**微服务开发环境搭建**

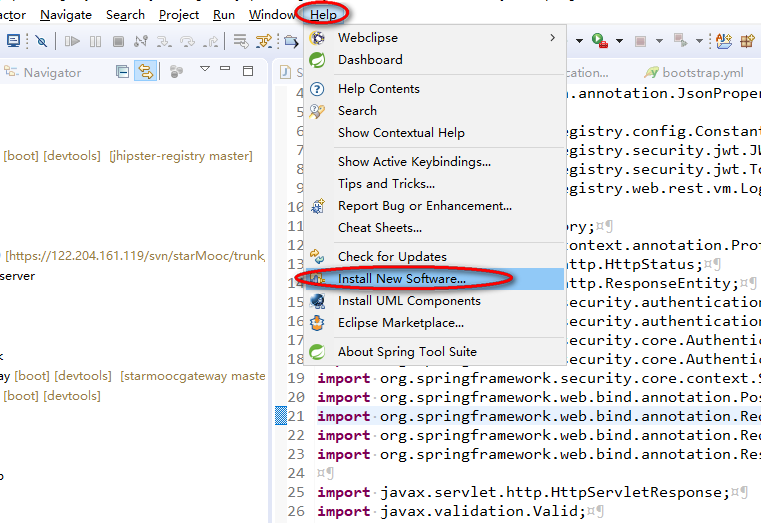
# 通过docker 镜像安装和部署zookeeper和kafka

此方法无需在window下自行安装zookeeper和kafka服务。

前提：正确安装docker

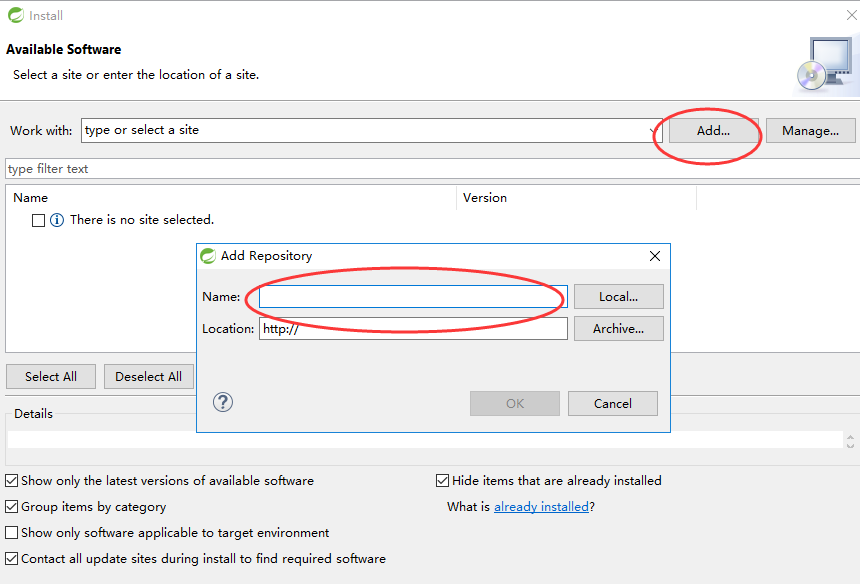
## 安装docker tooling

如下图，在eclipse中添加docker tools软件安装源。

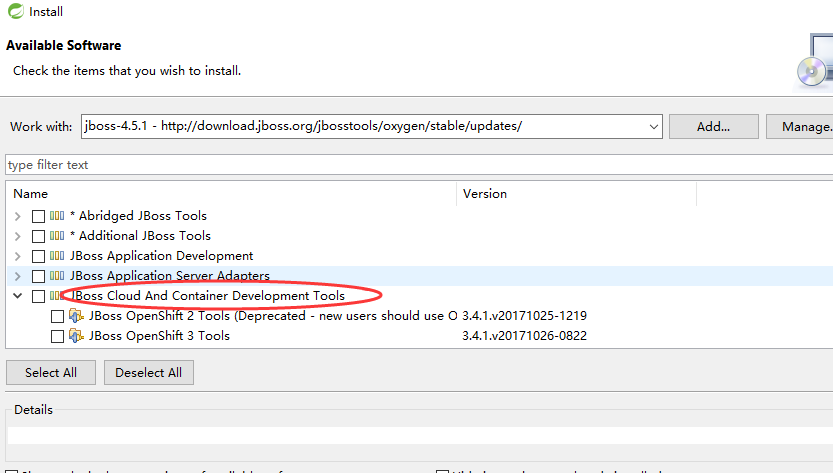


在弹出框中点击add，填写name和location，name名称用户根据需要自行填写，location填写如下地址：

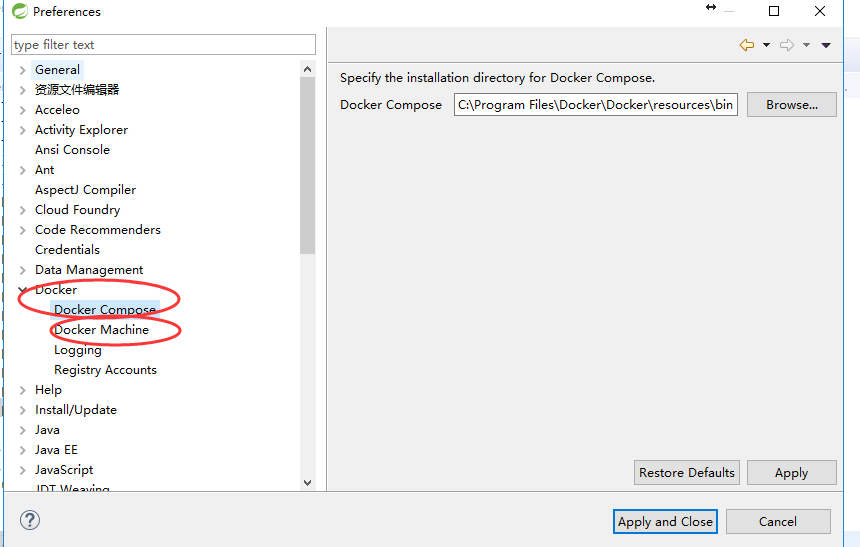
<http://download.jboss.org/jbosstools/oxygen/stable/updates/>



在如下图所示的条目下勾选docker相关条目。然后一路确认安装。



如下图，安装完成后，配置docker tooling， docker compose和Docker Machine路径选择docker安装路径；Docker Machine VM driver配置可读写的目录即可。

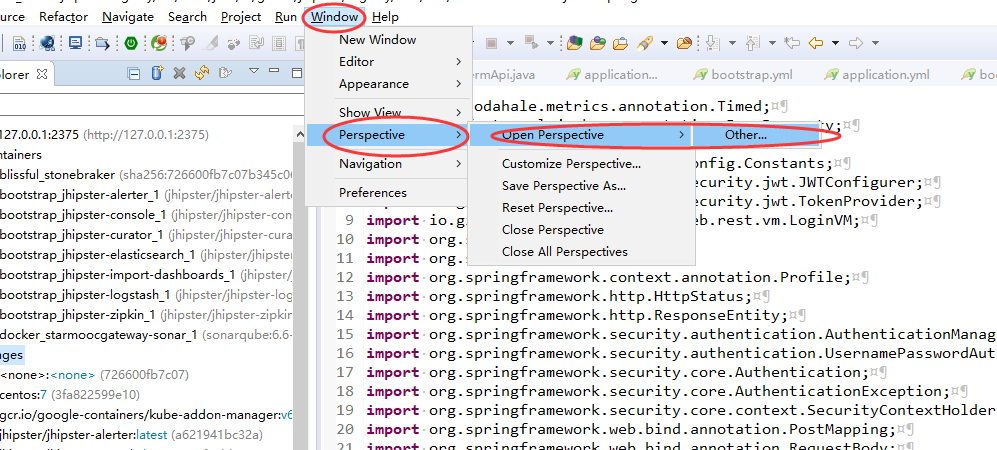


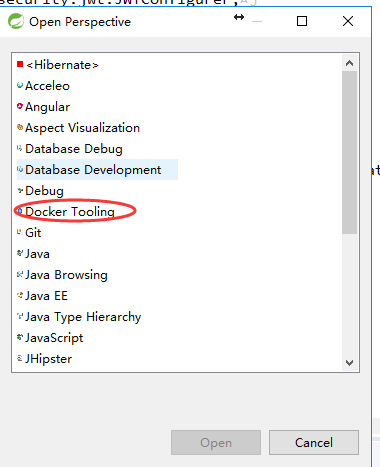
## 镜像拉取

配置好环境后，就可以在eclipse下用界面操作docker相关功能，免去命令行输入的麻烦。

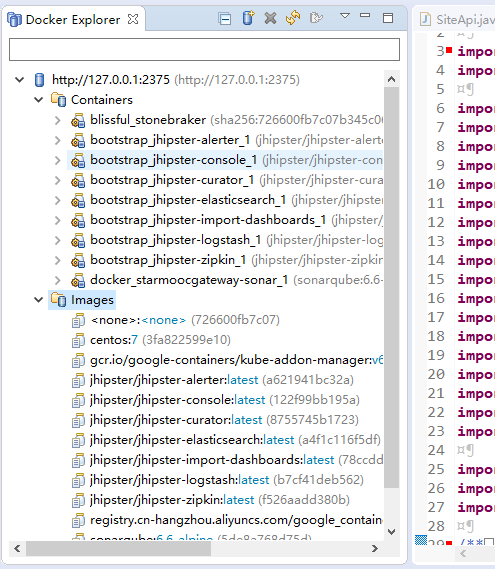
下面进行kafka服务镜像的拉取为例，说明如何从docker hub上获取镜像，zookeeper类似。

如下图，首先打开docker exploer，选择docker tooling。

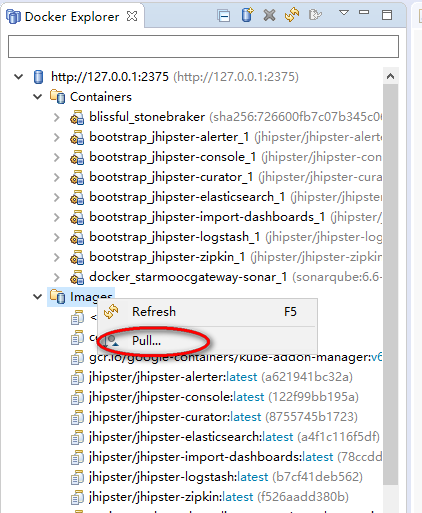




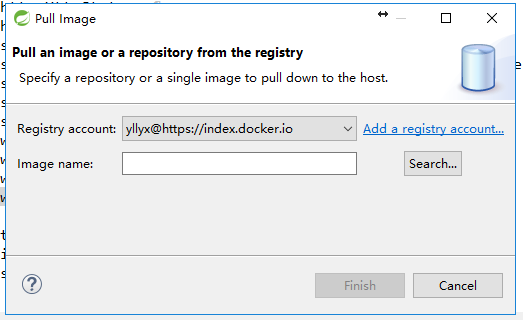
选择后，docker exploer如下图：



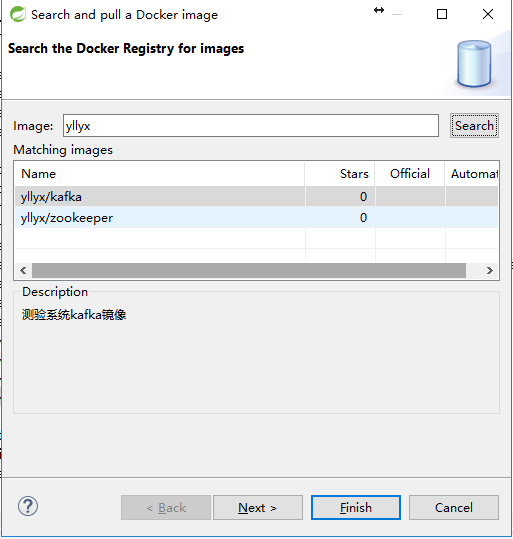
如下图，右键点击Images，选择pull



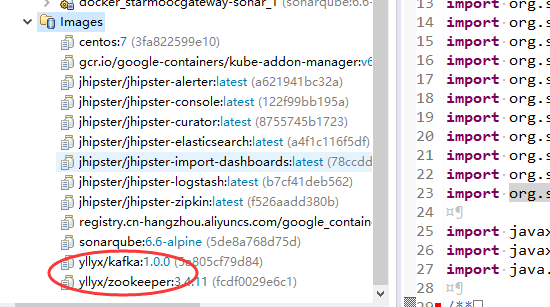
在弹出框中填写docker hub库地址。没有账户的到docker hub官网<https://hub.docker.com/>注册一个。



如下图，点击上图中的search，弹出如下镜像查找对话框，在对话框中搜索关键字yllyx。

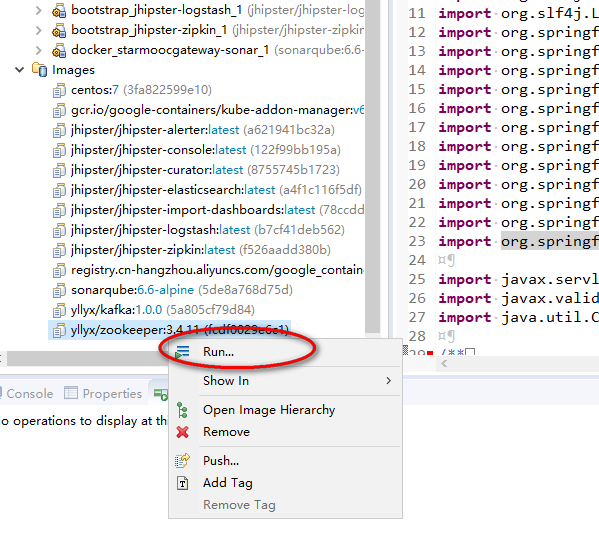


选择kafka后一路next,选择完镜像后，在如下图所示的对话框中点击finish。Eclipse开始下载镜像。同样的方法下载Zookeeper镜像。下载完后如下图：

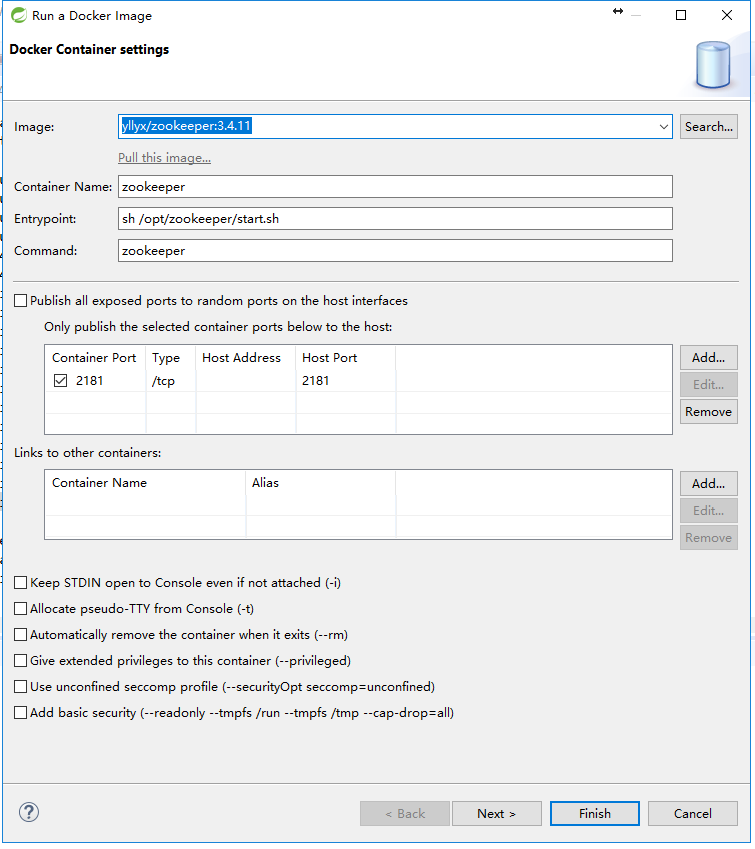


## 配置启动镜像

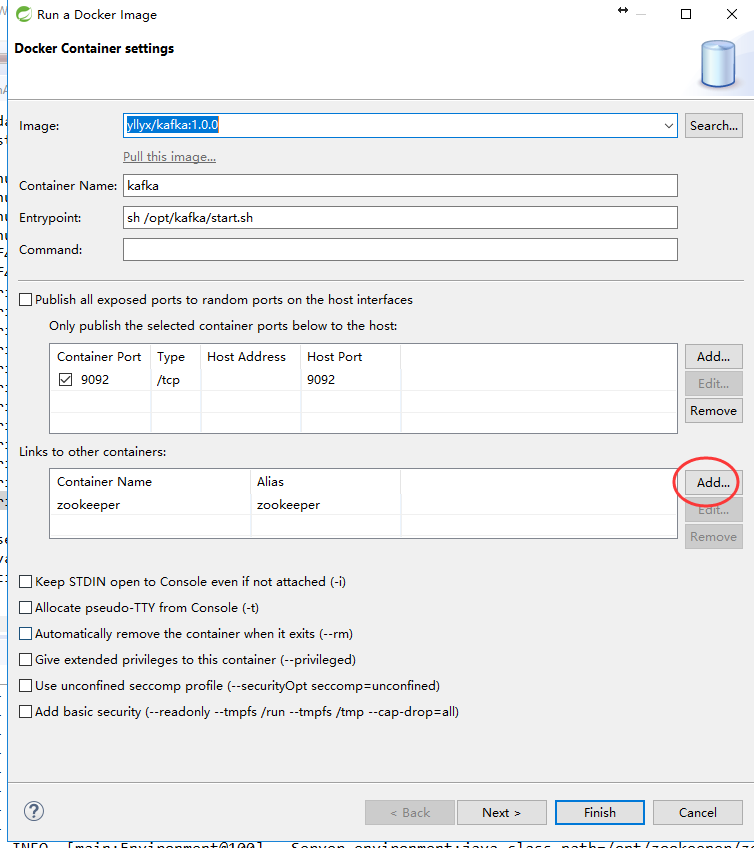
由于kafka启动依赖于zookeeper，首先先启动zookeeper。如下图，右键点击zookeeper镜像，在弹出框中选择Run菜单。



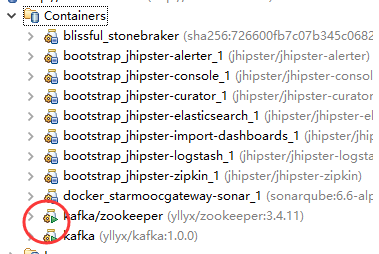
选择Run，弹出如下图所示运行配置界面，按下图界面配置即可。点next按钮，后续界面只需修改Network Mode，选择bridge模式。最后点击finish。



Zookeeper启动完成后，配置kafka运行参数，由于kafka需要连接Zookeeper，在配置页面需要配置links to other container，在添加与zookeeper的关联。如下图：



其他配置步骤不变。最后点击finish。运行状态如下图：



# Jhipster registry服务源码下载

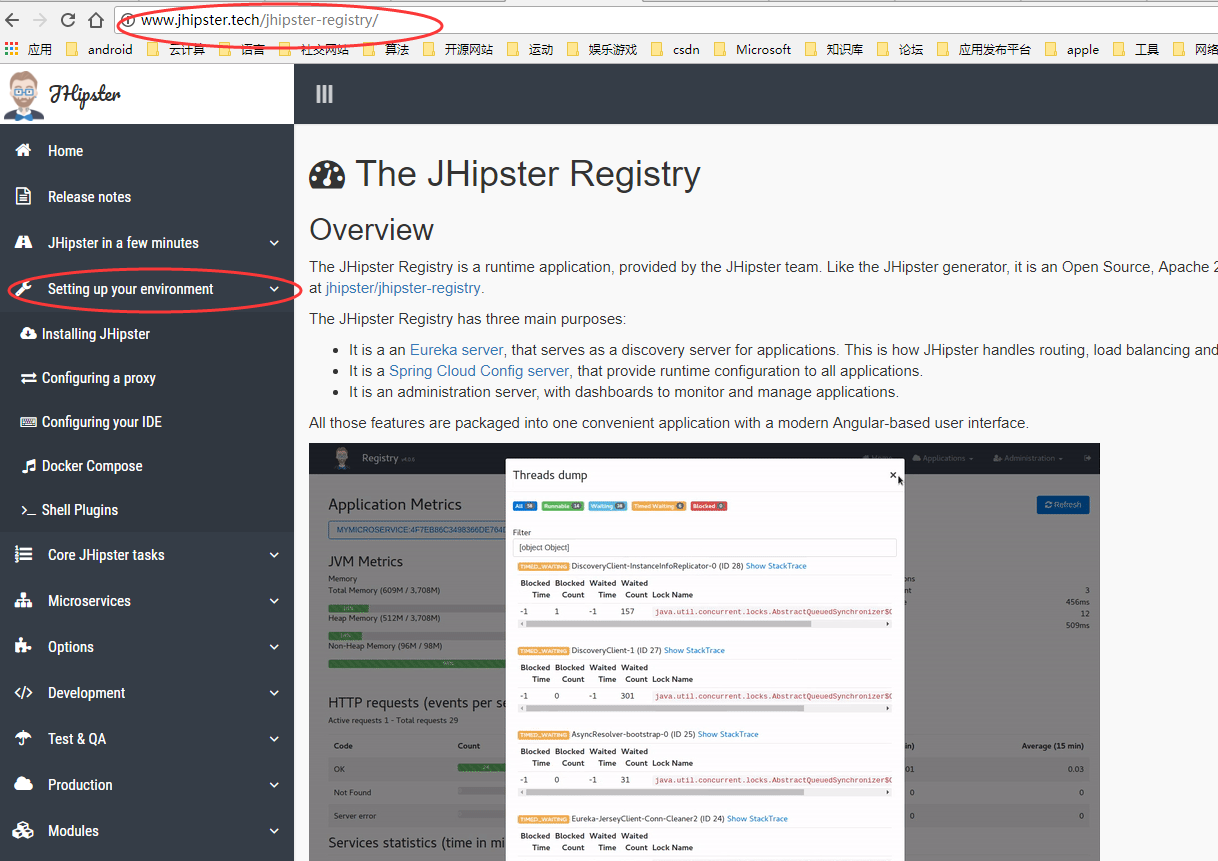
源码下载路径：

<https://github.com/jhipster/jhipster-registry>

下载完成后解压到指定目录，并以maven工程的方式导入eclipse。

# 配置jhipster开发环境

开发环境可以按照jhipster官网说明配置，按照如下图所示章节配置。



# 生成gateway 服务

生成gateway服务，可参考如下网址

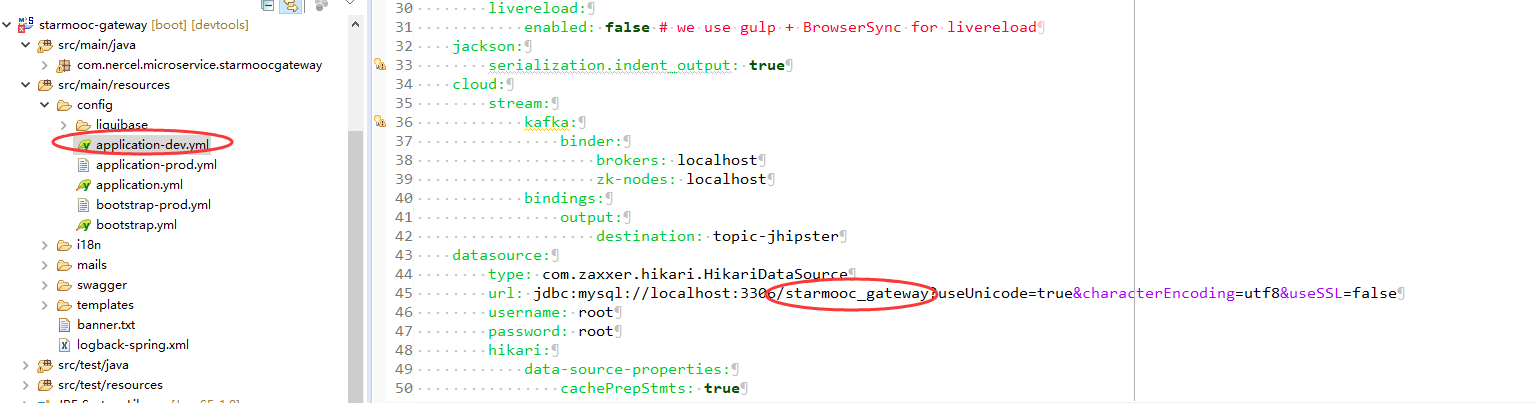
<https://dzone.com/articles/develop-and-deploy-microservices-with-jhipster-1>

主要参考其中的Create an API Gateway部分

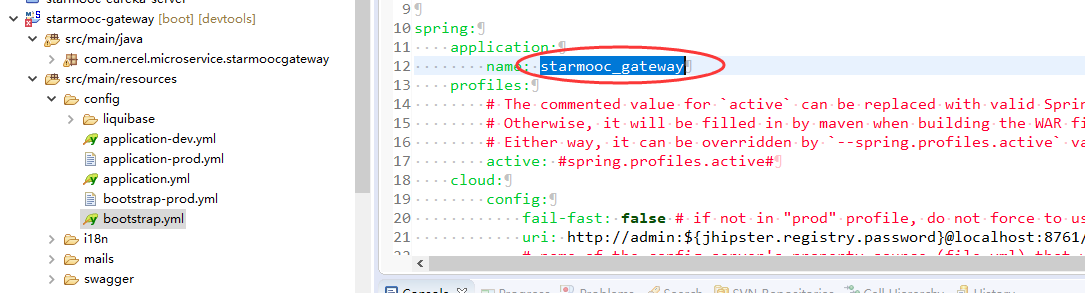
生成源码后，以maven工程的方式引入到eclipse。

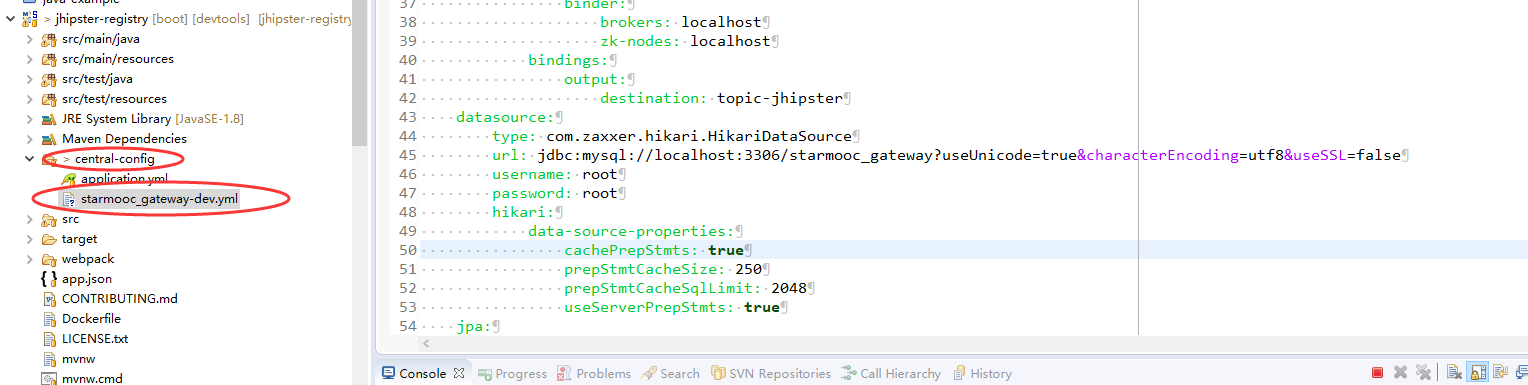
# 启动服务发现和网关服务

在启动服务之前，需要配置服务运行所属的数据库。在本地新建如下图所示的数据库，数据库名自行配置。

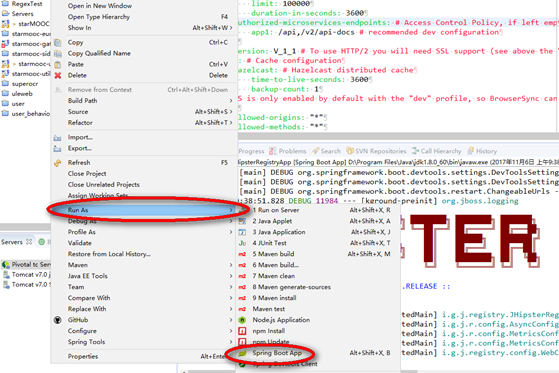


将application-dev.yml复制到jhispter registry本地配置文件中心，并按照服务应用名称-开发环境方式取名。示例gateway服务名称如下图。复制到服务注册中心的配置文件名称如下图：

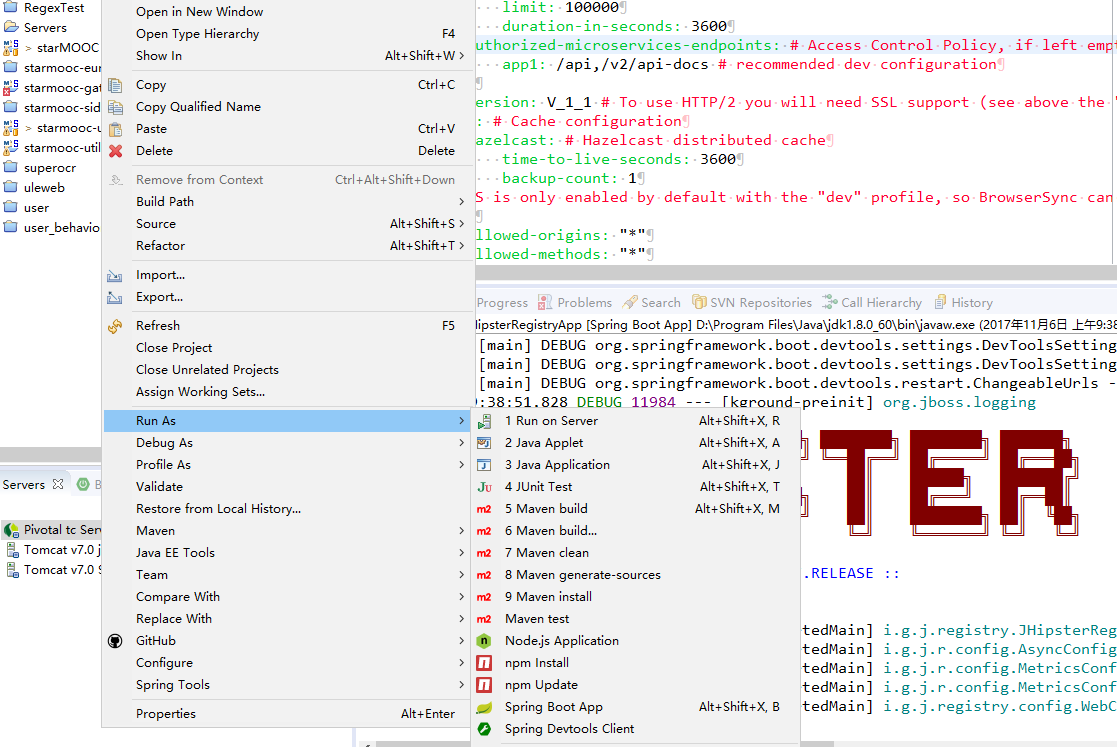


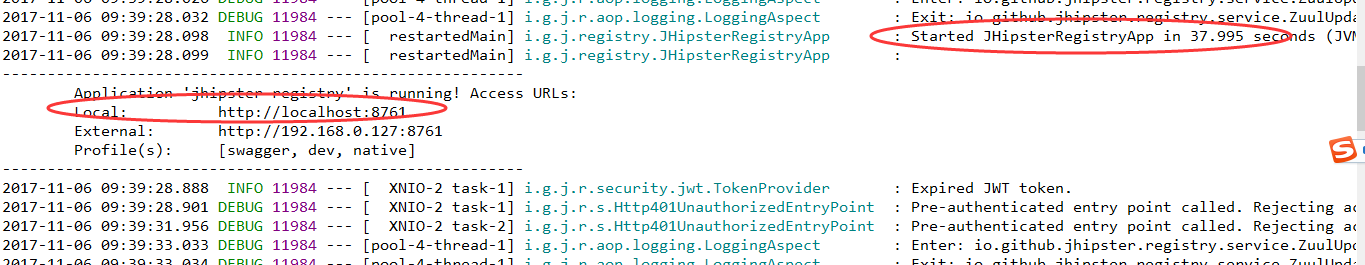


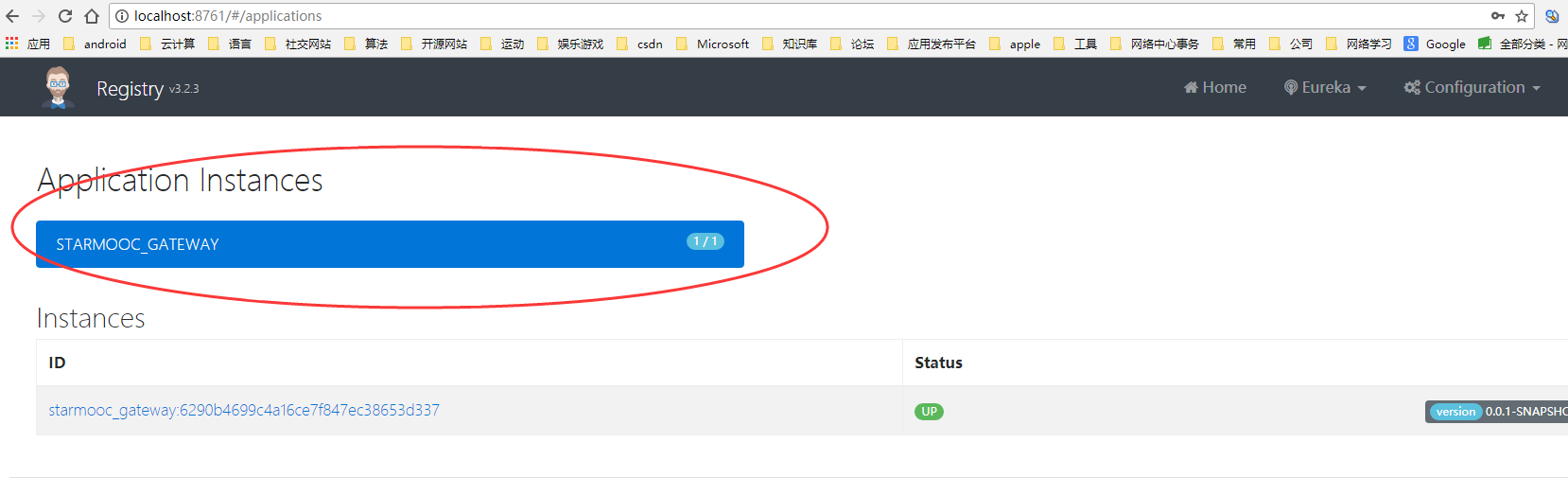
完成上述所有步骤后，以spring boot方式先后启动服务发现组件（jhipster registry）和 网关组件（gateway）。



启动成功后，服务发现地址<http://localhost:8761>，登录后可以看到注册到服务中心的gateway服务。







# 后续说明

此微服务开发框架如下图所示，完成上述所有步骤后，我们已经搭建好下图红圈标示的服务组件，后续开发只需要按照业务要求开发所需的微服务和前端所需的页面。下图黄圈所示部分。

