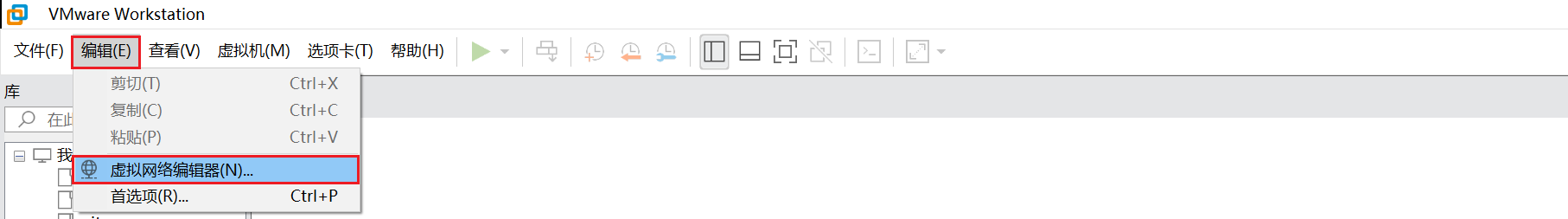
**第0章-环境搭建**

**1、导入虚拟机【方式一】**

在四方保险项目中，有很多使用到了的组件；这些组件的安装下载在课堂中下载、配置比较费时；所以已经在资料中提供好了这些组件的安装后的虚拟机；只需要导入虚拟机【方式一】就可使用。如果执行导入虚拟机出现难以解决问题或者环境、电脑不匹配的话选择 【方式二】来执行。

**1.1、虚拟机网络设置**

由于导入的虚拟机的网段是固定ip；所以需要设置一下虚拟机软件vmware的网段以适配导入的虚拟机的运行网络环境。具体步骤如下：



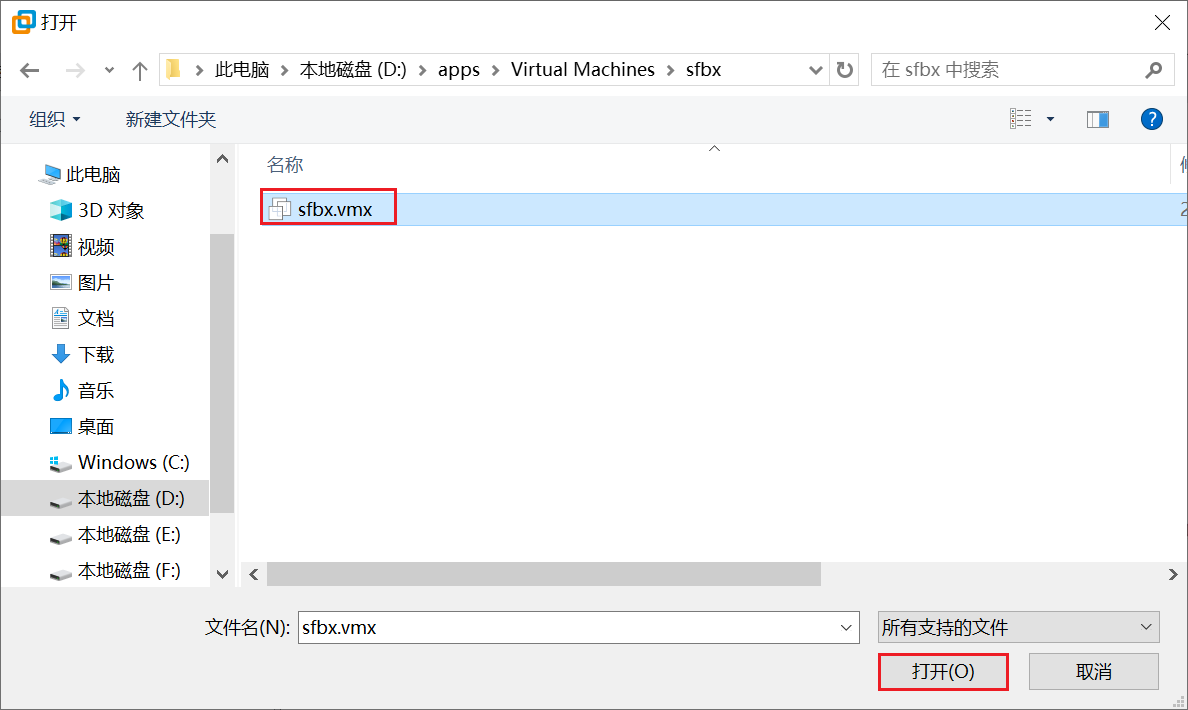


设置完vmware中的nat的网卡的网段之后，就可以导入虚拟机啦

**1.2、导入虚拟机**

解压 sfbx.zip ；然后在vmware打开即可。

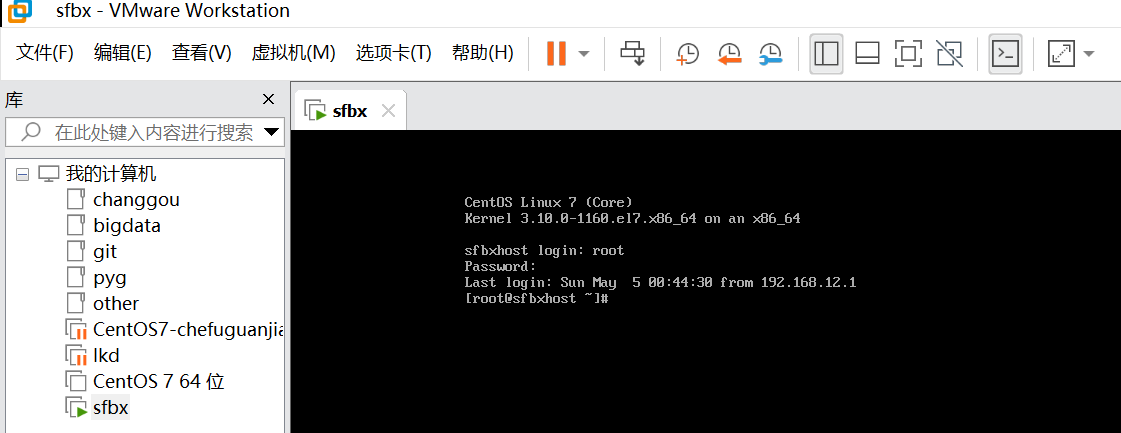




在打开之后；vmware显示该虚拟机之后；可以 开启此虚拟机。

|  |
| --- |
| Tips：**启动** 的时候；如果提示：  我已移动该虚拟机；我已复制该虚拟机 的选择的话；请选择：**我已移动该虚拟机** |

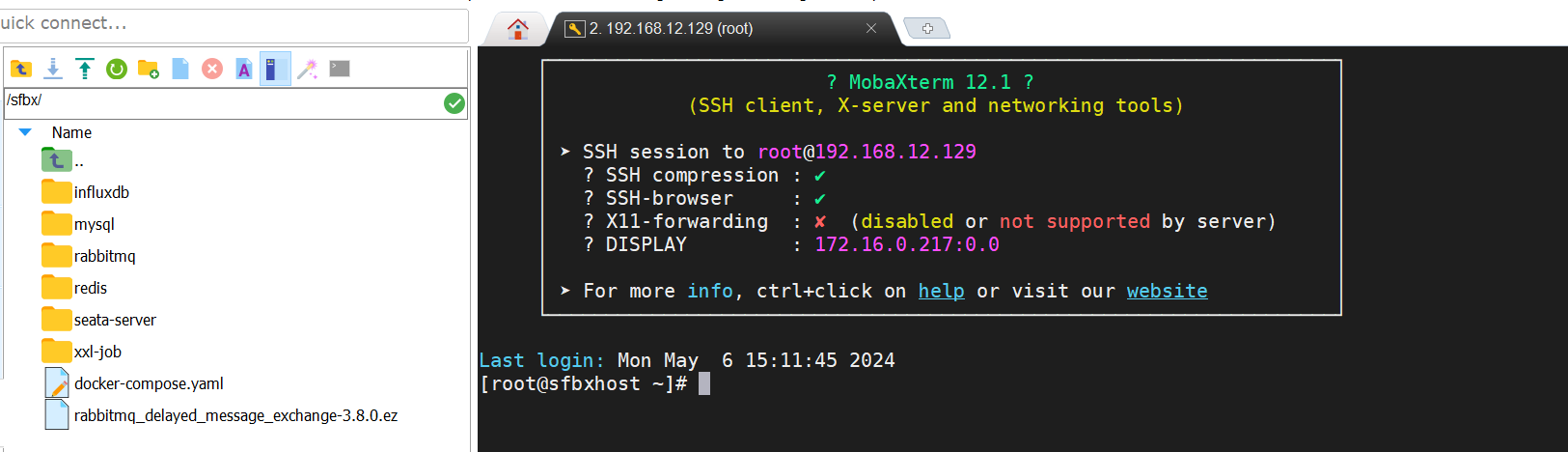
启动后如下：



远程登录的IP：192.168.12.129

系统用户名：root

密码：pass



**1.3、组件介绍**

在虚拟机中安装了各类组件；这些组件的在docker中的容器名及信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 访问地址 | 用户名 | 密码 |
| nacos | http://192.168.12.129:8848/nacos | nacos | nacos |
| redis | 192.168.12.129 |  | pass |
| rabbitmq | http://192.168.12.129:15672/ | admin | pass |
| seata-server | 192.168.12.129:9200 | seata | seata |
| mysql | 192.168.12.129:3306 | root | pass |
| xxl-job-admin | http://192.168.12.129:8280/xxl-job-admin/ | admin | 123456 |
| influxdb | http://192.168.12.129:8086 | admin | 123456 |

**2、Docker Compose【方式二】**

**说明**：方式二，也就是这个章节是为了导入虚拟机失败或者mac的同学准备的；如果导入虚拟机并且启动没有问题的话；这个章节就可以不用看了。

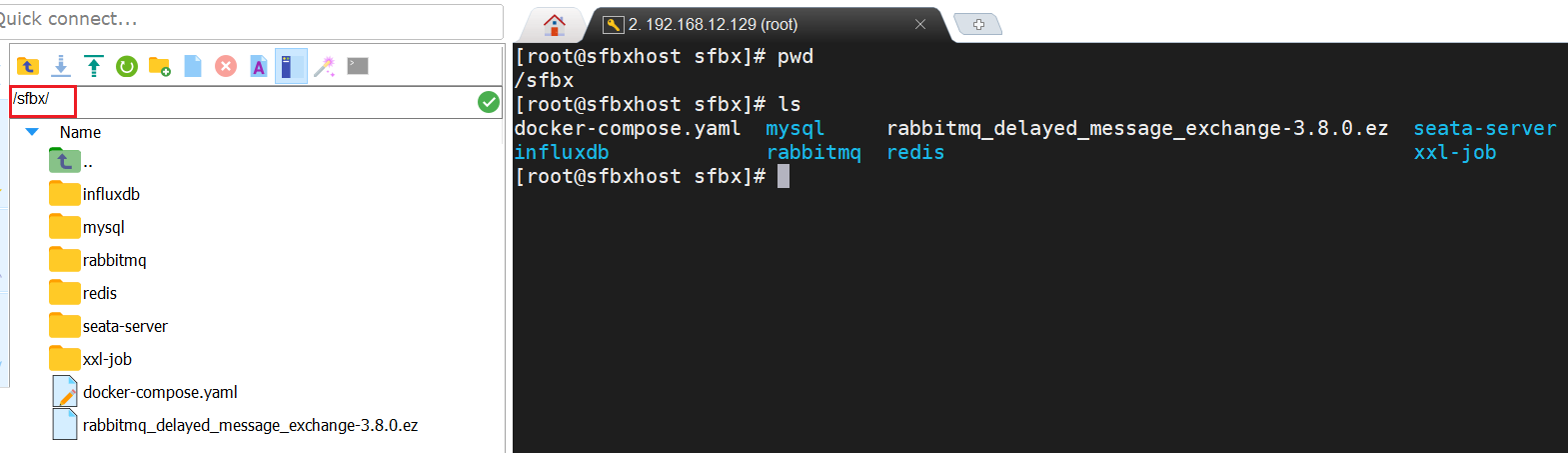
**2.1、执行compose**

1）**修改资料**

打开 资料/sfbx/mysql/source/nacos.sql 文件；然后将里面的 192.168.12.129 替换为你自己虚拟机对应的IP即可。

2）**复制资料**

将 资料/sfbx 这个文件夹复制到linux系统的根路径下



3）**执行compose文件**

使用模拟终端；比如：MobaXterm、finalSheel、Xshell等；执行如下命令：

|  |
| --- |
| Shell cd /sfbx  # 执行compose文件；后台方式创建并启动所有组件 docker compose up -d |

启动完后；可以按照 1.3 章节的组件介绍确认组件是否启动成功。

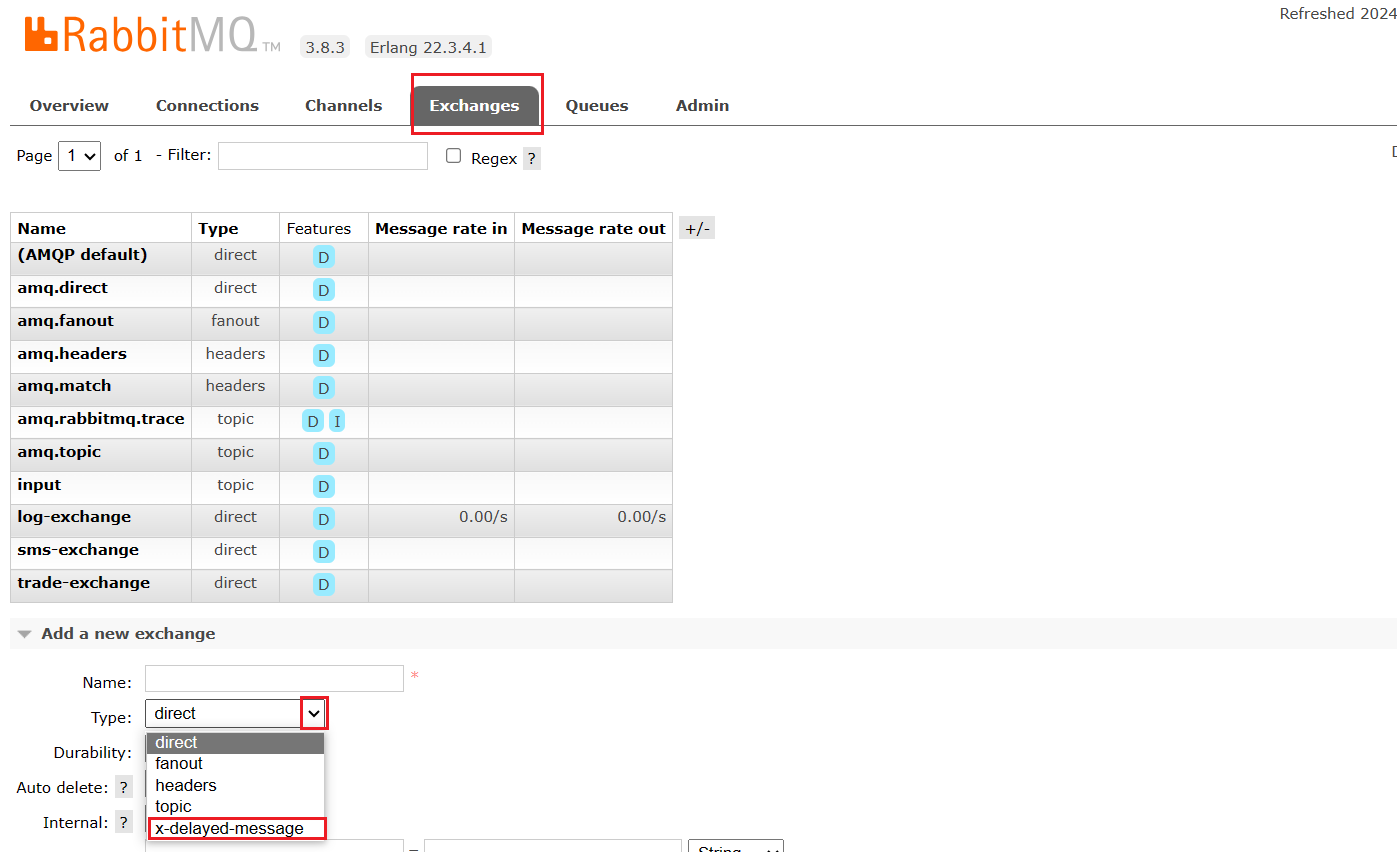
**2.2、安装rabbitmq延迟插件**

|  |
| --- |
| 后续开发中会使用到延迟队列，这里采用安装插件的方式来实现延迟队列，需要安装**delayed-message-exchange插件**  RabbitMQ官方提供插件列表：https://www.rabbitmq.com/community-plugins.html  下载插件地址：<https://github.com/rabbitmq/rabbitmq-delayed-message-exchange/releases>  资料/sfbx 中已提供rabbitmq\_delayed\_message\_exchange-3.8.0.ez版本匹配RabbitMQ 3.8.3 |

执行如下命令安装：

|  |
| --- |
| Shell cd /sfbx  #复制插件到容器内部 docker cp rabbitmq\_delayed\_message\_exchange-3.8.0.ez rabbitmq:/plugins  #进入rabbitmq容器 docker exec -it rabbitmq /bin/bash  #允许插件启用 rabbitmq-plugins enable rabbitmq\_delayed\_message\_exchange |

http://192.168.12.129:15672 admin/pass；确认是否已经安装成功延迟插件：



在 Type 下拉框选择后出现 x-delayed-message 的话；说明插件安装成功。

**2.3、influxdb创建数据库**

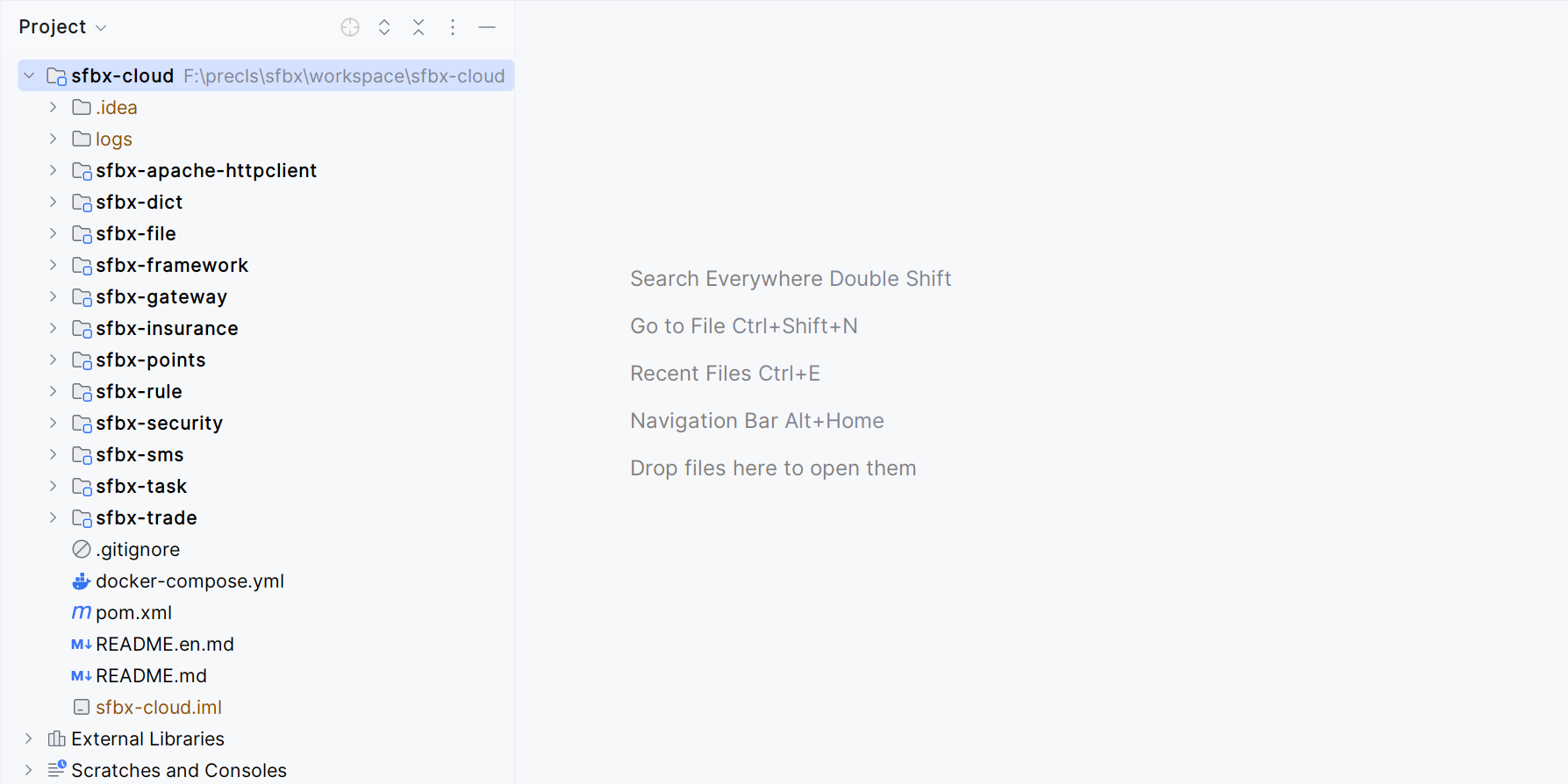
在后期课程中将埋点数据存入到了influxdb；使用的数据为 point\_data；所以要为新创建的influxdb容器创建对应的数据库。具体如下：

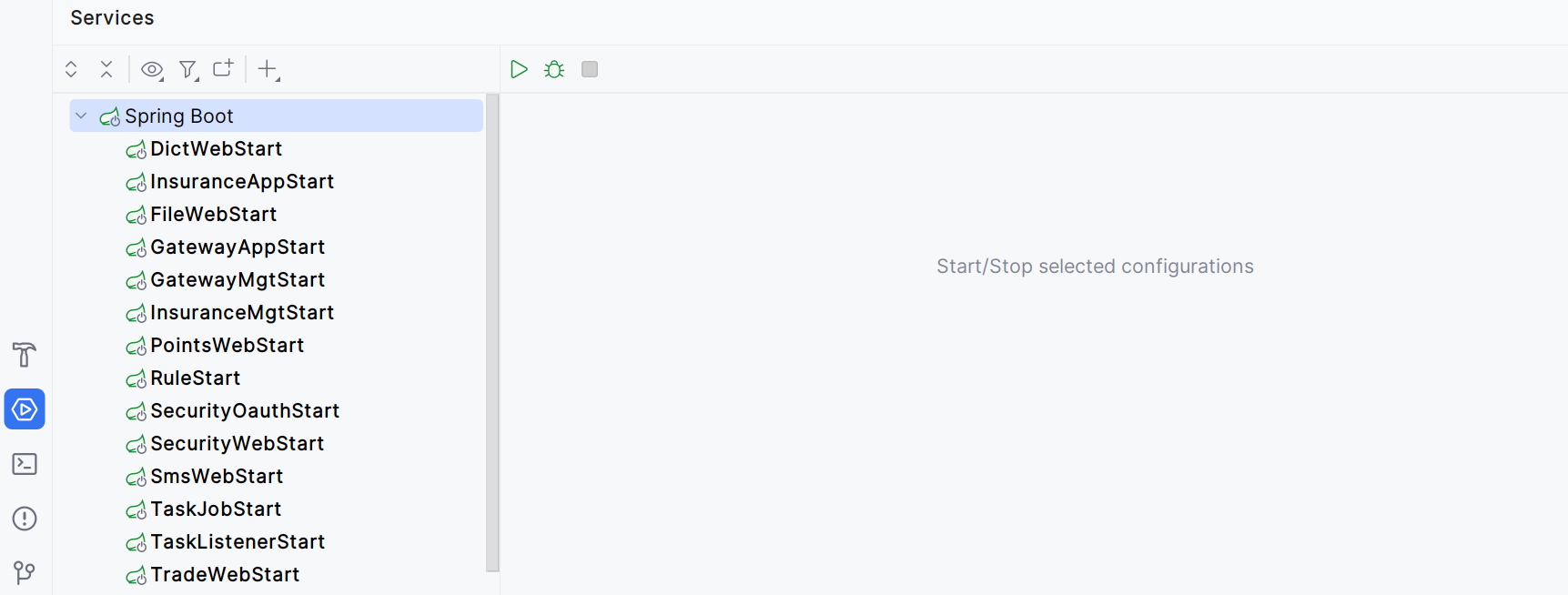
|  |
| --- |
| Shell # 进入influxdb容器 docker exec -it influxdb /bin/bash  # 进入influx数据库环境 influx  # 查看当前所有数据库 show databases  # 创建 point\_data 数据库 create database point\_data  # 查看当前所有数据库；确认新创建的 point\_data是否在输出中；在则说明创建成功 show databases |

**3、启动后端项目**

1）将 资料/sfbx-repository 解压覆盖自己本地maven仓库节省下载依赖时间；

2）将 资料/代码/sfbx-cloud 目录复制到自己的工程目录下；再使用IDEA打开工程，启动各个微服务即可。





**4、启动前端项目**

**4.1、设置域名**

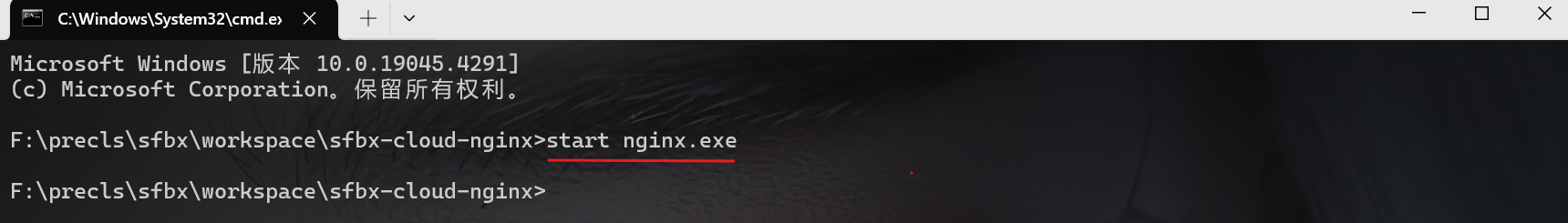
《四方保险》对于域名访问是做安全校验的，需要使用域名方式才可进行访问，为了本地环境访问方便，我们需要配置本机的host文件。Windows下找到本机系统盘C:\Windows\System32\drivers\etc\hosts的文件中添加如下内容：

|  |
| --- |
| C++ 127.0.0.1 sf.app.itheima.net 127.0.0.1 sf.mgt.itheima.net |

**4.2、启动前端**

四方保险的后端管理系统、前台app界面都已经被打包成静态页面；只需要能够正常访问即可。可利用nginx进行部署访问。

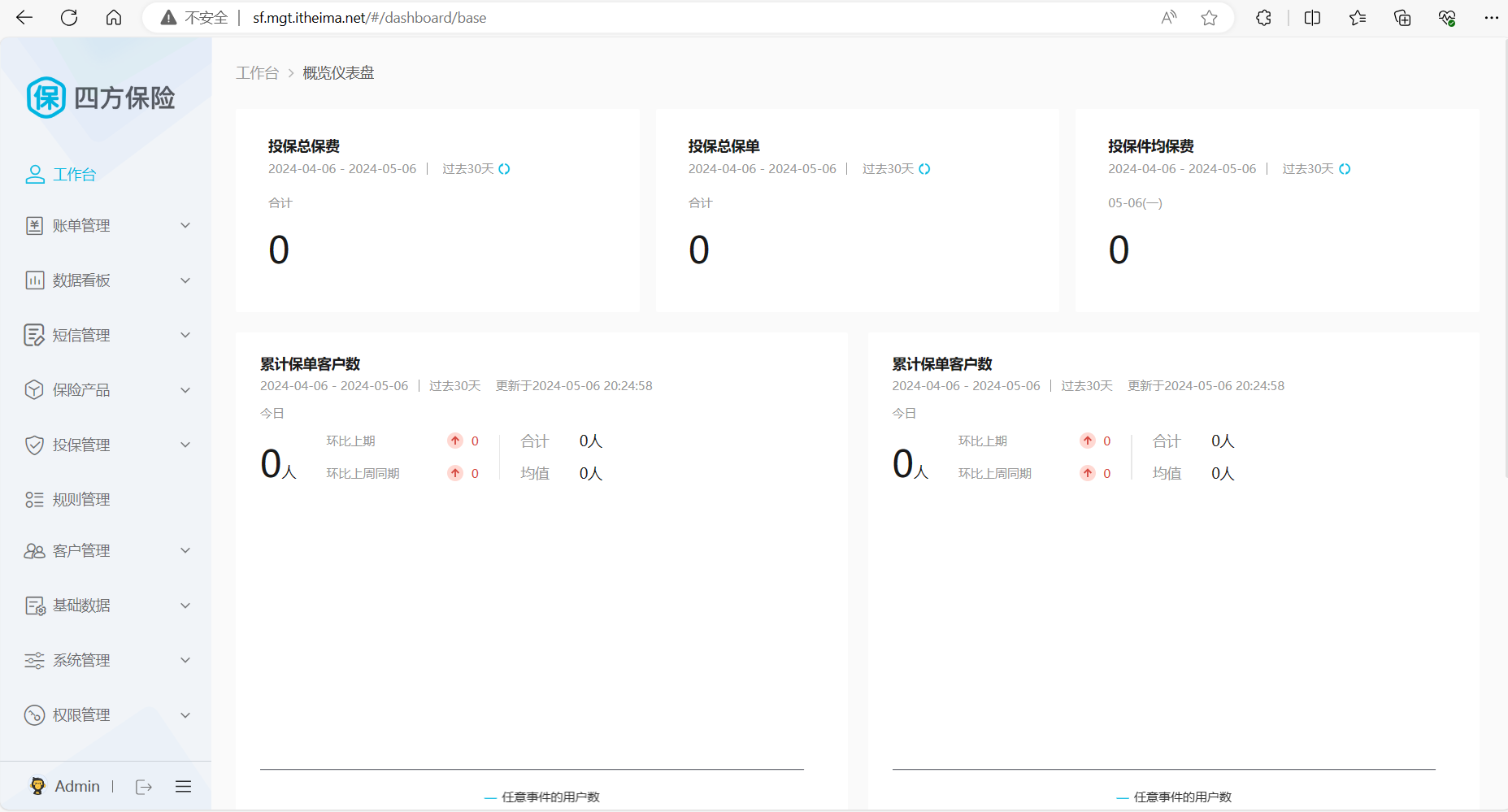
将 资料/代码/sfbx-cloud-nginx 复制到一个没有中文或空格的路径下；然后打开 sfbx-cloud-nginx 文件夹；然后使用命令行启动nginx。



**4.3、访问**

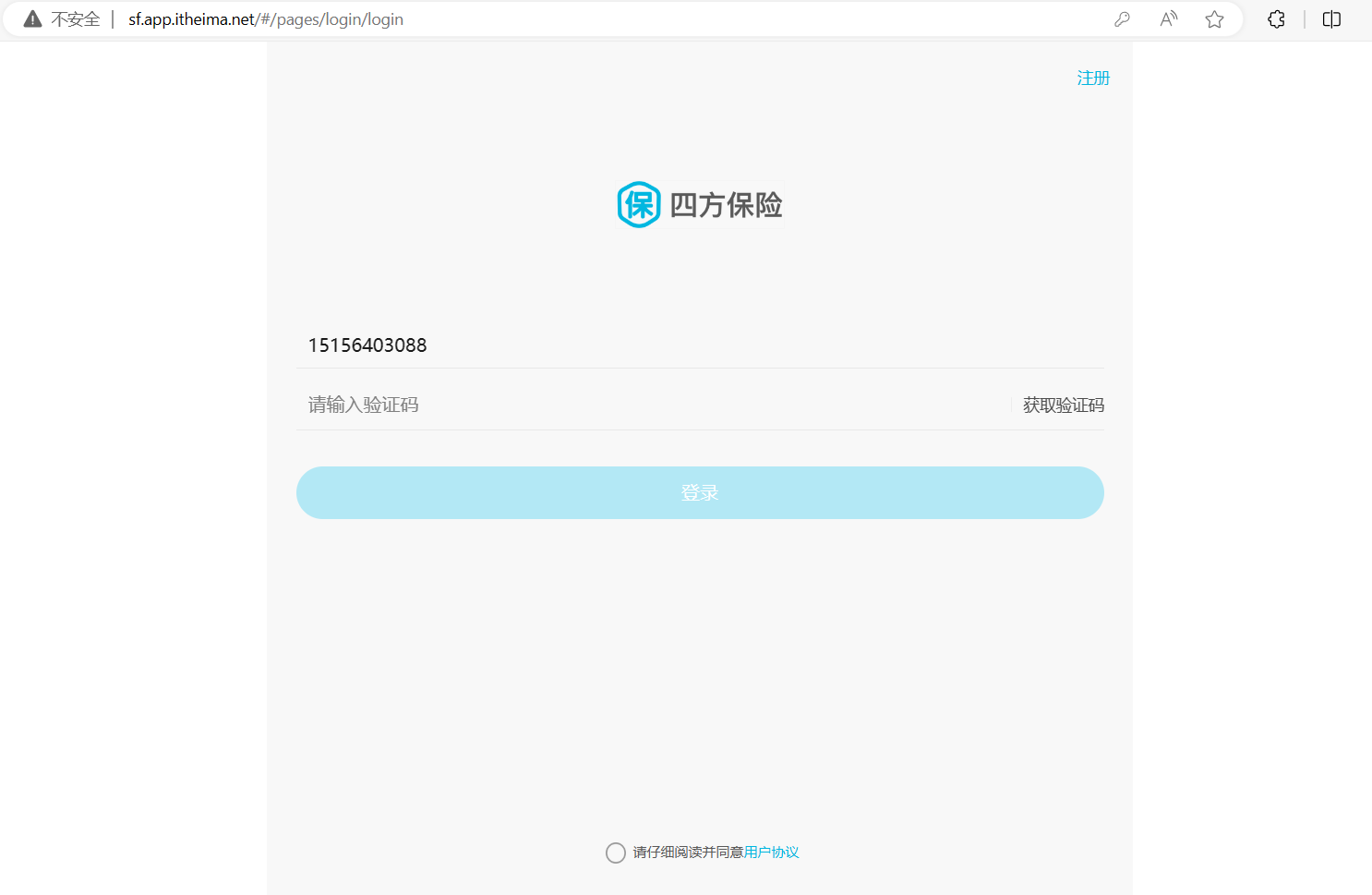
管理端访问地址：http://sf.mgt.itheima.net/

默认账号：admin@qq.com 密码：pass



app端访问地址：http://sf.app.itheima.net/#/pages/login/login

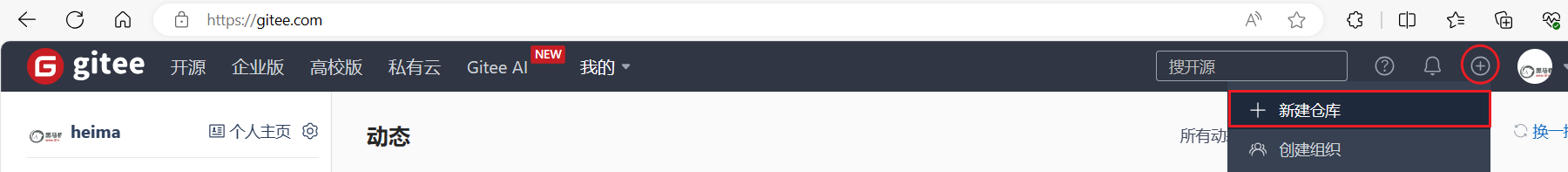
账号：15156403088 密码：123456



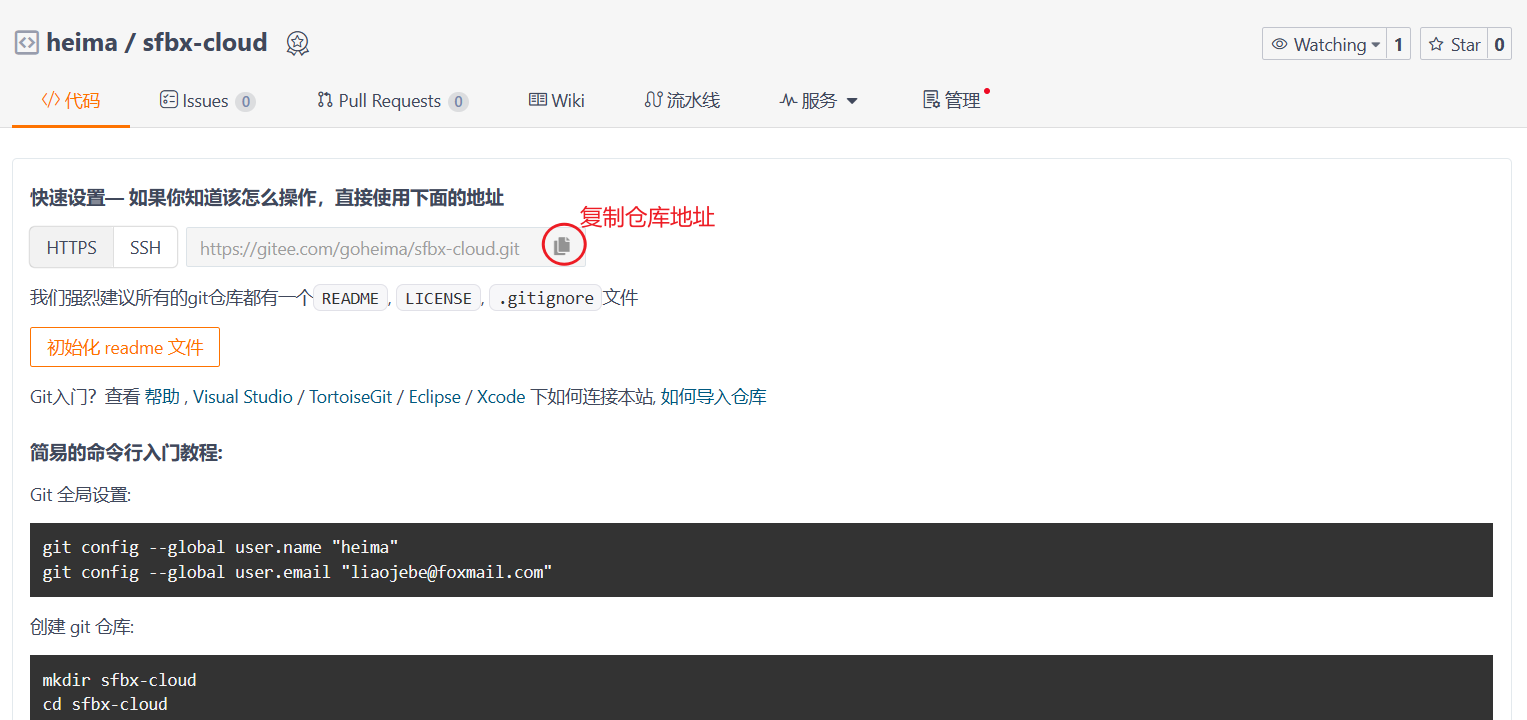
**5、Git项目管理【了解】**

**5.1、创建gitee远程仓库**

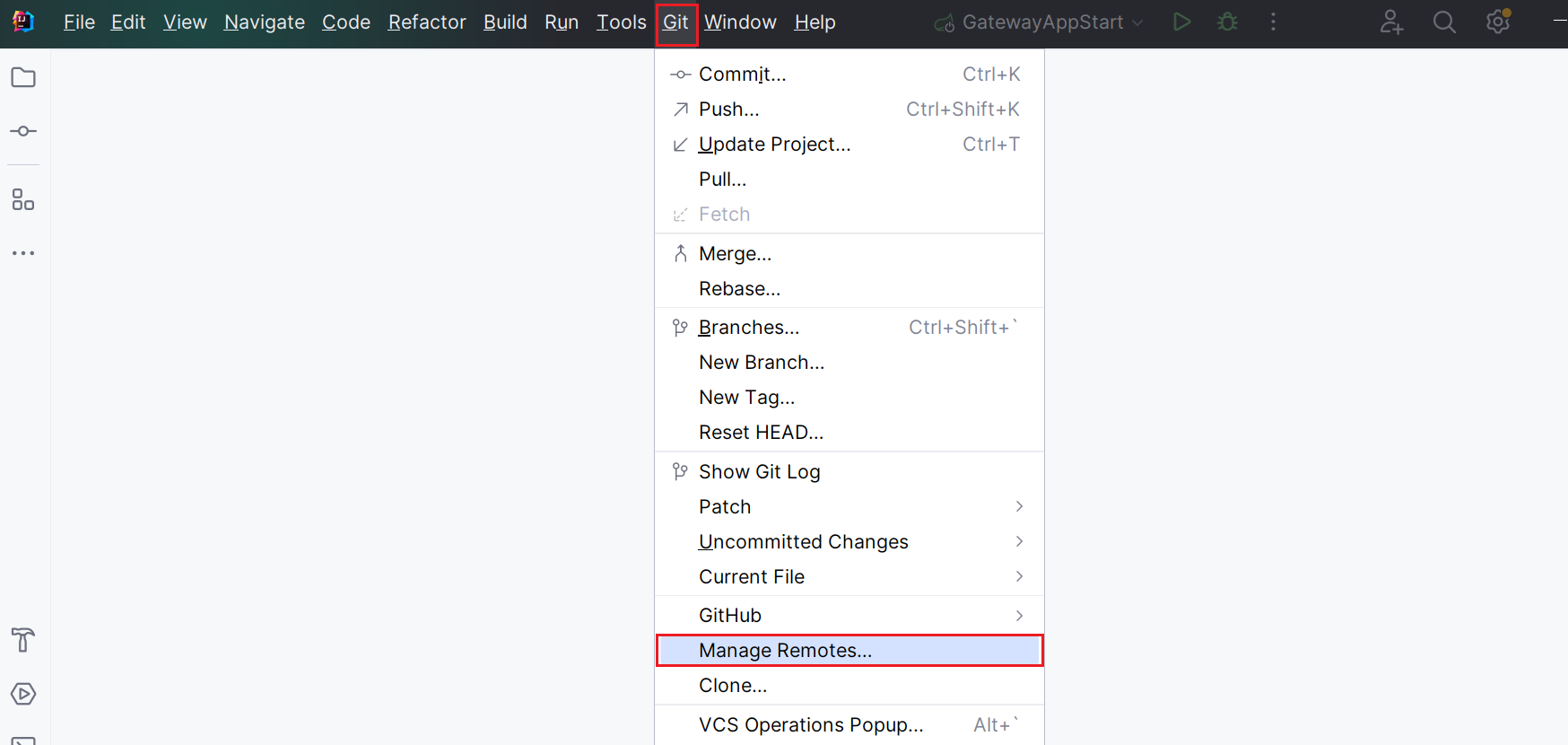
访问 https://gitee.com/ 我们将代码都提交到码云上；方便协作和代码检查。

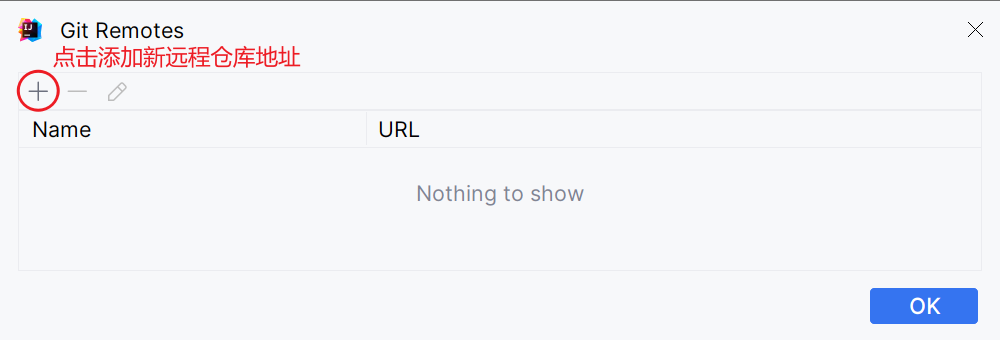


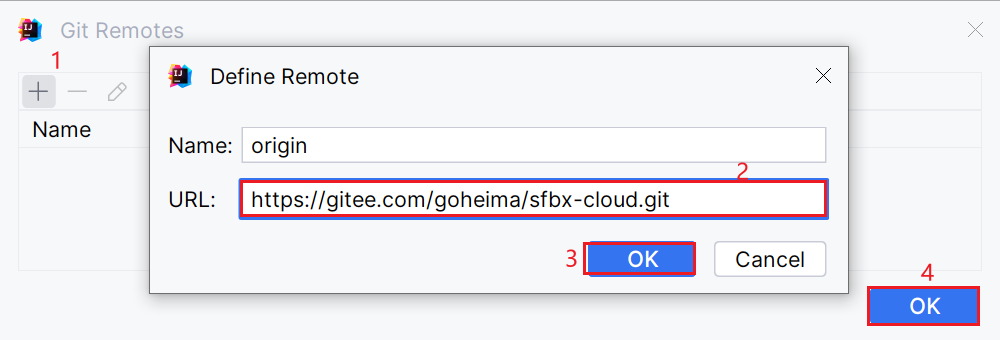


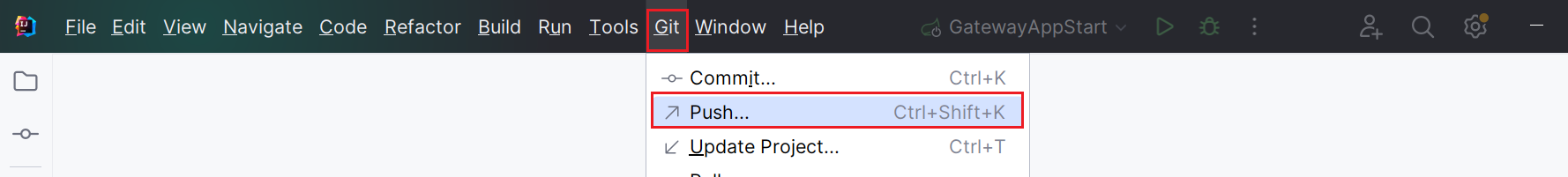


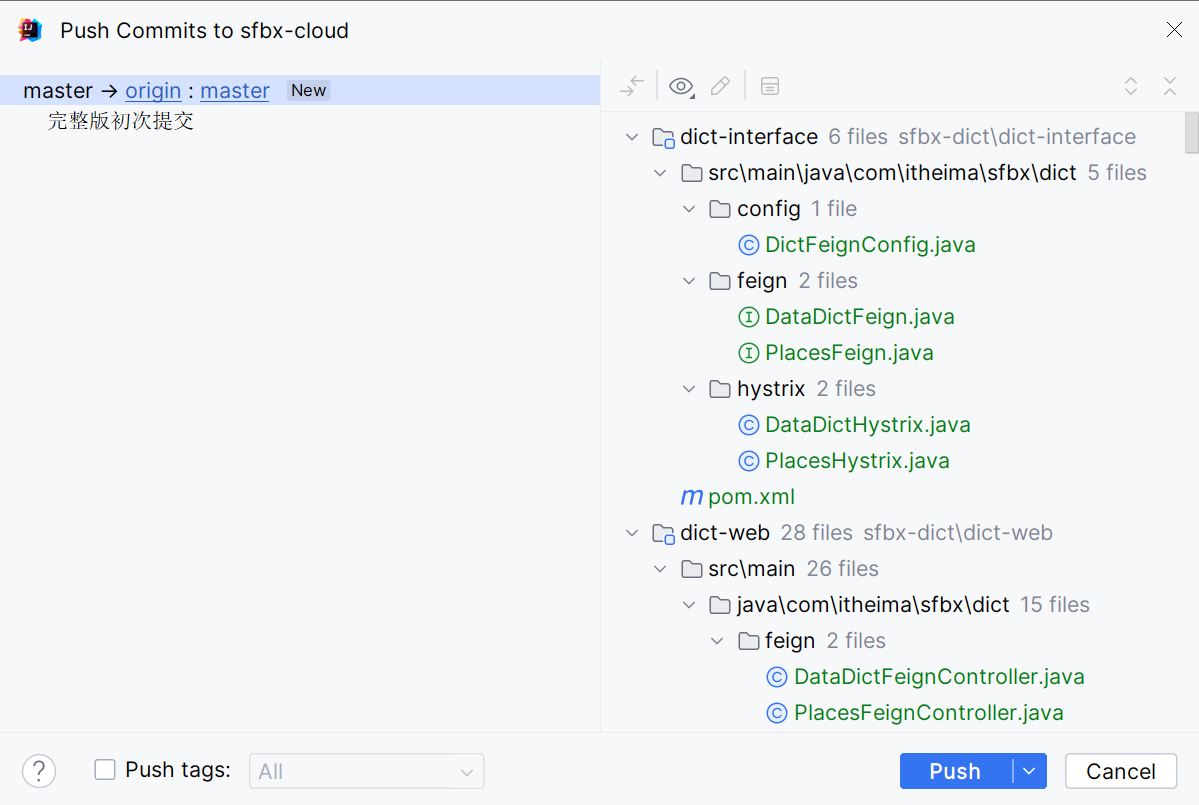
**5.2、关联gitee远程仓库**

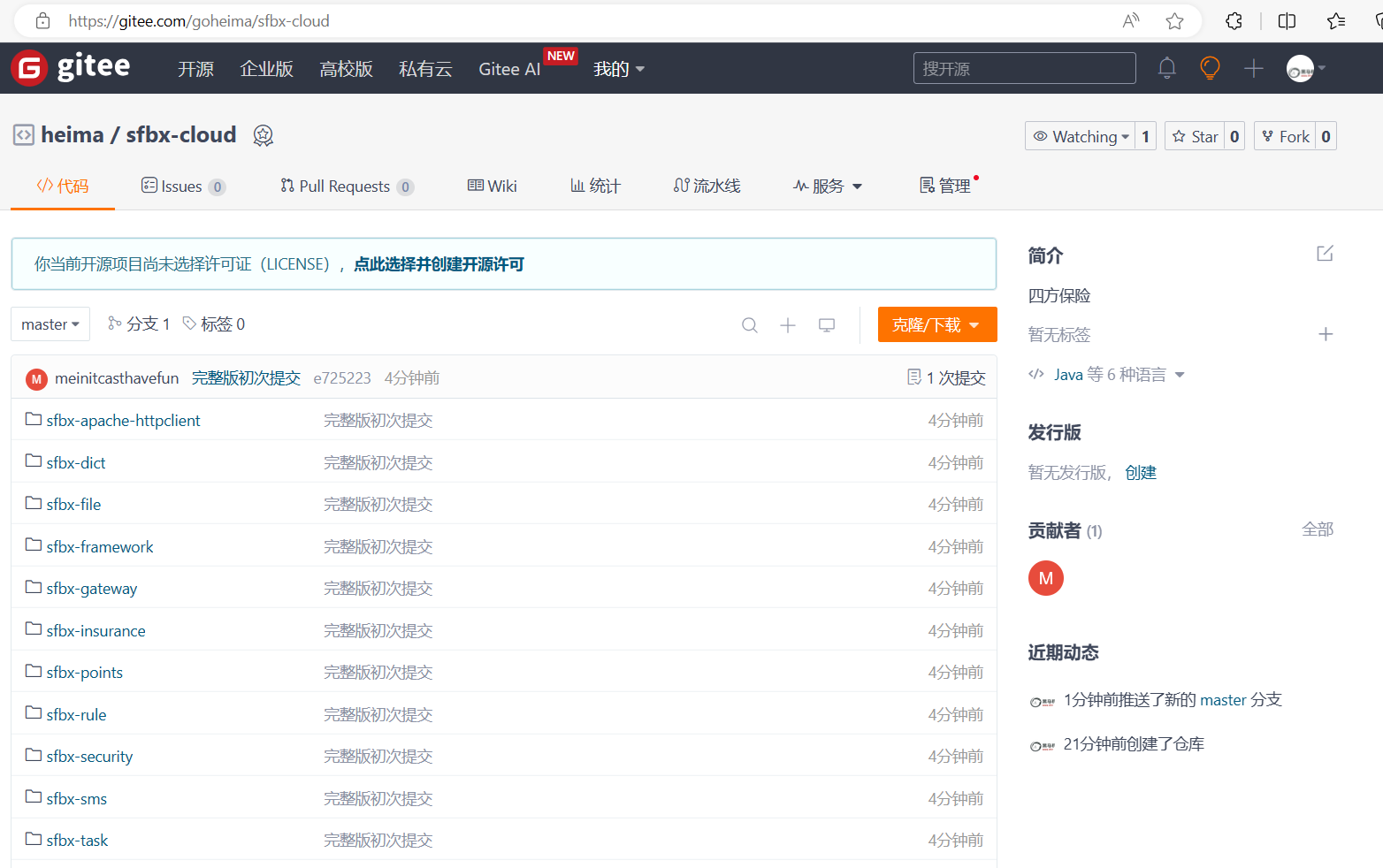












|  |
| --- |
| 注意：  如果在推送过程中出现要输入 账号、密码的界面；那么直接输入gitee网站的账号密码即可 |