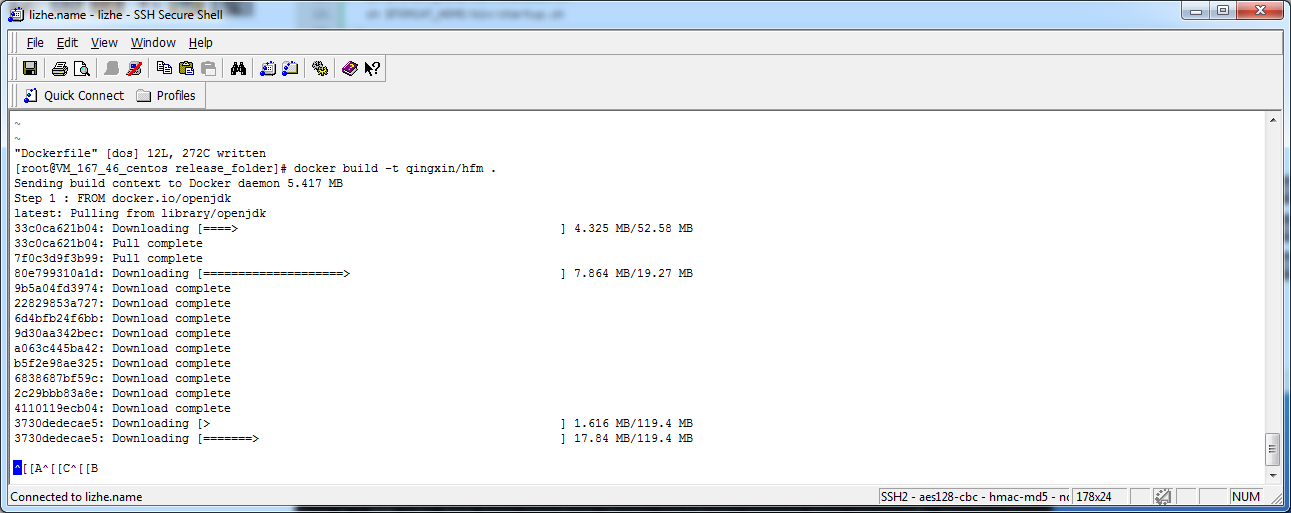
例如/lz/release\_folder

然后在这个目录中通过Dockerfile 构建docker 镜像

[root@vagrant release\_folder]# docker build -t qingxin/hfm .

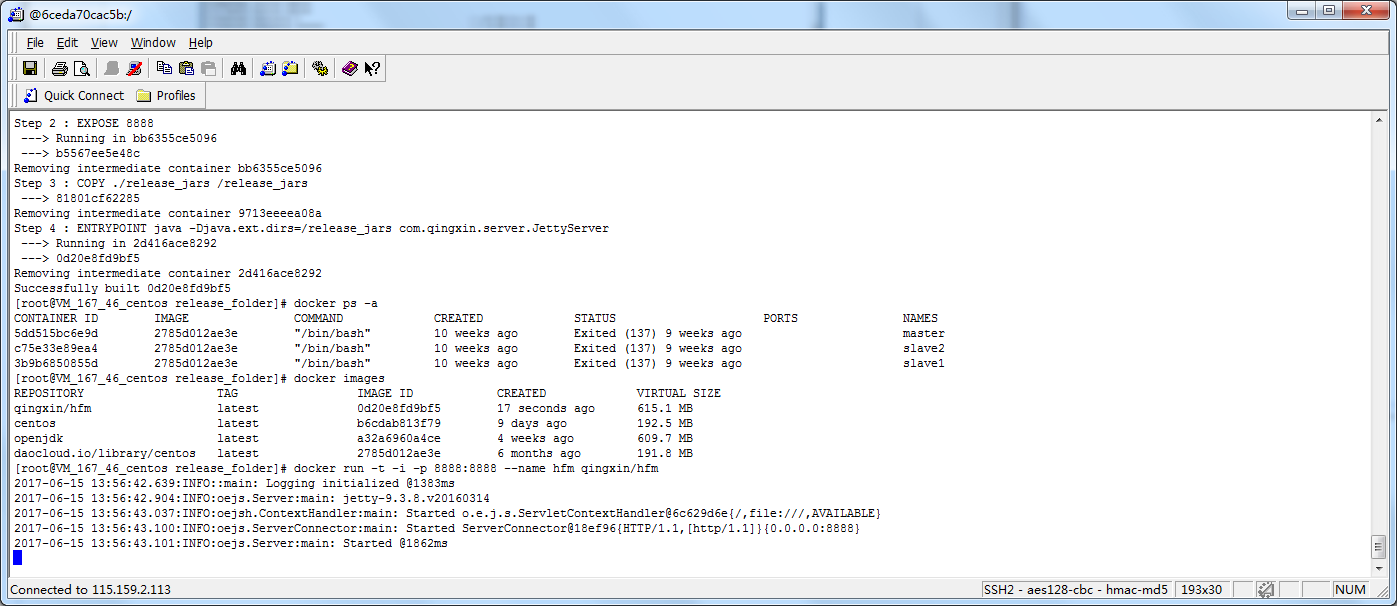


下载速度太慢了,放弃yum 了, 直接pull个带jdk的镜像

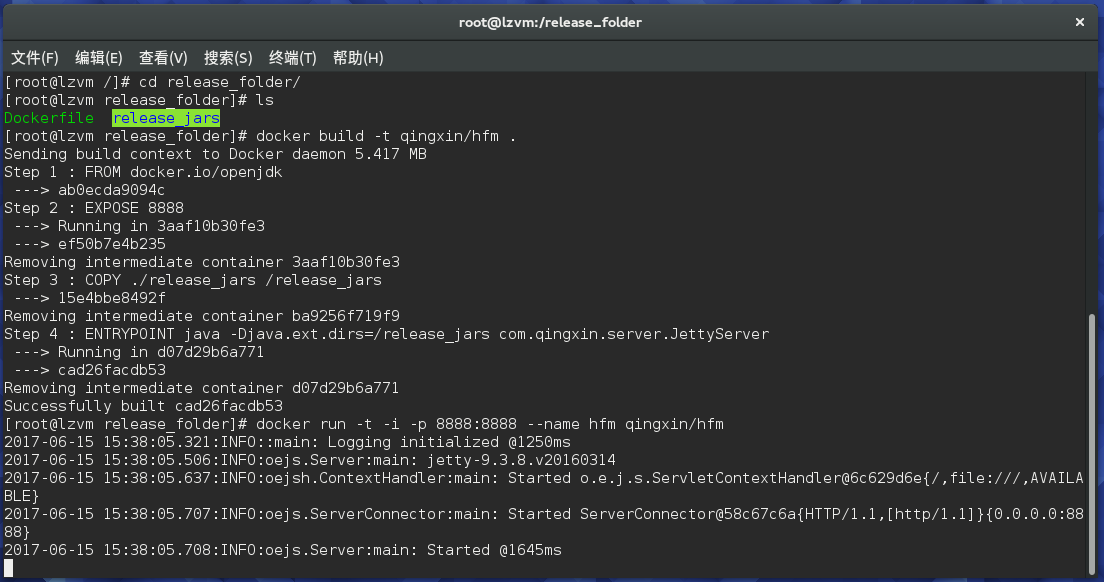


docker build -t qingxin/hfm .

docker run -t -i -p 8888:8888 --name hfm qingxin/hfm



换个虚拟机试一下



用本地客户端测试一下

