

# 病例收集项目部署文档

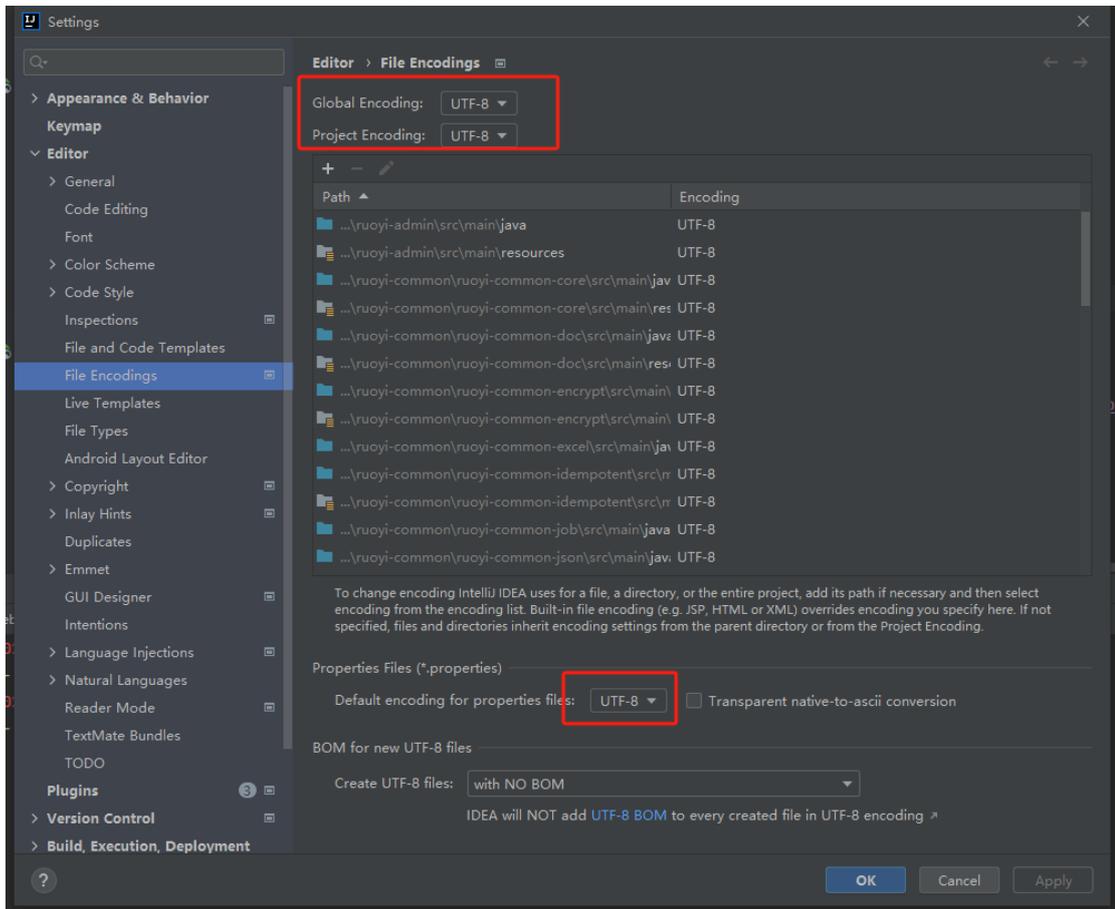
版本	编写人	编写时间
V1.0.0	吴志刚	2023/12/15

## 服务器环境配置要求

CPU	内存	宽带
4核	8G	5M

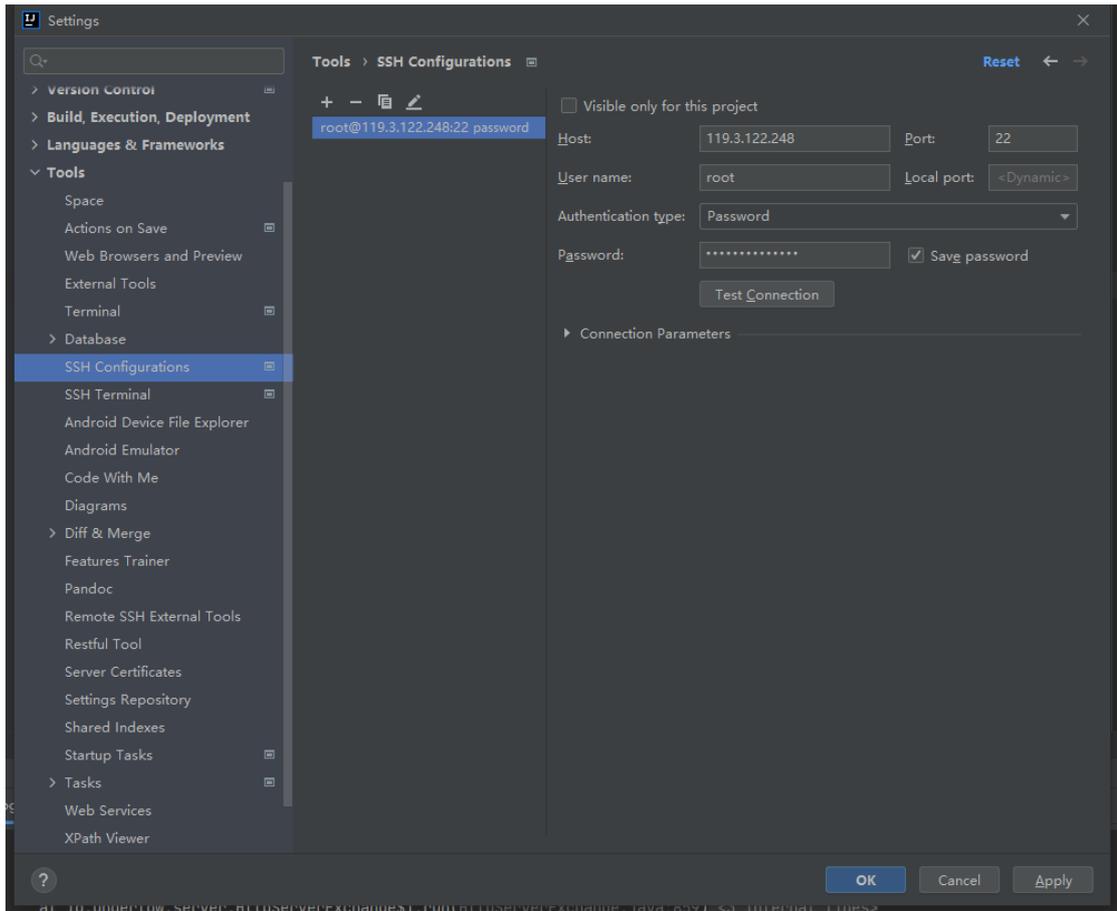
## 一、idea 环境配置

### 1. Settings -> Editor -> File Encodings 编码调整全部调整为 UTF-8

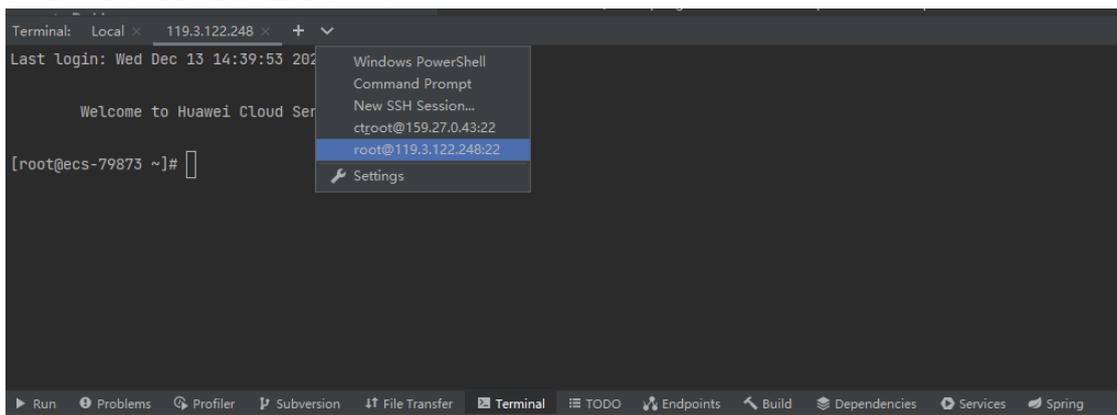


### 2. 配置服务器 SSH 连接

进入 Settings -> Tools -> SSH Configurations 点击加号创建 SSH 连接配置  
填写 服务器 IP 用户名 密码 端口号 点击 Test Connection 测试连接



使用 Terminal 工具，点击箭头找到上方创建的 SSH 连接。选择即可进入 SSH 连接界面在这里可以对服务器进行命令操作。



## 二、docker 安装

root 账号 Terminal 控制台或者 Xshell 等工具操作

1. 卸载 docker 旧版本

```
yum remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-selinux docker-engine-selinux docker-engine
```

2. 安装相关工具类

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

3. 配置 docker 仓库

```
yum-config-manager --add-repo  
https://download.docker.com/linux/centos/docker-ce.repo
```

4. 配置 aliyun 镜像仓库

```
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

5. 安装 docker

```
yum install docker-ce
```

6. 验证 docker 安装成功

```
启动 docker  
systemctl start docker
```

```
验证 docker  
docker run hello-world
```

出现以下内容则表示安装成功

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest:
sha256:c79d06dfdfd3d3eb04cafd0dc2bacab0992ebc243e083cabe208bac4dd7
759e0
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!  
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

## 7. 设置开机启动

```
chkconfig docker on
```

## 8. 设置加速镜像和存储目录(存储目录根据服务器实际情况配置, 选择磁盘空间大的目录)

```
vim /etc/docker/daemon.json
```

添加内容

```
{  
    "registry-mirrors": ["http://hub-mirror.c.163.com"],  
    "data-root": "/data/docker"  
}
```

#### 9. 加载配置

```
systemctl daemon-reload
```

#### 10. 重启 docker

```
systemctl restart docker
```

### 三、安装 docker-compose

#### 1. 安装

github 下载

```
curl -SL  
https://github.com/docker/compose/releases/download/v2.21.0/docker-  
compose-linux-x86_64 -o /usr/local/bin/docker-compose
```

如果无法下载，docker-compose-linux-x86\_64 手动上传至/usr/local/bin 并重命名（资源文件包含）

```
mv docker-compose-linux-x86_64 docker-compose
```

#### 2. 文件授权

```
chmod +x /usr/local/bin/docker-compose
```

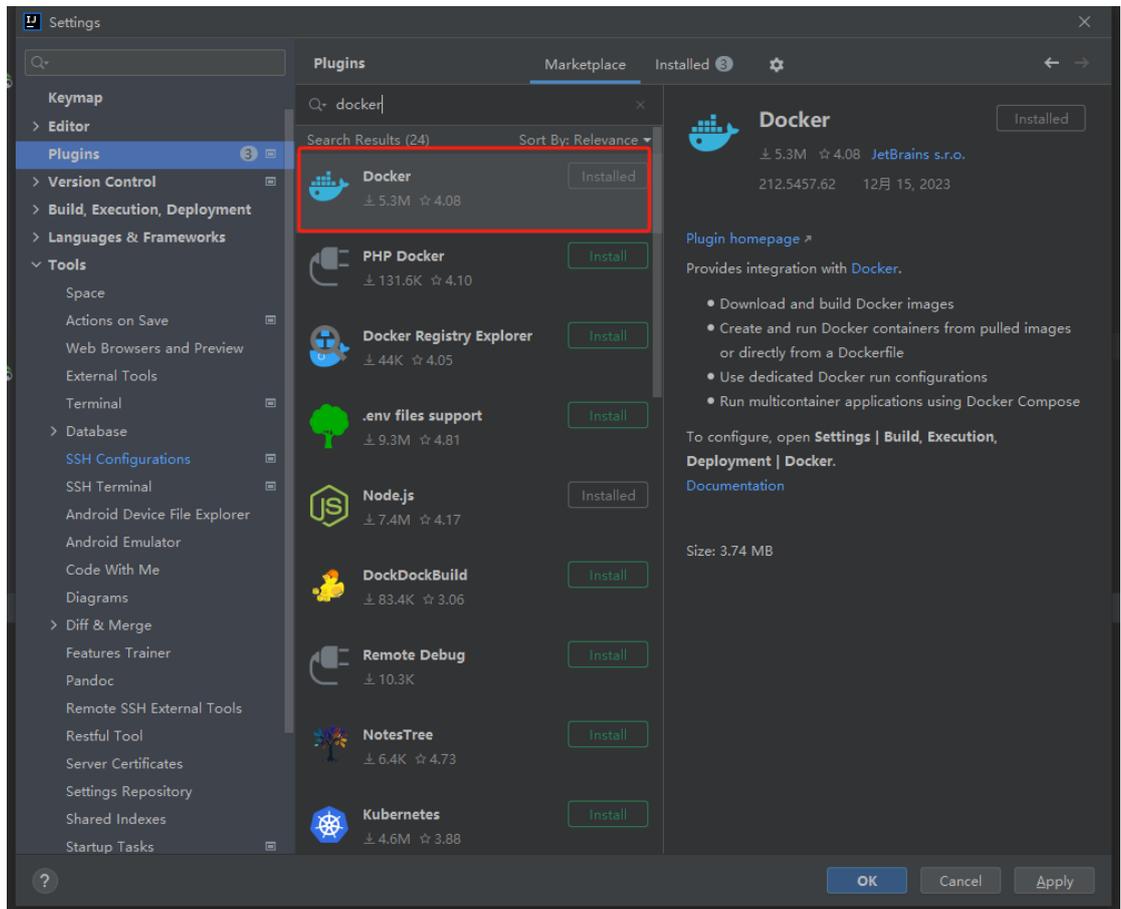
#### 3. 查看版本信息

```
docker compose version
```

### 四、Idea 连接 docker

## 1. idea 安装 docker 插件

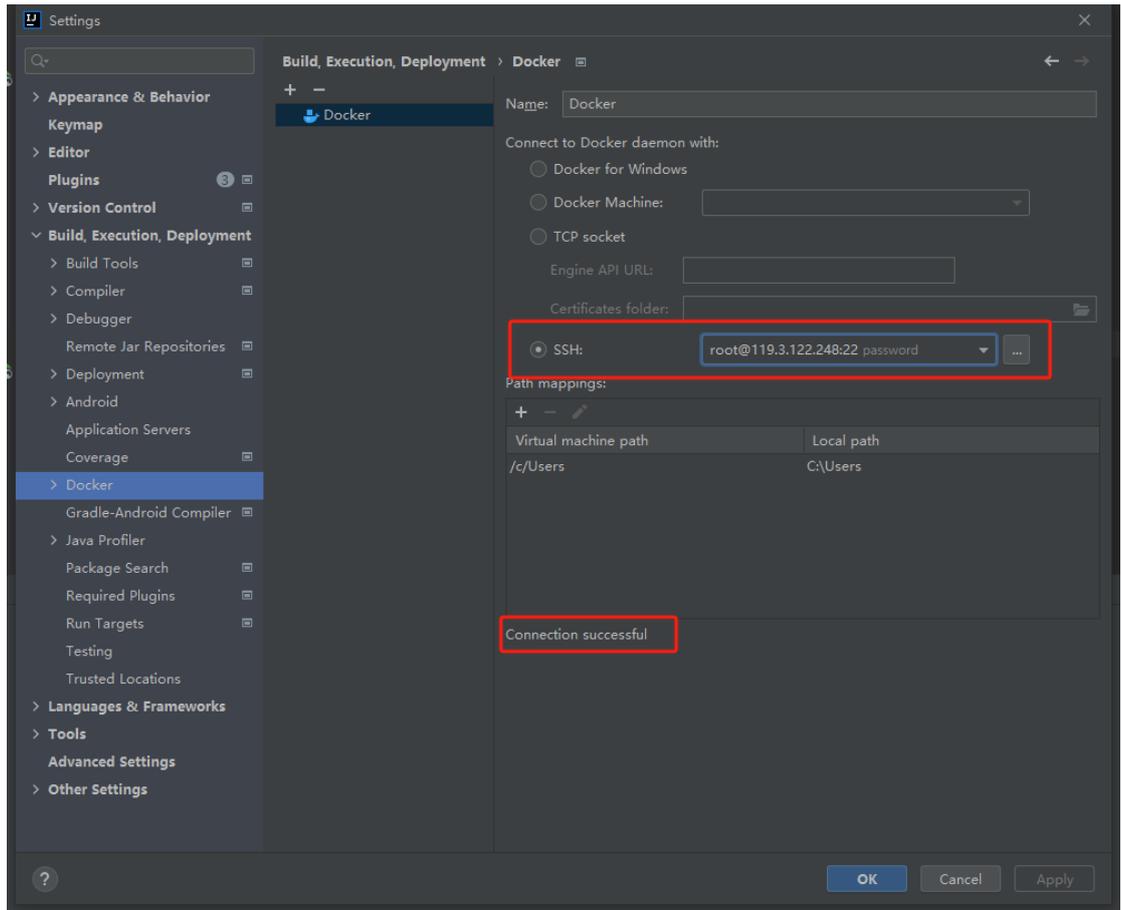
File-> Settings -> Plugins 选择 Docker 插件并安装



安装完成后，重启 idea。

## 2. 配置 Docker 连接

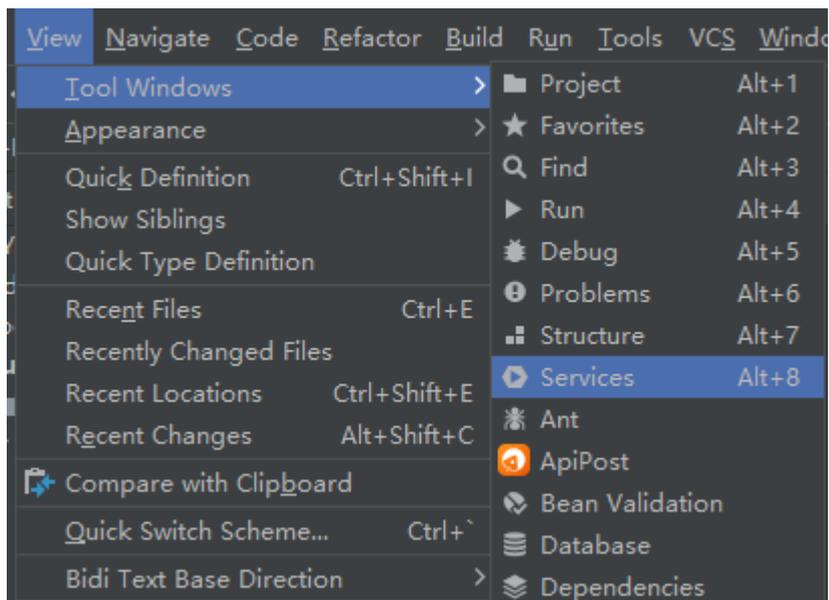
File->Settings->Build...->Docker, 选择 SSH, 下拉选择上面创建的 SSH 服务器连接



连接成功显示 Connection successful

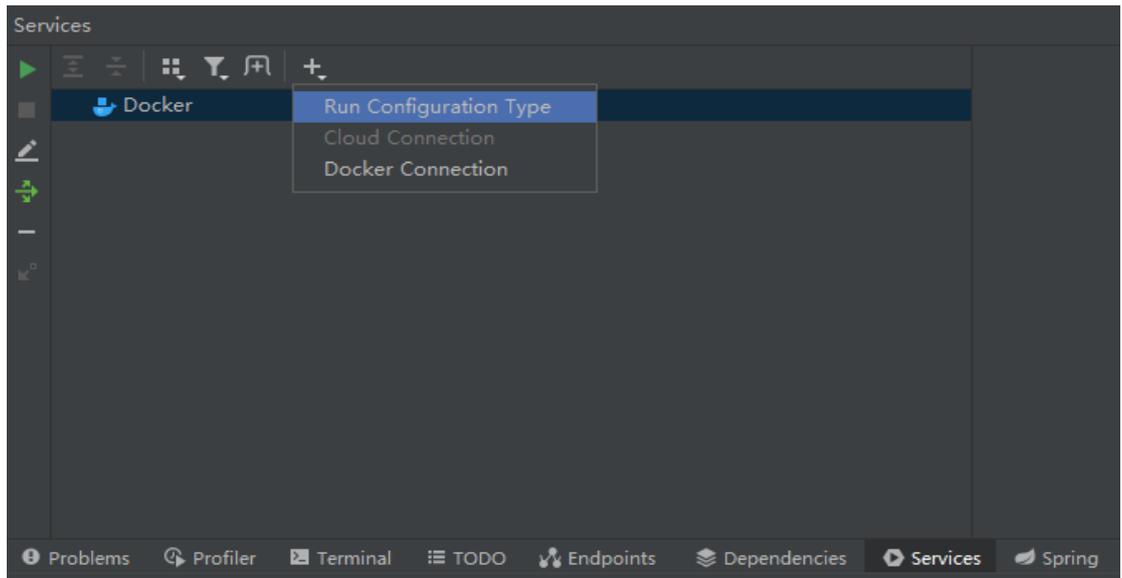
### 3. 配置运行看板

View->Tool Windows->Services

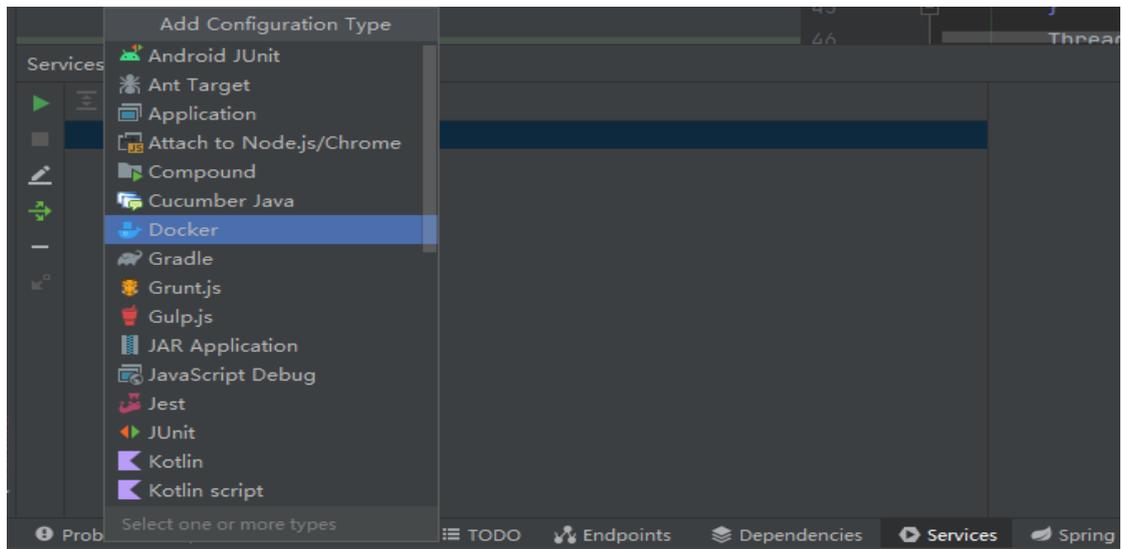


### 4. 配置 spring 与 docker 看板

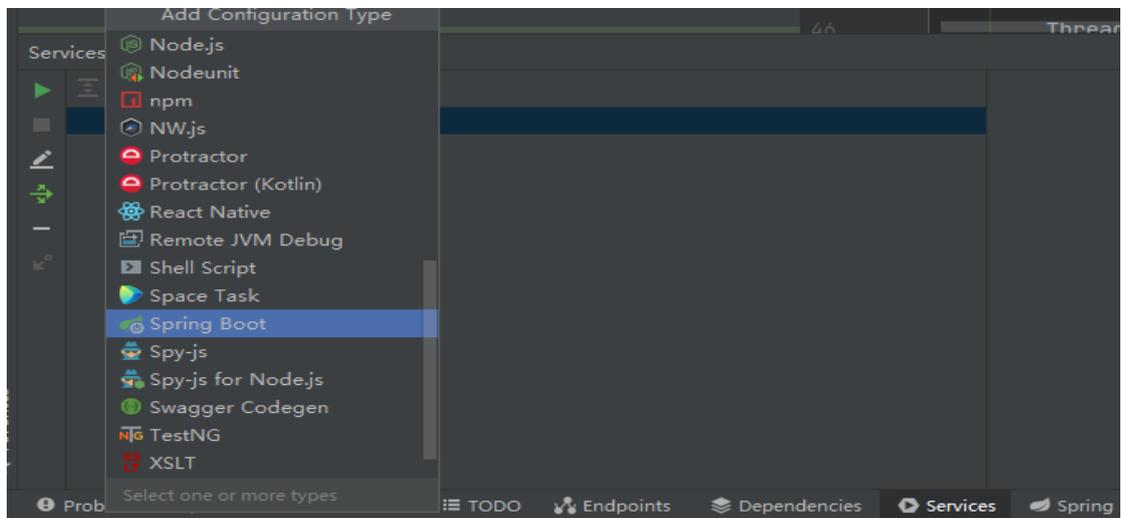
Services 窗口点击+号下拉，选择 Run Configuration Type



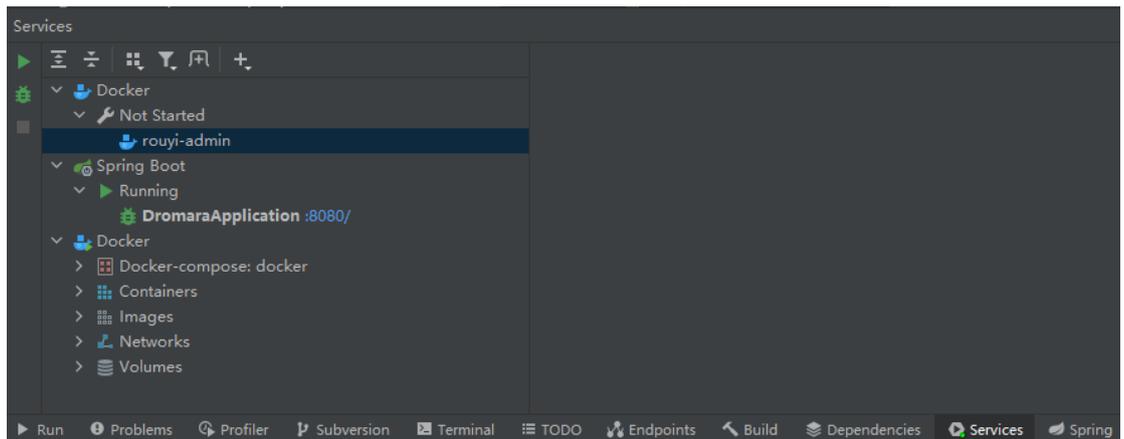
选择 Docker



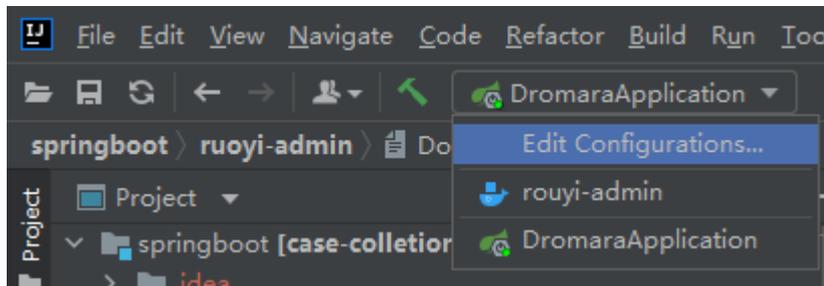
选择 Spring Boot



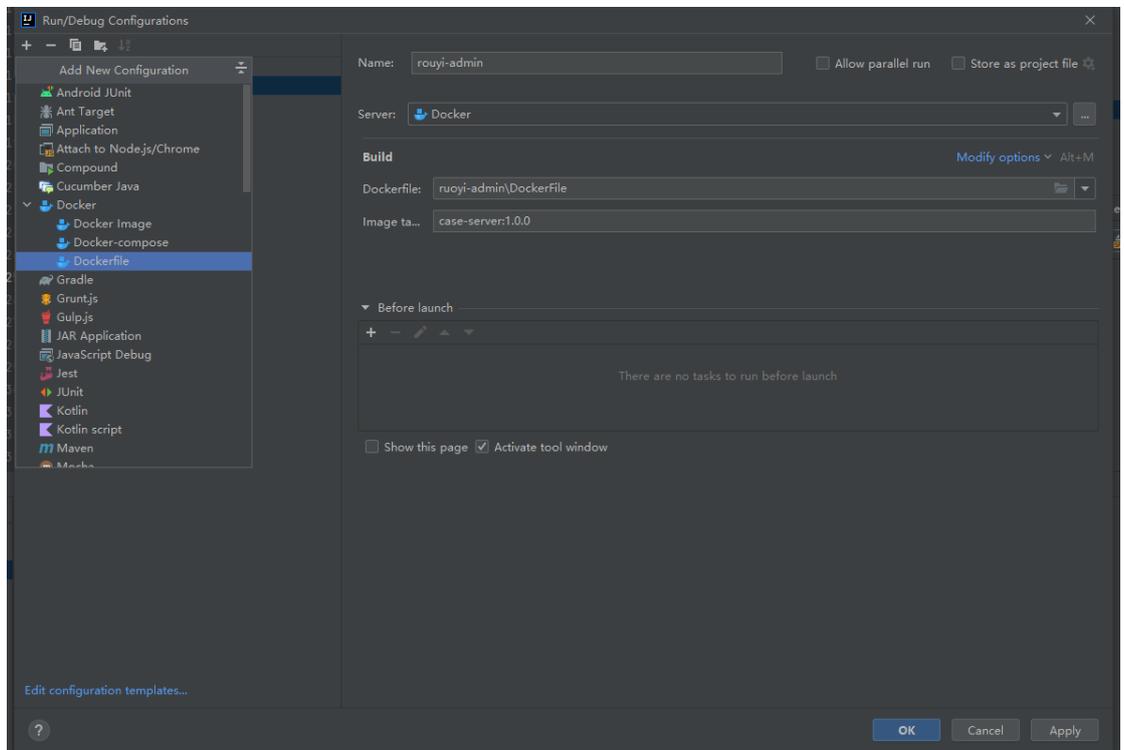
选择完成后 Services 窗口



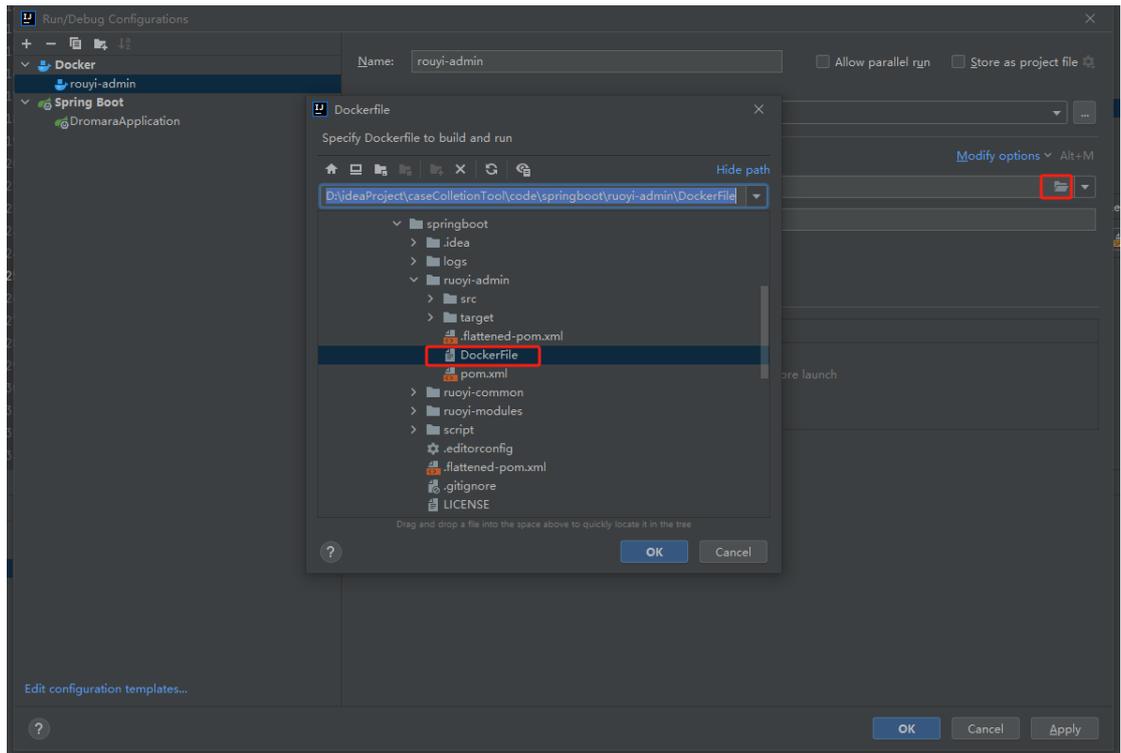
如果没有显示 Docker, 点击 Edit Configurations...



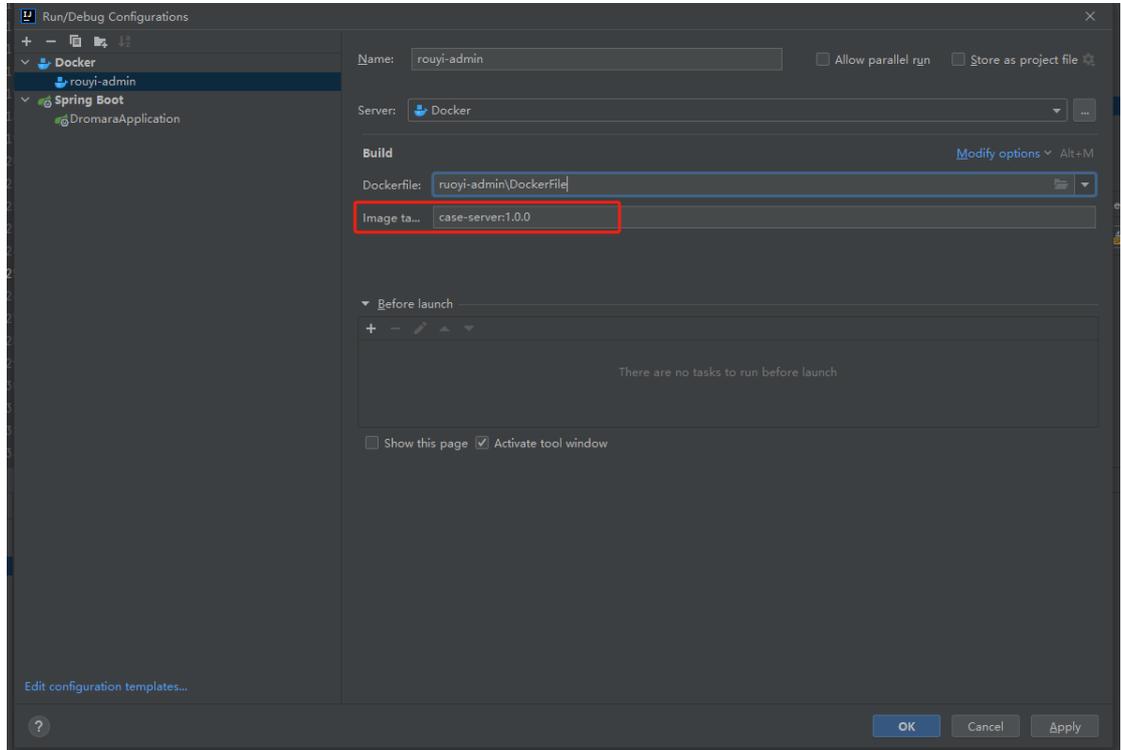
点击+号, 选择 Docker->Dockerfile



Server 选择已创建的 docker 连接，选择项目中的 DockerFile 文件



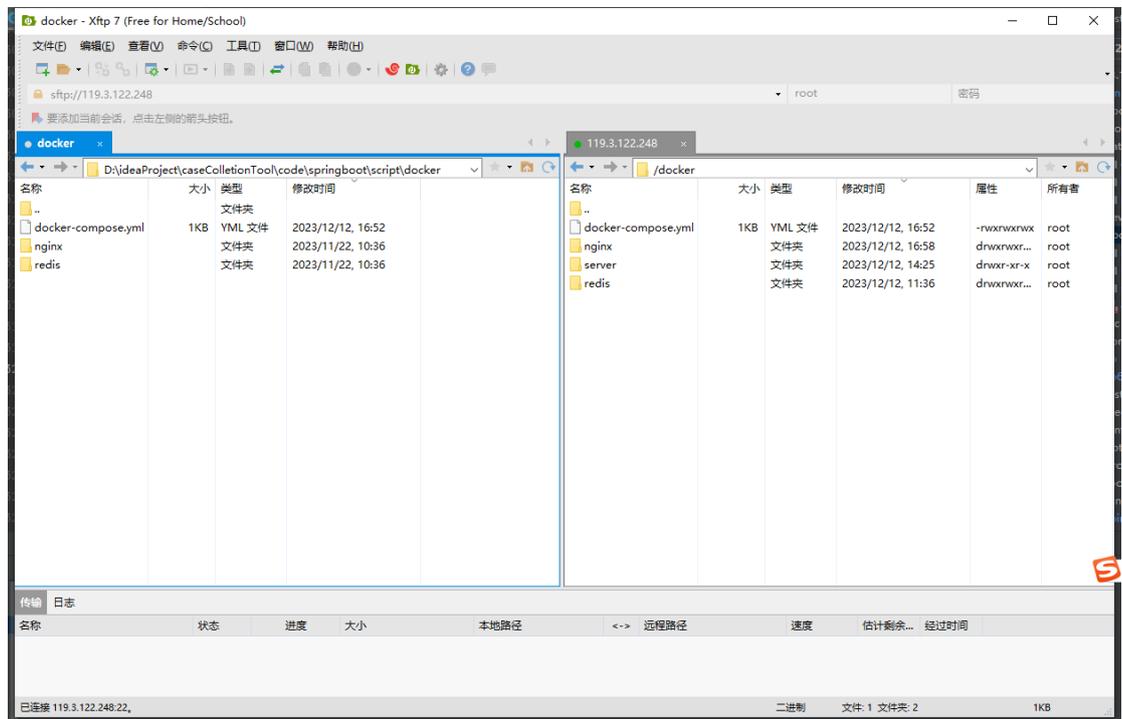
填写镜像名称 case-server:1.0.0



确认并保存。

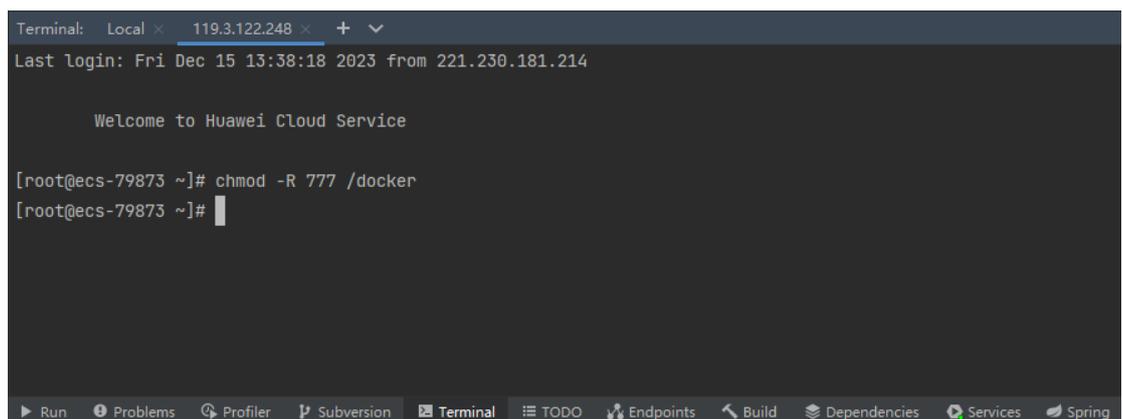
## 五、Docker 配置文件上传服务器

### 1. 项目文件夹 script 下的 docker 文件夹上传服务器根目录



### 2. 文件夹授权

```
chmod -R 777 /docker
```

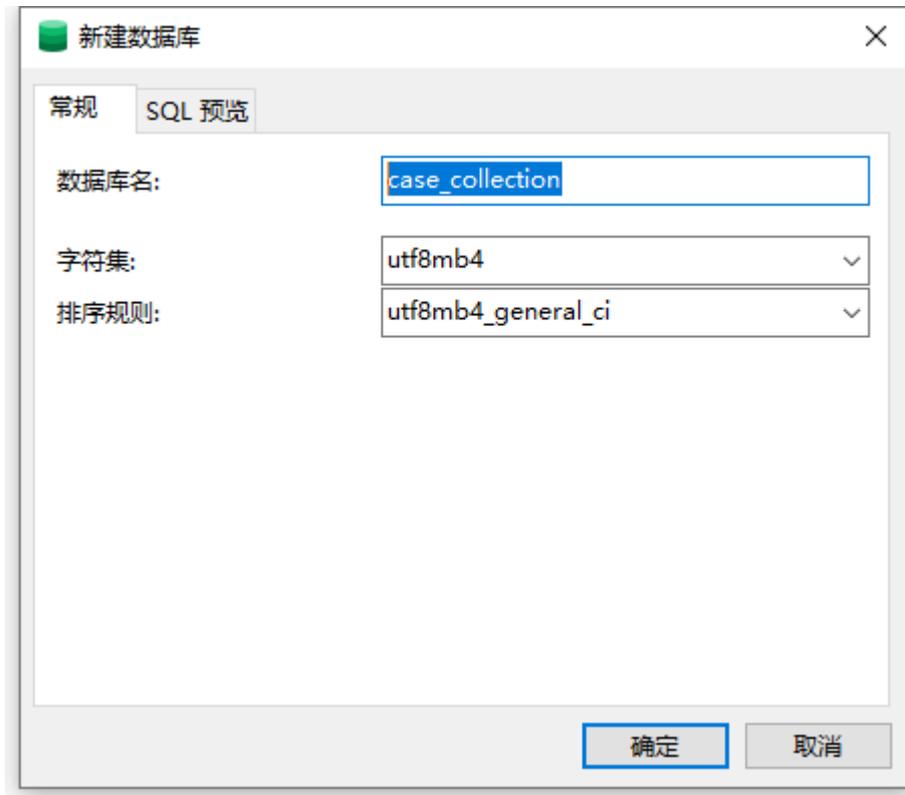


一定要确保目录 `/docker` 及其所有子目录 具有写权限 如果后续出现权限异常问题 重新执行一遍分配权限。

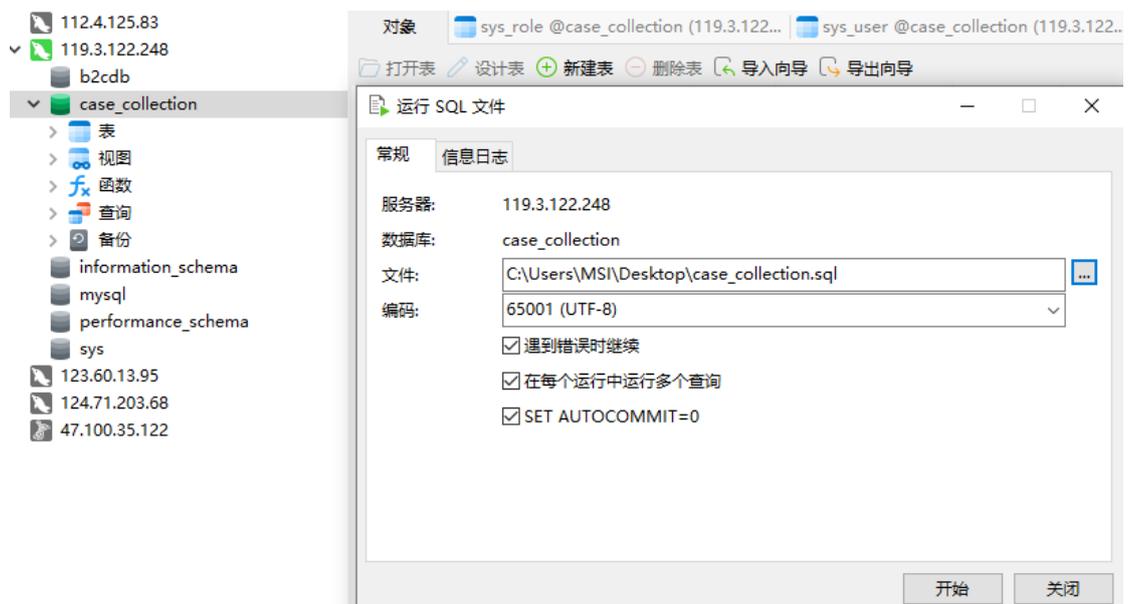
## 六、构建应用镜像

### 1. 数据库初始化

生产库中创建 case\_collection 数据库



运行 case\_collection.sql 初始化数据库文件



## 2. 项目生产环境配置文件修改

数据库连接信息替换生产数据库（数据库已初始化）

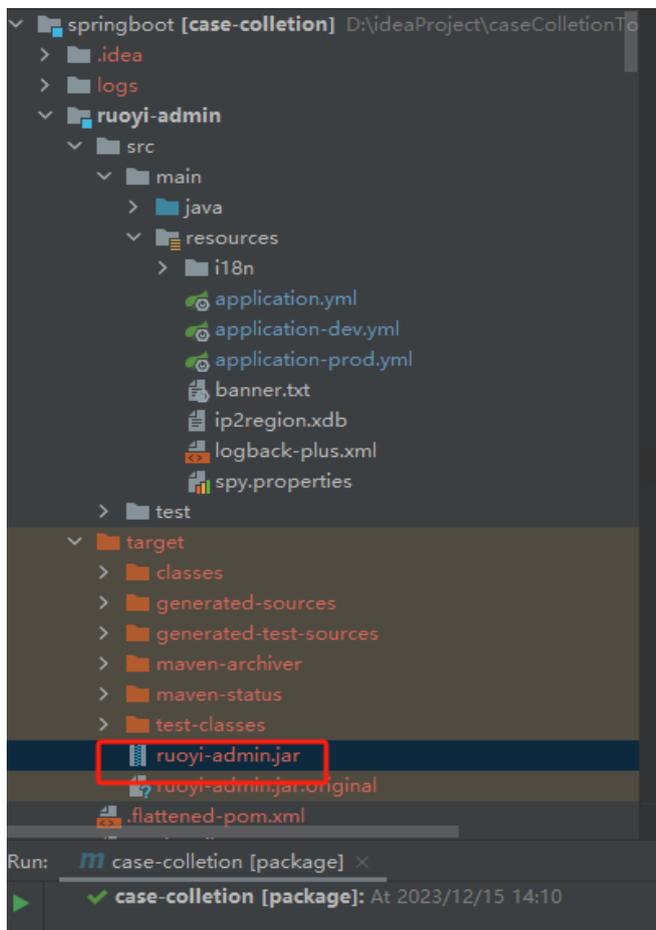
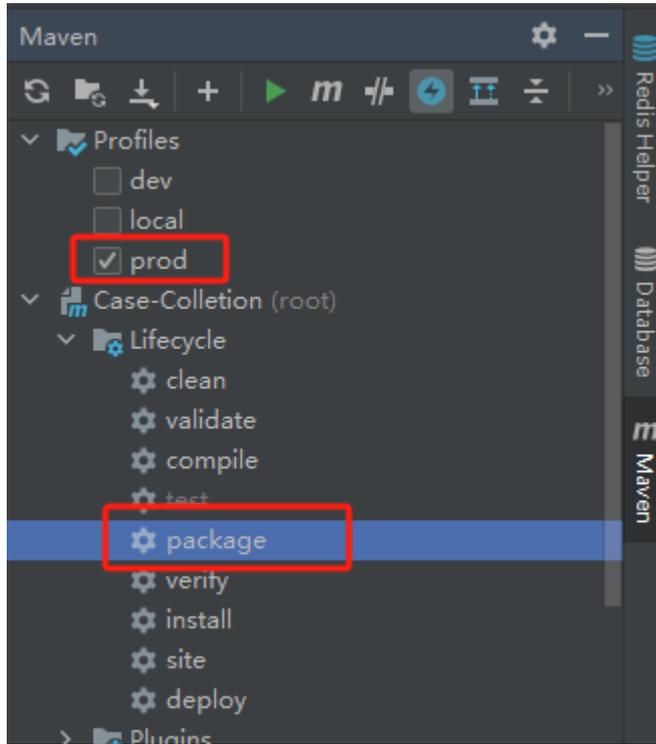
```
--- # 数据源配置
spring:
  datasource:
    type: com.zaxxer.hikari.HikariDataSource
    # 动态数据源
    dynamic:
      # 性能分析插件(有性能损耗 不建议生产环境使用)
      p6spy: false
      # 设置默认的数据源或者数据源组, 默认值即为 master
      primary: master
      # 严格模式 匹配不到数据源则报错
      strict: true
    datasource:
      # 主库数据源
      master:
        type: ${spring.datasource.type}
        driverClassName: com.mysql.cj.jdbc.Driver
        # rewriteBatchedStatements=true 批处理优化 大幅提升批量插入更新删除性能(对数据库有性能损耗 使用批量操作应考虑性能问题)
        url: jdbc:mysql://119.3.122.248:3306/case_collection?useUnicode=true&characterEncoding=utf8&zeroDateBehavior=
        username: root
        password: Sa
```

客户端和管理端 IP 地址替换生产环境 IP 或者域名:

```
nginx.conf x docker-compose.yml x application-prod.yml x application.yml x
150     accessKeySecret: xxxxxxxx
151     # 短信签名
152     signature: 测试
153     # 短信sdkAppId
154     sdkAppId: appid
155     # 地域信息默认为 ap-guangzhou 如无特殊改变可不用设置
156     territory: ap-guangzhou
157
158     --- # 系统基础登录地址
159     login:
160     address:
161     # 病例上传端地址
162     user: http://119.3.122.248:8062/#/
163     # 学会后台管理端地址
164     management: http://119.3.122.248:8061/login
165
```

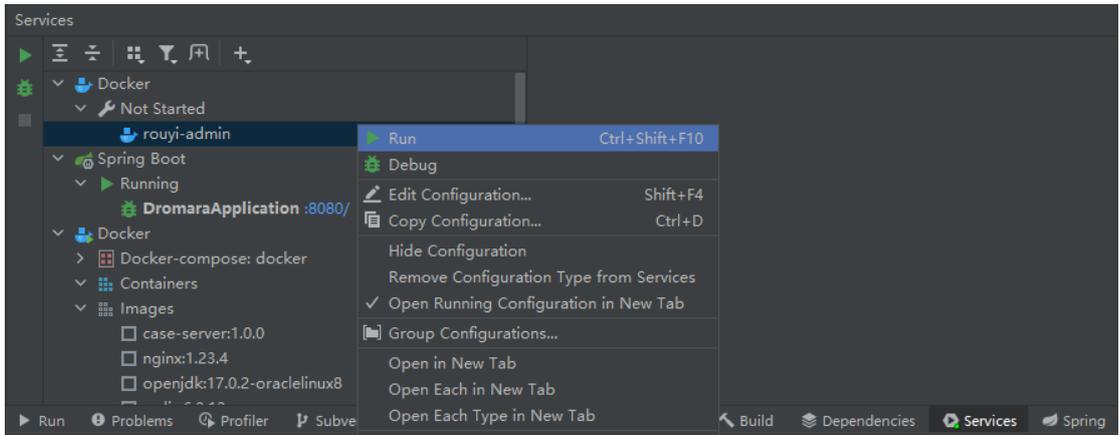
### 3. 项目 maven 打包成 jar 包

选择 prod 环境，双击 package 生成 jar 包

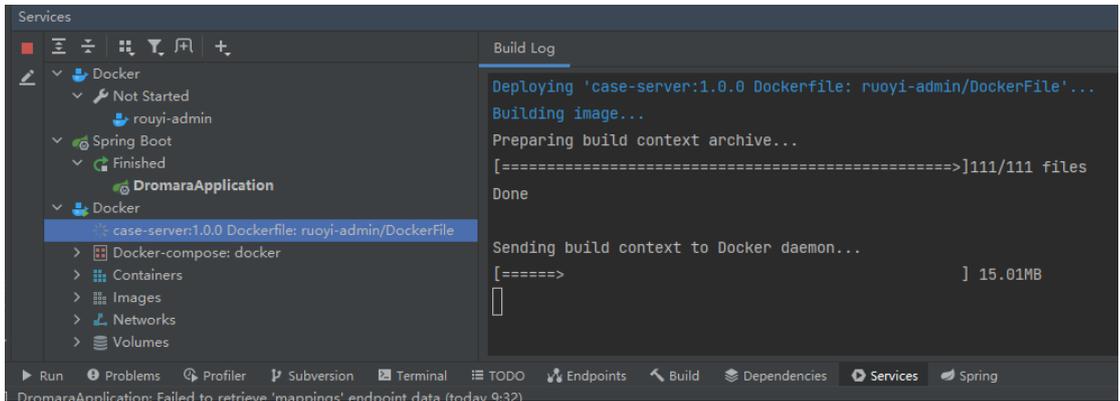


#### 4. 执行镜像构建

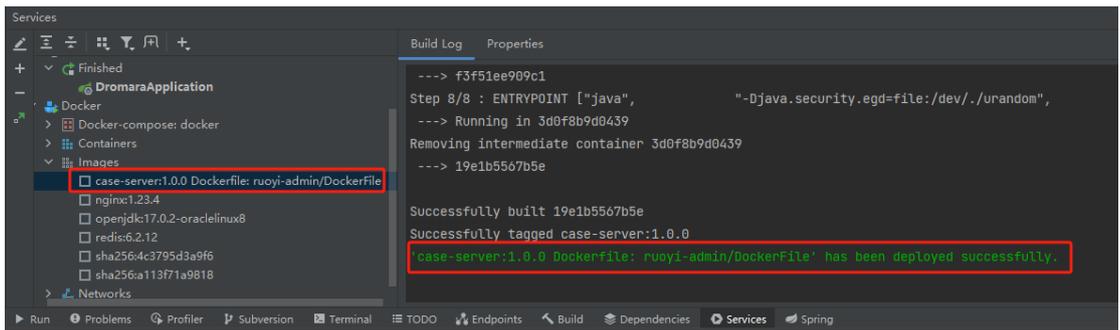
右击选择 run 执行构建，第一次需要下载 jdk 等依赖项，可能会慢一点。



#### 构建中



#### 构建完成



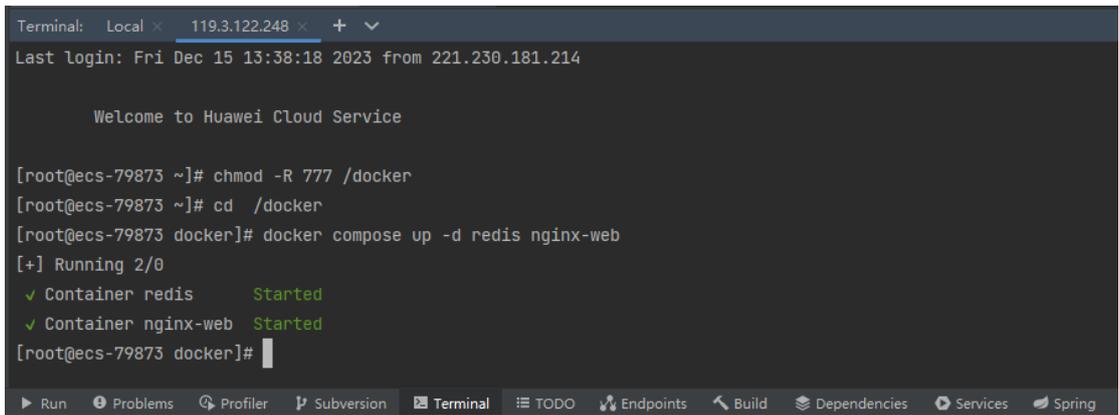
## 七、服务器 docker 服务创建

### 1. Redis、nginx 基础服务启动

```
cd /docker
```

```
docker compose up -d redis nginx-web
```

服务创建成功



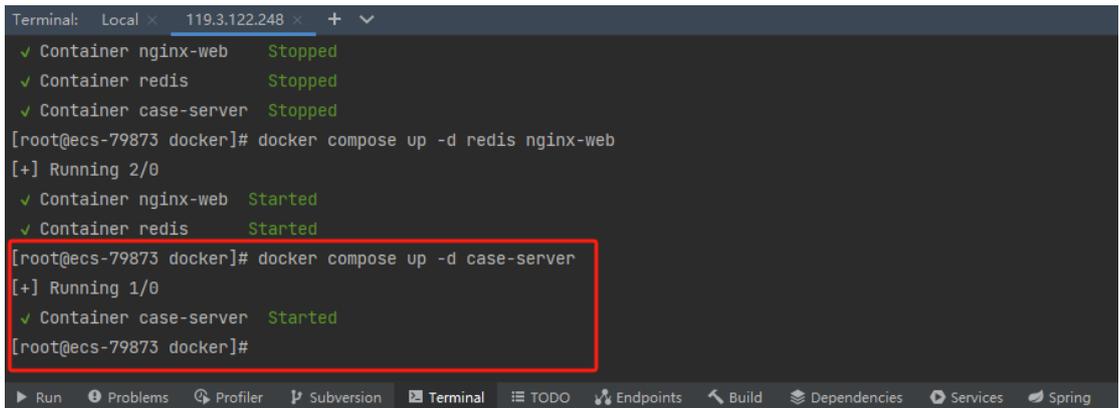
```
Terminal: Local x 119.3.122.248 x + v
Last login: Fri Dec 15 13:38:18 2023 from 221.230.181.214

Welcome to Huawei Cloud Service

[root@ecs-79873 ~]# chmod -R 777 /docker
[root@ecs-79873 ~]# cd /docker
[root@ecs-79873 docker]# docker compose up -d redis nginx-web
[+] Running 2/0
  ✓ Container redis      Started
  ✓ Container nginx-web Started
[root@ecs-79873 docker]#
```

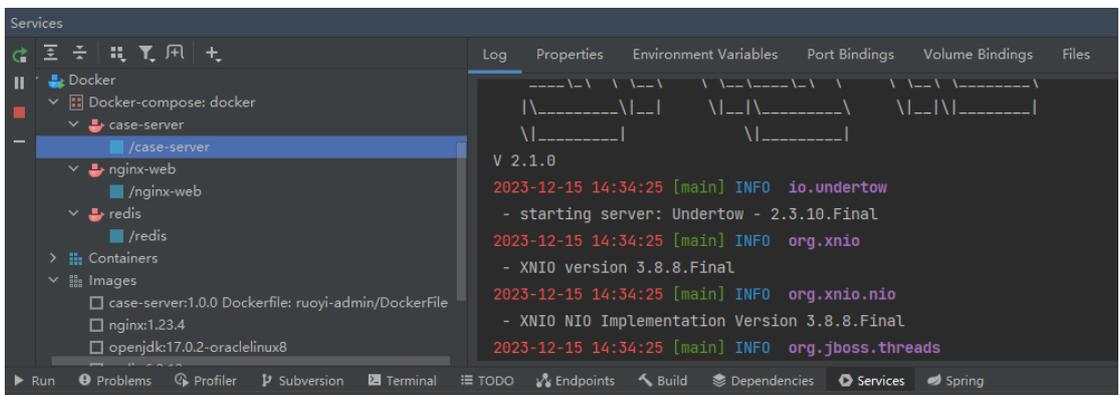
### 2. 业务服务创建

```
docker compose up -d case-server
```



```
Terminal: Local x 119.3.122.248 x + v
  ✓ Container nginx-web  Stopped
  ✓ Container redis      Stopped
  ✓ Container case-server Stopped
[root@ecs-79873 docker]# docker compose up -d redis nginx-web
[+] Running 2/0
  ✓ Container nginx-web Started
  ✓ Container redis      Started
[root@ecs-79873 docker]# docker compose up -d case-server
[+] Running 1/0
  ✓ Container case-server Started
[root@ecs-79873 docker]#
```

### 3. 服务创建成功

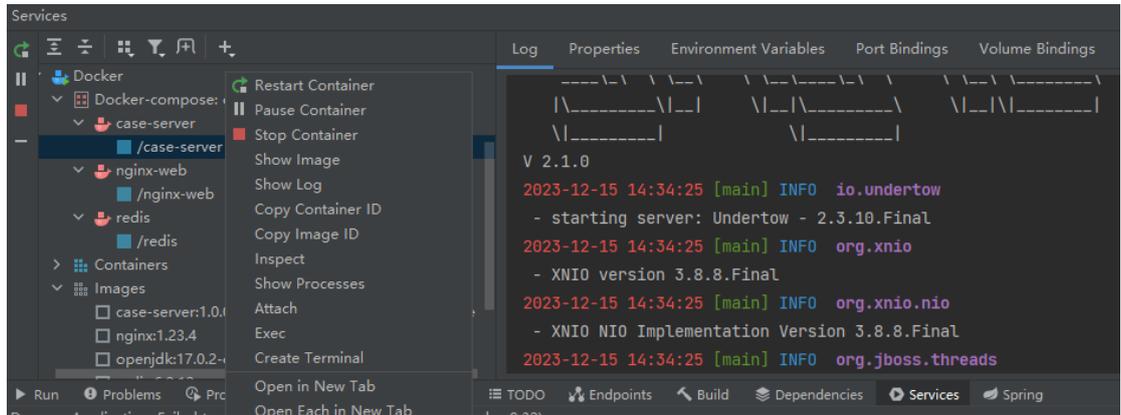


```
Services
├── Docker
│   ├── Docker-compose: docker
│   │   ├── case-server
│   │   │   ├── /case-server
│   │   │   ├── nginx-web
│   │   │   │   ├── /nginx-web
│   │   │   │   └── redis
│   │   │   └── /redis
│   └── Containers
│       ├── case-server:1.0.0 Dockerfile: ruoyi-admin/DockerFile
│       ├── nginx:1.23.4
│       └── openjdk:17.0.2-oraclelinux8
└── Images

Log
-----
V 2.1.0
2023-12-15 14:34:25 [main] INFO io.undertow
- starting server: Undertow - 2.3.10.Final
2023-12-15 14:34:25 [main] INFO org.xnio
- XNIO version 3.8.8.Final
2023-12-15 14:34:25 [main] INFO org.xnio.nio
- XNIO NIO Implementation Version 3.8.8.Final
2023-12-15 14:34:25 [main] INFO org.jboss.threads
```

#### 4. Docker 其他操作

右击查看可执行操作



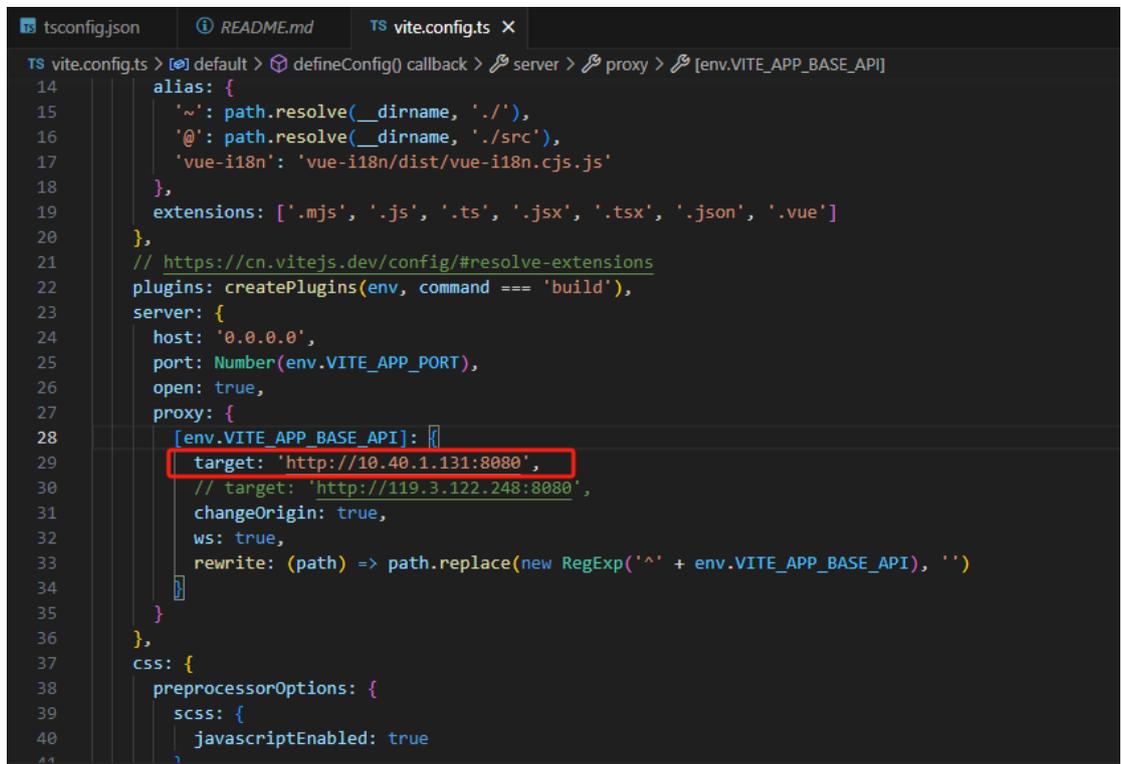
#### 八、管理端前段部署

前端环境要求:

nodejs 16 18

npm 8.X

1. 修改后端服务地址, 替换为生产后端地址。



## 2. 前端项目打包

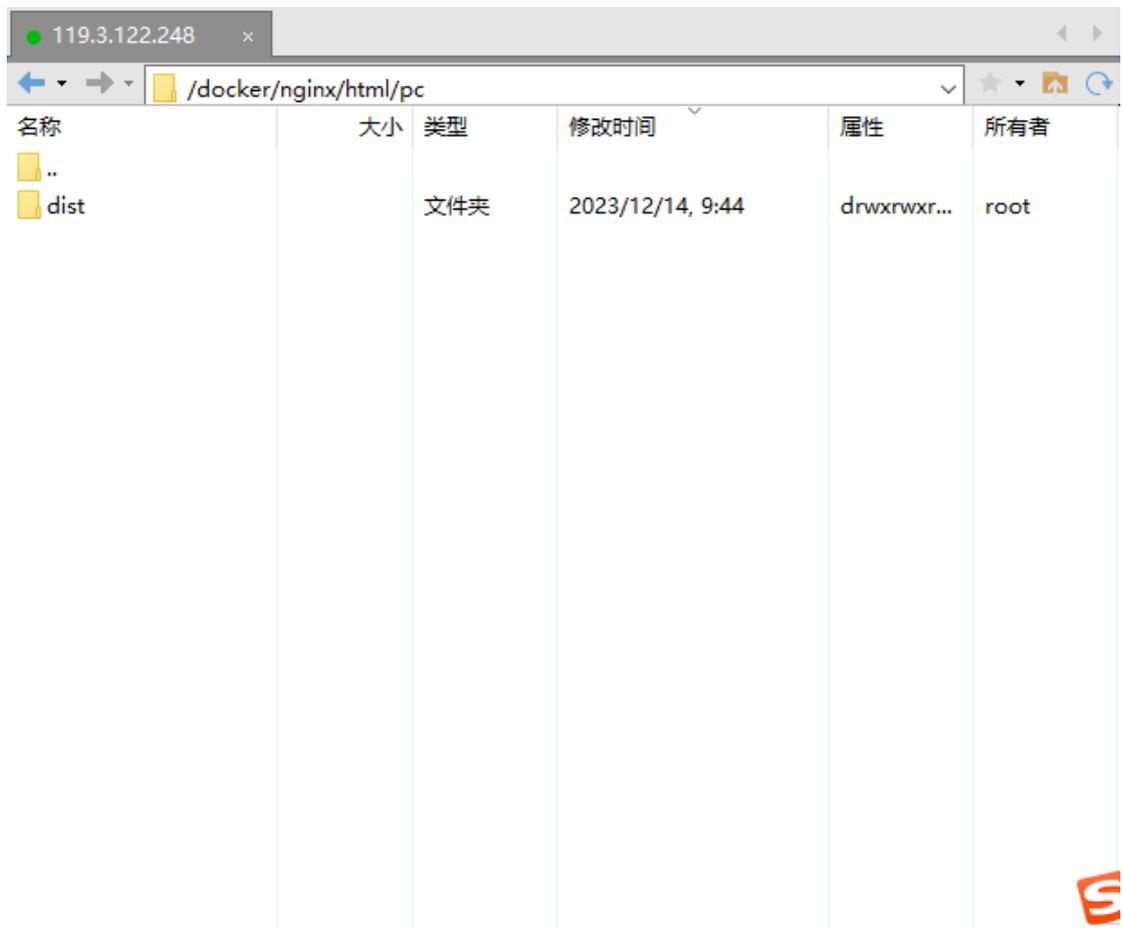
安装依赖

```
npm install --registry=https://registry.npmmirror.com
```

打包正式环境

```
npm run build:prod
```

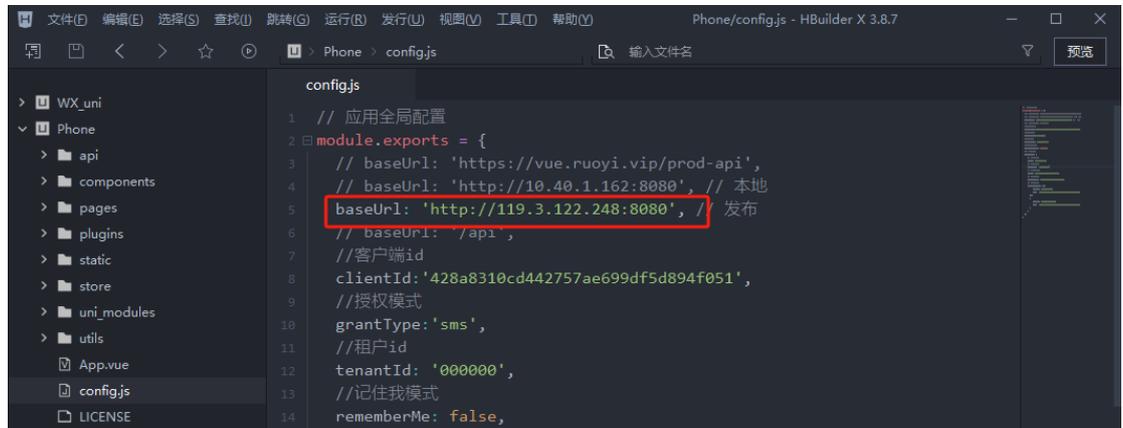
## 3. 前端生成的 dist 文件夹上传服务器/docker/nginx/html/pc 目录



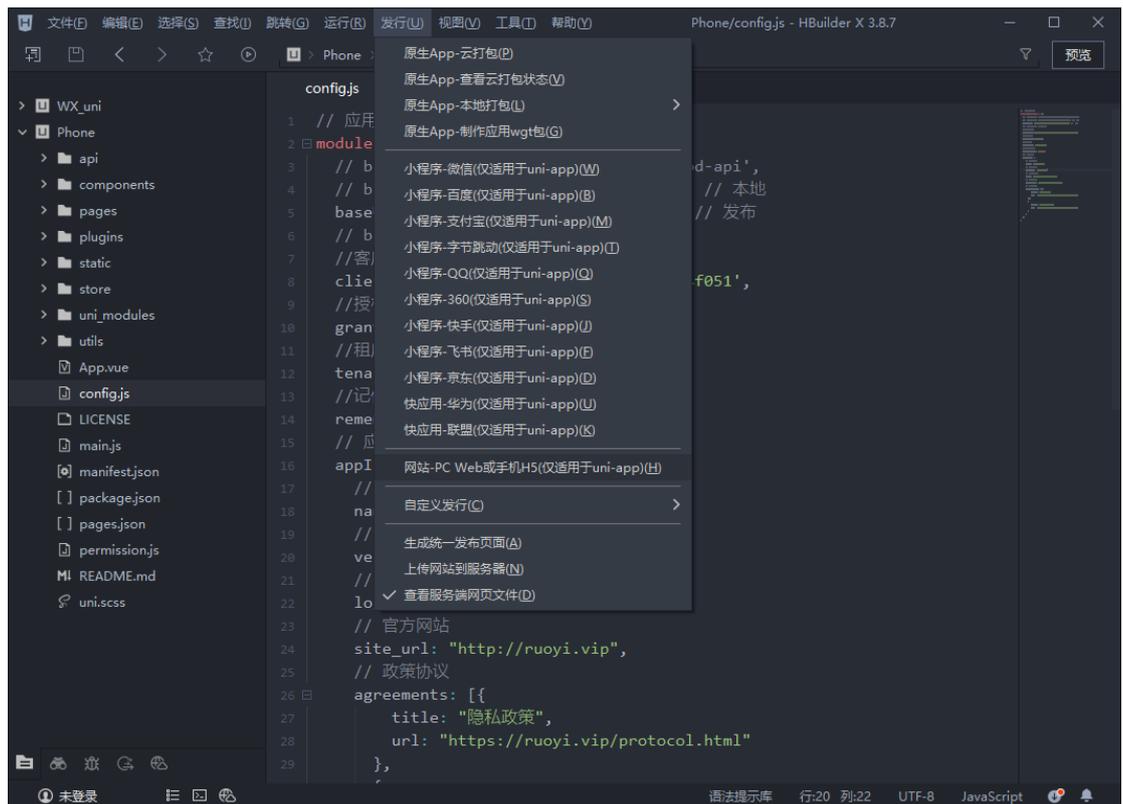
## 九、客户端 H5 部署

使用 HBuilderX 开发工具打包

### 1. 替换后端服务地址为生产环境地址



### 2. 点击发行，选择 网站-PC Web 或手机 H5



### 3. 项目打包

输入网站标题，点击发行

## H5发行 cli程序化部署教程

网站标题

网站域名

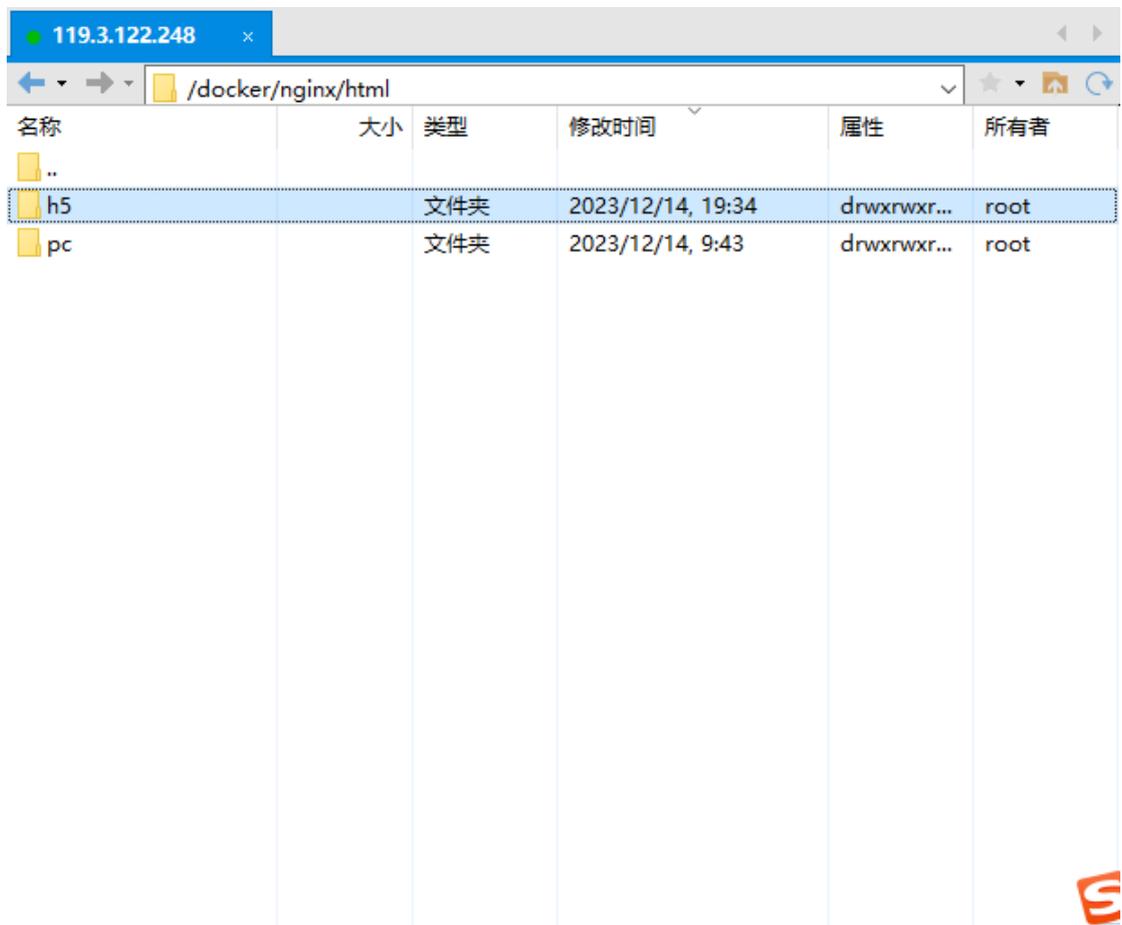
生成sourcemap (可用于uni统计的错误分析) [详情](#)

将编译后的资源部署到前端网页托管 [详情](#)

欢迎开通 [uniAD](#) 广告进行变现, [申请入口](#) | [开发文档](#)

开发者控制台打包成功后，会返回打包的文件路径。

### 4. 返回文件路径下的 h5 文件夹上传服务器/docker/nginx/html 目录。



名称	大小	类型	修改时间	属性	所有者
..					
<b>h5</b>		文件夹	2023/12/14, 19:34	drwxrwxr...	root
pc		文件夹	2023/12/14, 9:43	drwxrwxr...	root

## 十、项目登录

### 1. 后台管理系统

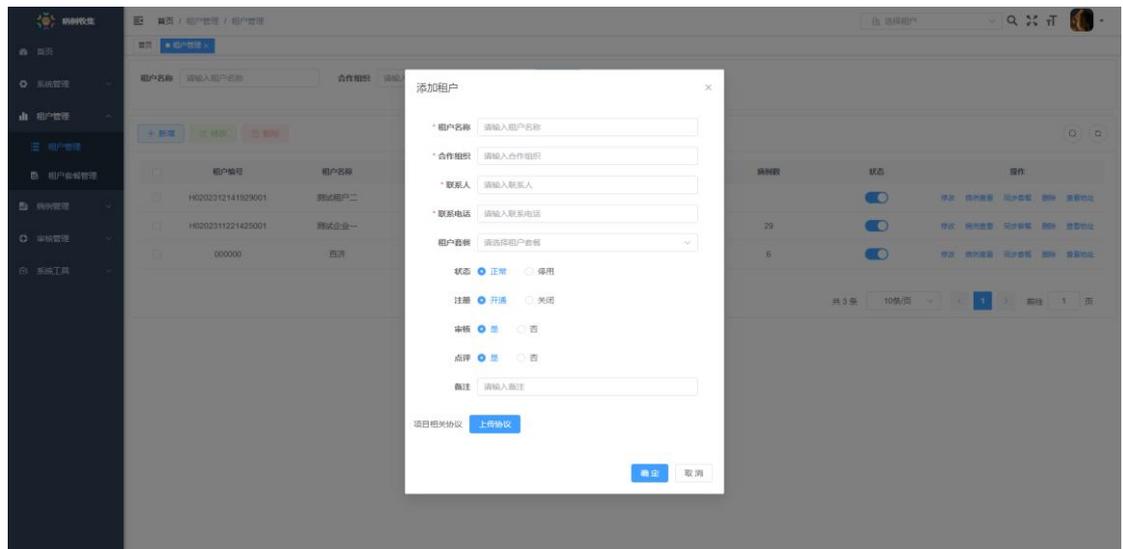
登录地址：

<http://ip:8061/login?tenantId=000000>

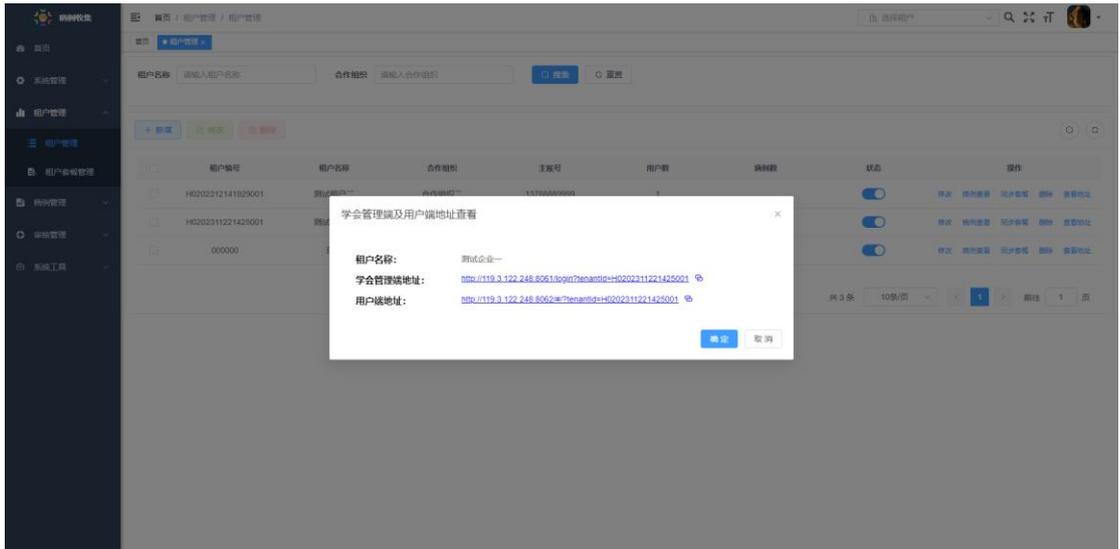
ip 为生产环境服务器 ip 或者域名，tenantId 为租户 ID，超级管理员租户 ID 固定为 000000。



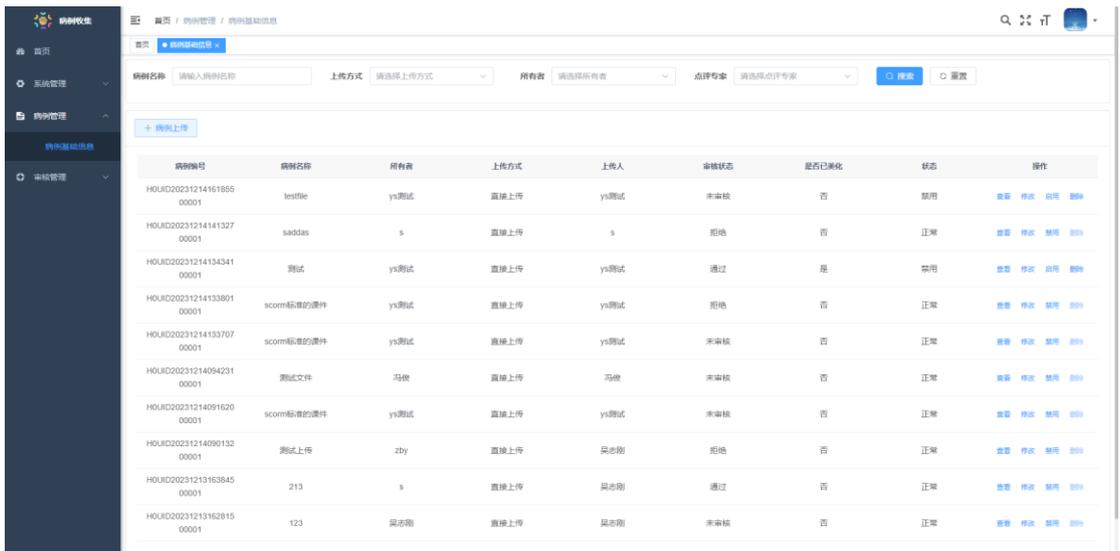
进入系统，选择租户管理，点击新增租户，创建新租户。



## 获取学会管理端和客户端地址



2. 复制地址登录学会管理端和客户端。  
学会管理端:



用户端:

16:28

5G



首页



病例管理

## 所有病例

3

已上传病例

1

符合标准病例

1

不符合标准病例

123

审核中



testfile1.jpg



客户端PC上传测试一

已美化



新建 Microsoft PowerPoint 演示文稿.pptx



测试ios上传病例一

未通过



FD405608-C7FD-4AEE-8E7E-07F8D5D0591... >



首页



新增病例



我的



