# 前端部署

## 文件结构

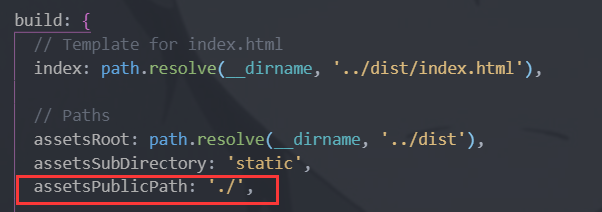
前端的项目文件结构如下：

|  |
| --- |
| ├─build //打包配置文件  ├─config //项目运行配置文件  ├─dist //打包文件存储  │  └─static  │      ├─css  │      ├─fonts  │      ├─images  │      │  ├─account  │      │  ├─homepage  │      │  ├─introduction  │      │  ├─recruitContent  │      │  ├─session  │      │  ├─teams  │      │  ├─teamZone  │      │  └─welcome  │      ├─img  │      └─js  ├─src //源代码存储  │  ├─apis //前端所有请求接口封装文件  │  ├─assets //第三方资源存储  │  │  └─css  │  ├─components //全局公用组件  │  ├─router //页面路由配置文件  │  ├─styles //全局顶级权限样式文件  │  └─views //页面存储文件  │      ├─account //个人账户页面及其私有子组件  │      ├─history //历史日志页面及其子组件  │      ├─homepage //主页侧栏  │      ├─introduction //介绍入口页  │      ├─message //消息列表页面及其子组件  │      ├─recruitContent //招募详细新页面及其子组件  │      ├─recruitEdit //招募编辑页面及其子组件  │      ├─recruitList //招募列表及其子组件  │      ├─session //二期开发的会议室页面及其子组件  │      ├─settings //个人账号设置页面及其子组件  │      ├─teams //团队列表页面及其子组件  │      ├─teamZone //团队空间展示详细成员信息页面及其子组件  │      └─welcome //登录注册页面及其子组件  └─static //第三方静态资源存储      └─images //分页面存储图片素材          ├─account          ├─homepage          ├─introduction          ├─recruitContent          ├─session          ├─teams          ├─teamZone          └─welcome |

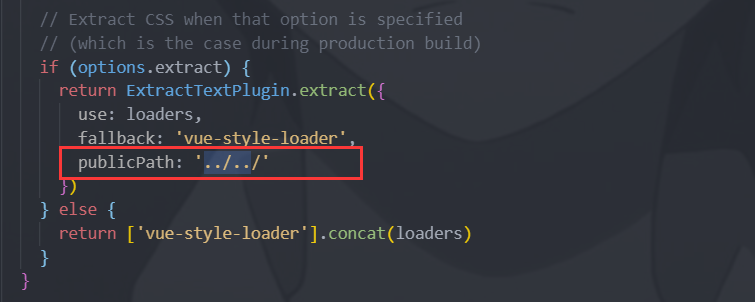
## 项目打包

### 修改资源路径

打开 config/index.js文件，然后在build环境下加入 assetsPublicPath:'./：

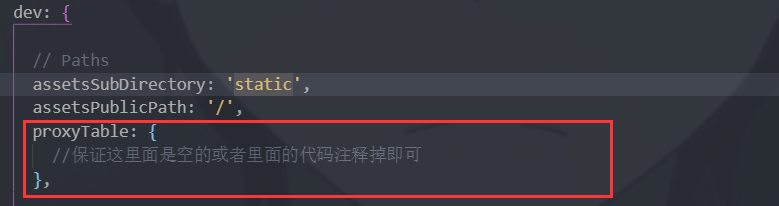


为了保证静态的资源（包括图片，elementui组件的icon图标等）正常加载显示，需要打开 build/utils.js文件，然后在 fallback下加入 publicPath:'../../'：



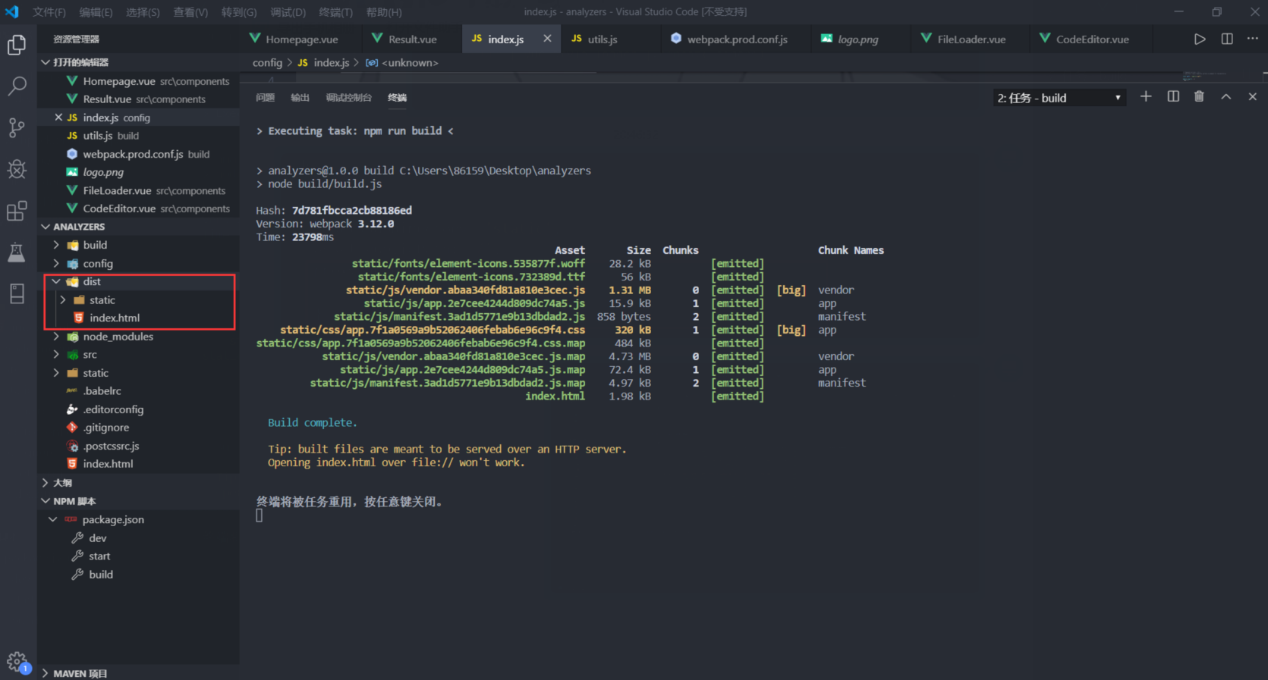
### 关闭跨域代理

在开发时，我们为了解决前后端跨域问题需要修改 proxyTable。但是部署到服务器后项目统一放置在一个文件件内，无需配置跨域代理。因此将 config/index.js文件的 proxyTable处的 '/api'后面的代码注释掉：



### 打包

新建终端，输入 npm run biuld打包项目，项目会生成一个index.html文件和一个用来存储静态依赖的static文件夹，统一放置在dist文件夹下：



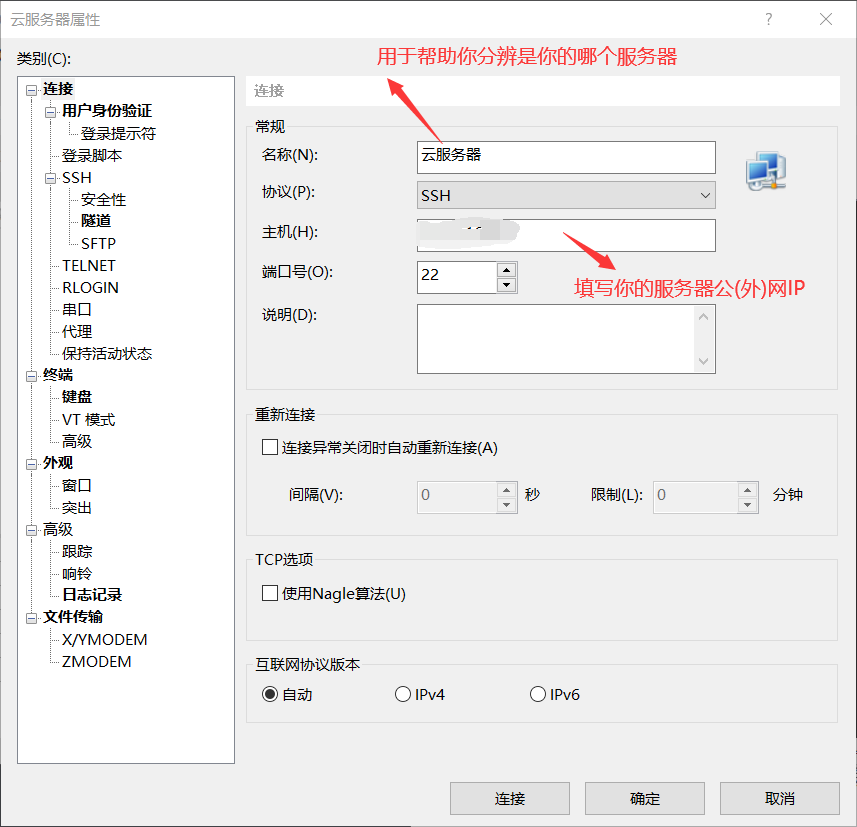
至此vue项目的打包就完成了。打开index.html查看能否正常显示页面，如果可以正常显示即说明打包成功。若使用了elementui组件库，还需要检查icon图标是否也能够正常显示。

## Vue项目部署以及Nginx转发

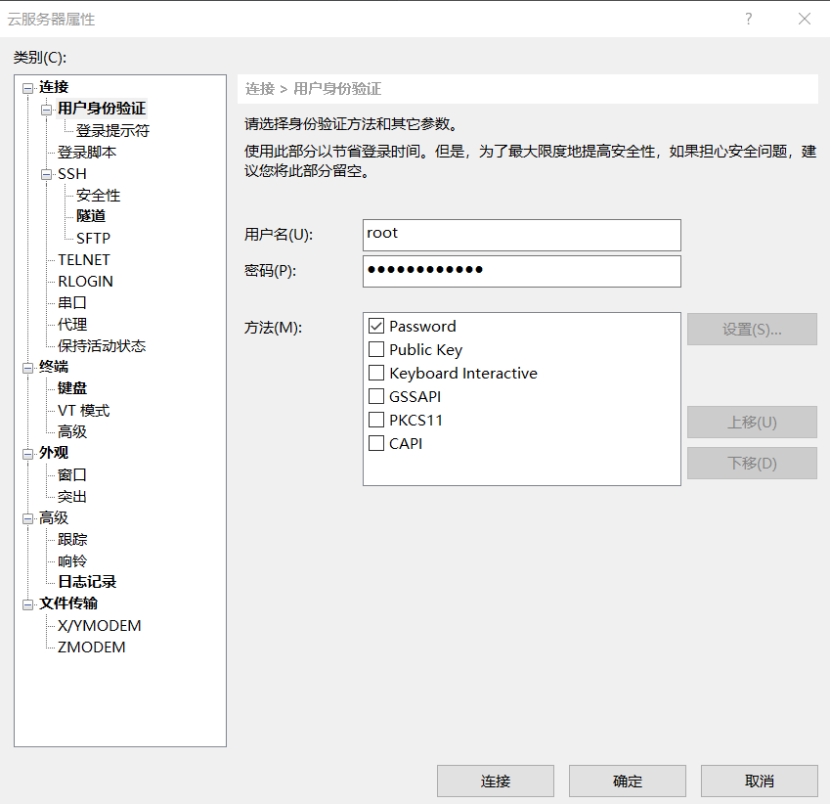
接下来通过xshell和xftp远程连接服务器，然后将打包好的dist文件夹放到特定的位置，修改nginx.conf文件的配置，最后重启服务即可。

### XShell配置

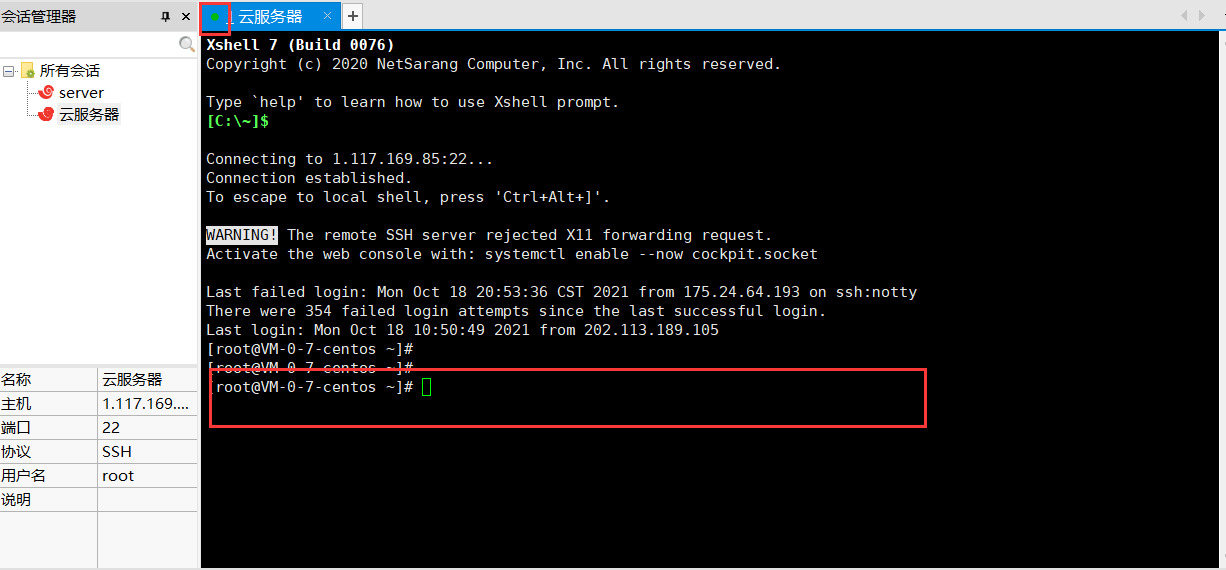
新建会话并填写要连接的服务器的相关信息：



点击用户身份验证，填写登录服务器的身份验证的账号和密码：

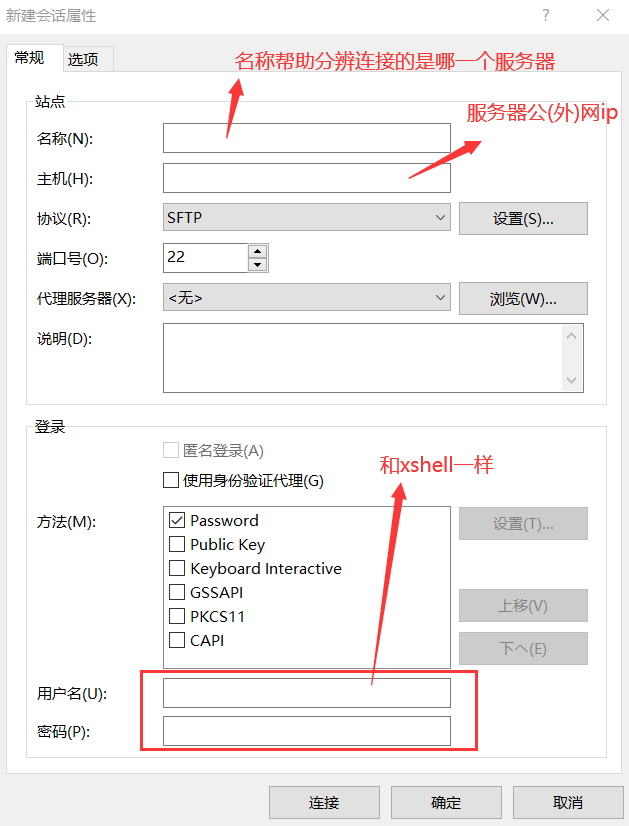


成功连接以后进行下一步操作：

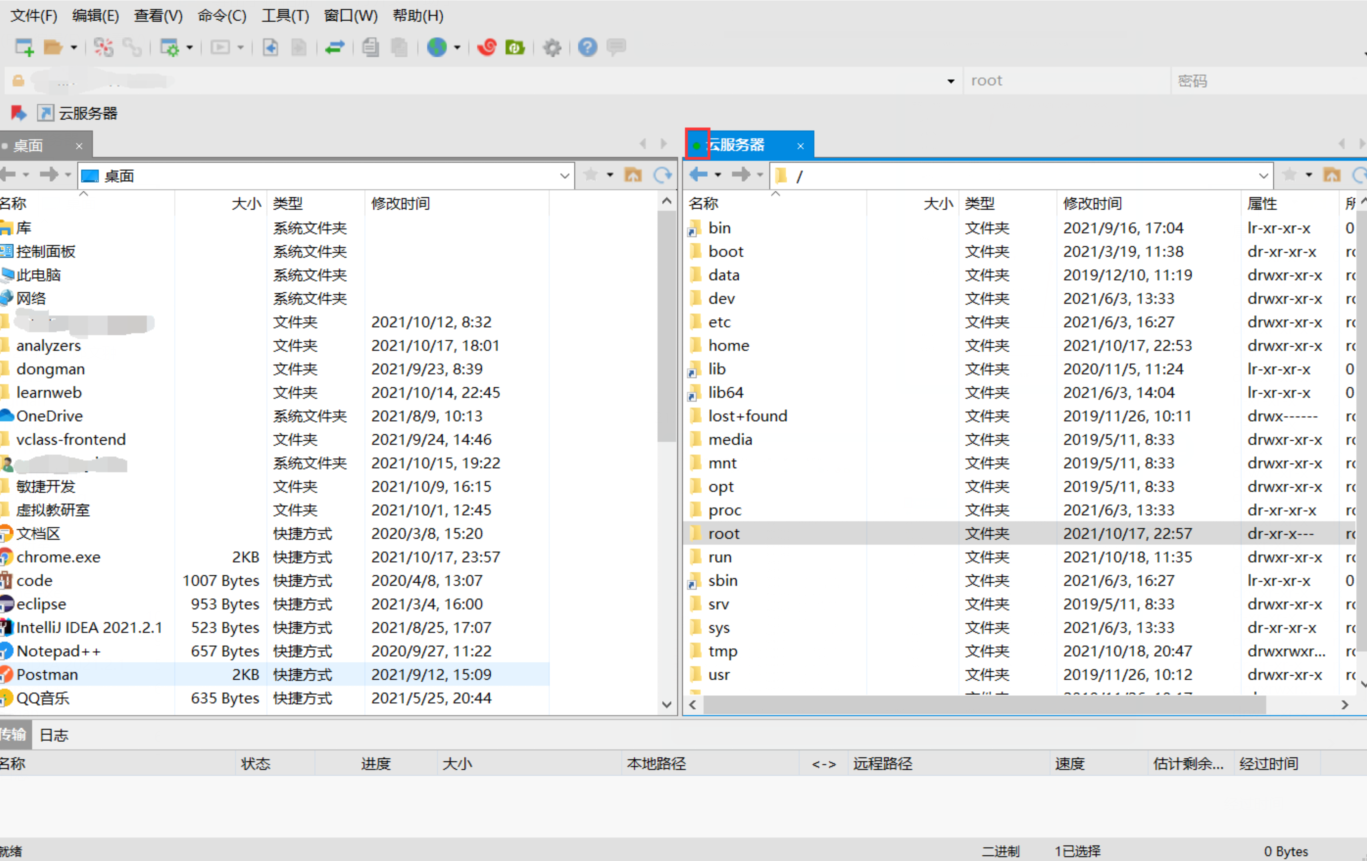


### Xftp连接远程服务器

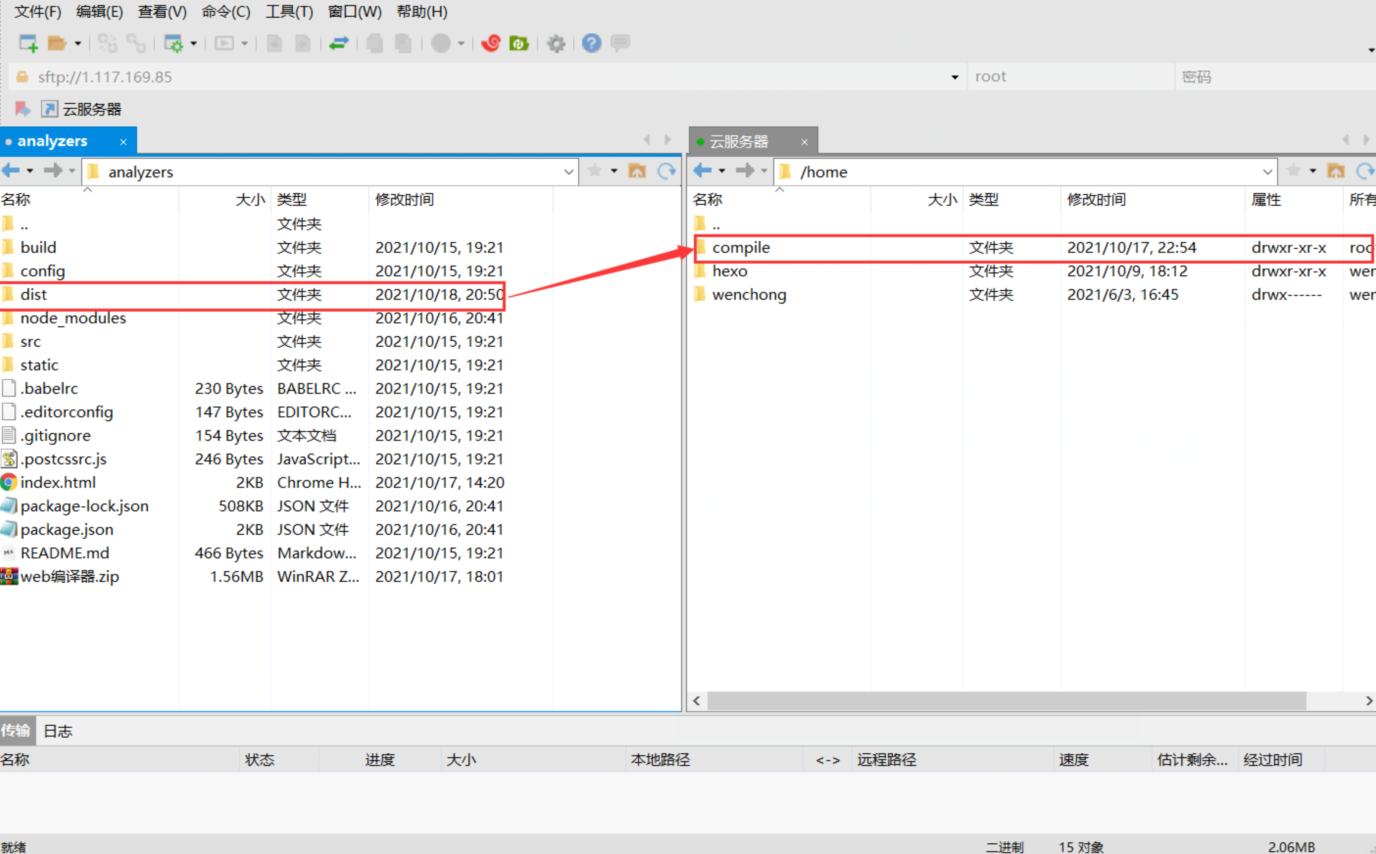
新建会话，填写相关的配置信息：



成功后如下所示，左侧是当前本地电脑的文件目录，右侧是远程服务器的文件目录：



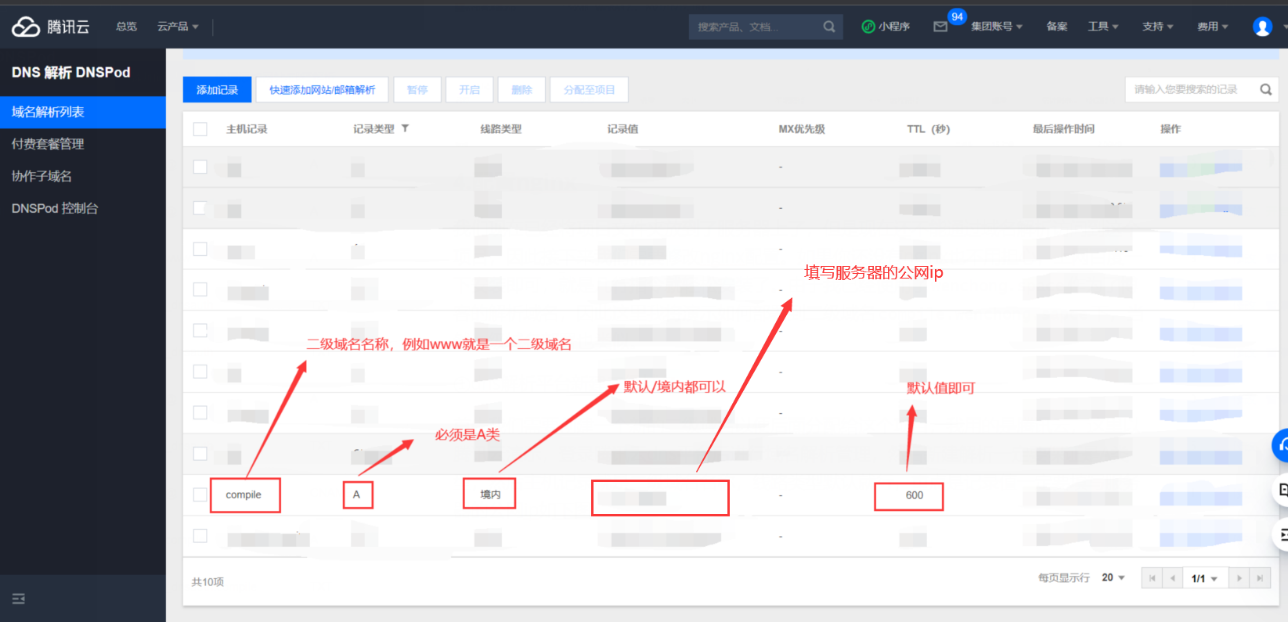
为了方便区分多个文件夹，将dist更改名称易于区别，例如这里的代码缺陷自动检测项目就是将dist文件夹更名为了compile文件夹。



### 配置nginx二级域名解析

现在已经将项目文件夹放到服务器上了，但是还不能通过域名解析访问项目，还需要修改nginx配置。由于我已经使用了 coolchong.cn作为博客的解析域名，因此这里我将演示如何部署到二级域名 vteam.coolchong.cn下，当然一级域名部署也类似。

在dns解析平台新建二级域名，首先需要创建一个新的二级域名以便后面分配给项目，这里以腾讯云演示。登录腾讯云dns平台后进行域名解析管理，然后新建解析一定要保证是A类型，然后主机记录就是二级域名名称，线路类型默认就行，重点是记录值一定要填写服务器的公网ip，如下图：



保存新添加的解析并且开启解析，然后配置 nginx.conf。在xftp找到/etc/nginx/nginx.conf文件，然后在最下方补充如下代码：

|  |
| --- |
| server {   gzip on;          gzip\_comp\_level 9;          gzip\_min\_length 1K;   gunzip on;          listen      443 ssl;          server\_name vteam.coolchong.cn;            ssl\_certificate /etc/ssl/1\_vteam.coolchong.cn\_bundle.crt;          ssl\_certificate\_key /etc/ssl/2\_vteam.coolchong.cn.key;          ssl\_session\_cache shared:SSL:1m;          ssl\_session\_timeout  10m;          ssl\_ciphers PROFILE=SYSTEM;          ssl\_prefer\_server\_ciphers on;          location / {             root    /home/vteam;    index index.html;          }       } |

保存退出，nginx配置完成。重启nginx服务，由于我们修改了nginx的配置，因此还要重启服务，打开xshell，进入nginx检查nginx配置是否语法正确：

|  |
| --- |
| cd /etc/nginx  nginx -t |

如果没有语法错误则会出现如下信息：

|  |
| --- |
| nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  nginx: configuration file /etc/nginx/nginx.conf test is successful |

输入指令重启nginx服务：

|  |
| --- |
| systemctl restart nginx.service |

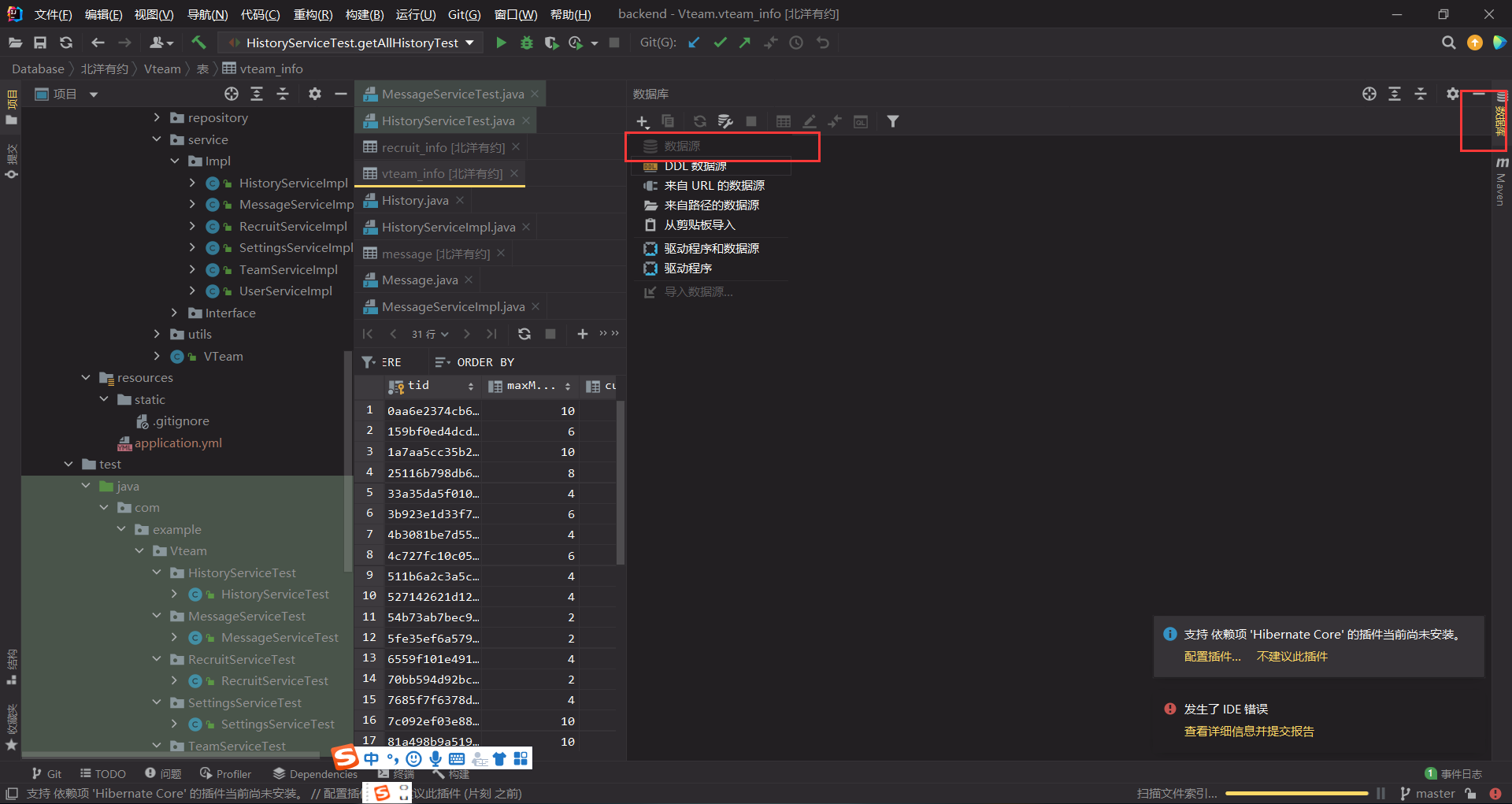
然后输入刚刚填写的二级域名 vteam.coolchong.cn，发现已经成功部署，可以正常解析。

# 后端部署

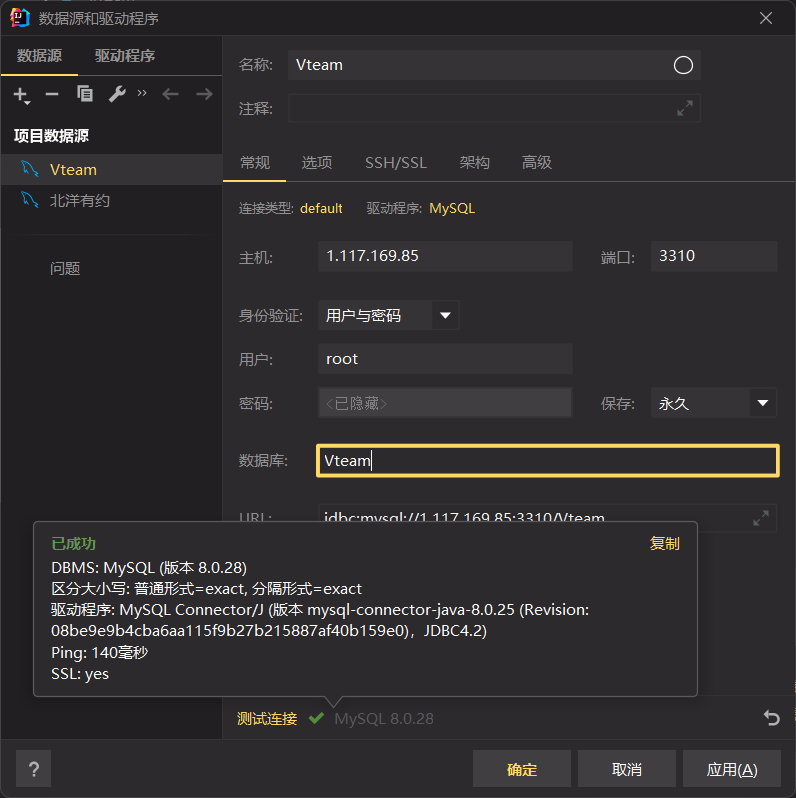
本项目使用的数据库并不运行在本地，而是在远程服务器的docker容器里运行。因此无需本地维护数据库，只需要通过填写账户以及远程服务器端口号即可进行连接。

## Idea连接远程服务器

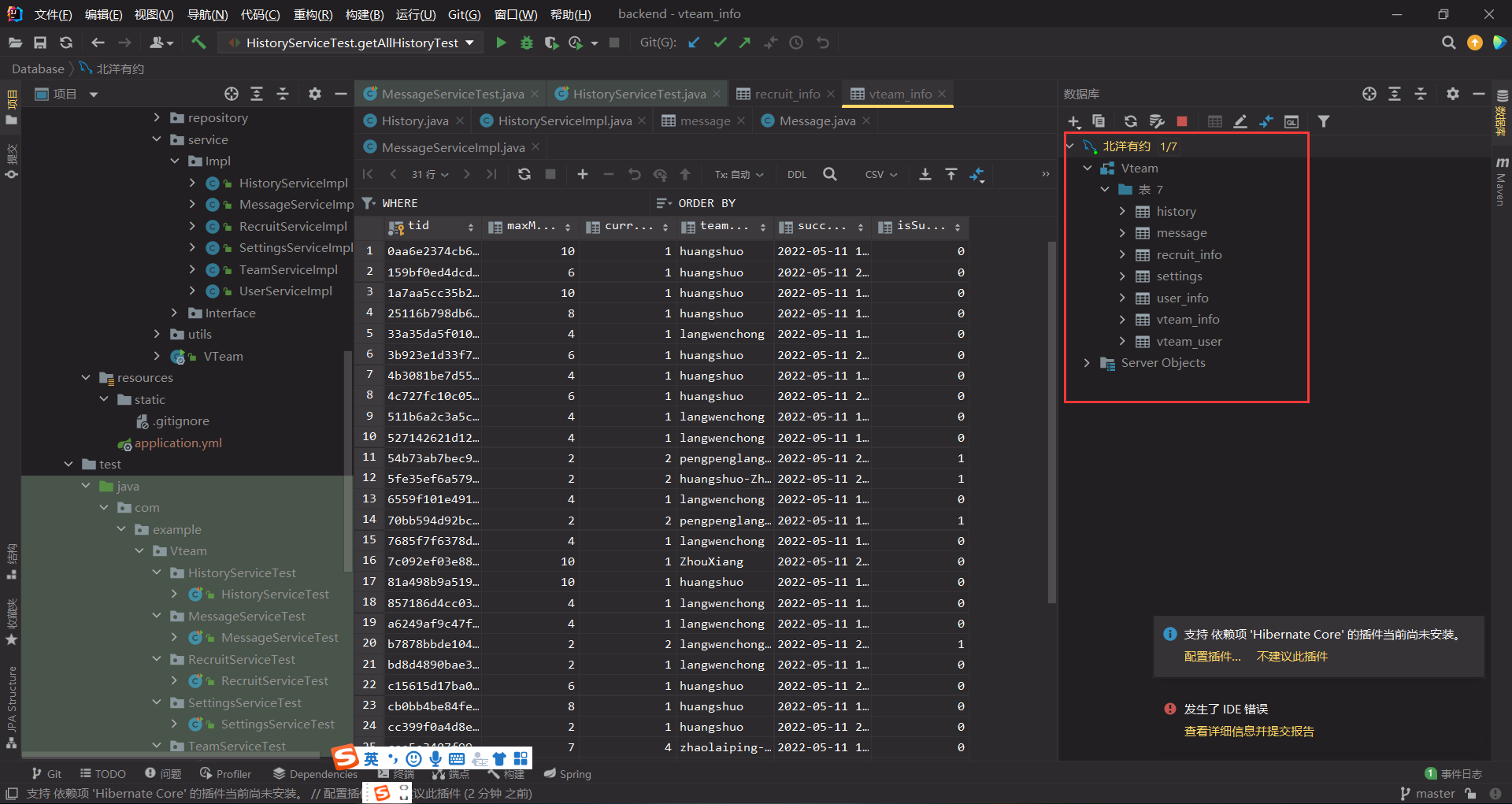
在本地通过idea连接远程服务器，打开idea（不能是付费版），点击右侧数据库选择连接数据源：



然后点击mysql数据源按照下图输入以下信息即可以连接到数据库：



连接成功后就可以在右侧查看到所有数据库表内数据：



## 项目文件结构

|  |
| --- |
| ├─.idea  │  └─dataSources  │      └─ac77b6a9-e7cc-4d63-b130-393f4a15218b  │          └─storage\_v2  │              └─\_src\_  │                  └─schema  ├─.mvn  │  └─wrapper  ├─src //代码存储文件  │  ├─data //建表sql语句文件  │  ├─main  │  │  ├─java  │  │  │  └─com  │  │  │      └─example  │  │  │          └─Vteam  │  │  │              ├─configuration //跨域配置文件  │  │  │              ├─controller //视图层，接收和返还数据  │  │  │              │  ├─Impl //具体实现  │  │  │              │  └─Interface //接口定义  │  │  │              ├─dao //数据交互层，和数据库交互  │  │  │              │  ├─Impl //具体实现  │  │  │              │  └─Interface //接口定义  │  │  │              ├─embed //联合主键类定义  │  │  │              ├─entity //实体对象类存储  │  │  │              ├─repository //jpa模板类定义  │  │  │              ├─service //业务层  │  │  │              │  ├─Impl //具体实现  │  │  │              │  └─Interface //接口定义  │  │  │              └─utils //自定义的常用工具函数  │  │  └─resources //静态资源存储点  │  │      └─static  │  └─test //代码测试文件  │      └─java  │          └─com  │              └─example  │                  └─Vteam  │                      ├─HistoryServiceTest //历史模块测试  │                      ├─MessageServiceTest //消息模块测试  │                      ├─RecruitServiceTest //招募模块测试  │                      ├─SettingsServiceTest //设置模块测试  │                      ├─TeamServiceTest //团队模块测试  │                      └─UserServiceTest //用户模块测试  └─target //打包文件存储      ├─classes      │  ├─com      │  │  └─example      │  │      └─Vteam      │  │          ├─configuration      │  │          ├─controller      │  │          │  ├─Impl      │  │          │  └─Interface      │  │          ├─dao      │  │          │  ├─Impl      │  │          │  └─Interface      │  │          ├─embed      │  │          ├─entity      │  │          ├─repository      │  │          ├─service      │  │          │  ├─Impl      │  │          │  └─Interface      │  │          └─utils      │  └─static      ├─generated-sources      │  └─annotations      ├─generated-test-sources      │  └─test-annotations      ├─maven-archiver      ├─maven-status      │  └─maven-compiler-plugin      │      ├─compile      │      │  └─default-compile      │      └─testCompile      │          └─default-testCompile      ├─surefire-reports      └─test-classes          └─com              └─example                  └─Vteam                      ├─HistoryServiceTest                      ├─MessageServiceTest                      ├─RecruitServiceTest                      ├─SettingsServiceTest                      ├─TeamServiceTest                      └─UserServiceTest |

## Application.yml文件配置

为了能够本地调试和服务端运行配置，需要在resources下新建一个application.yml文件来配置一些选项，具体的配置见下图：

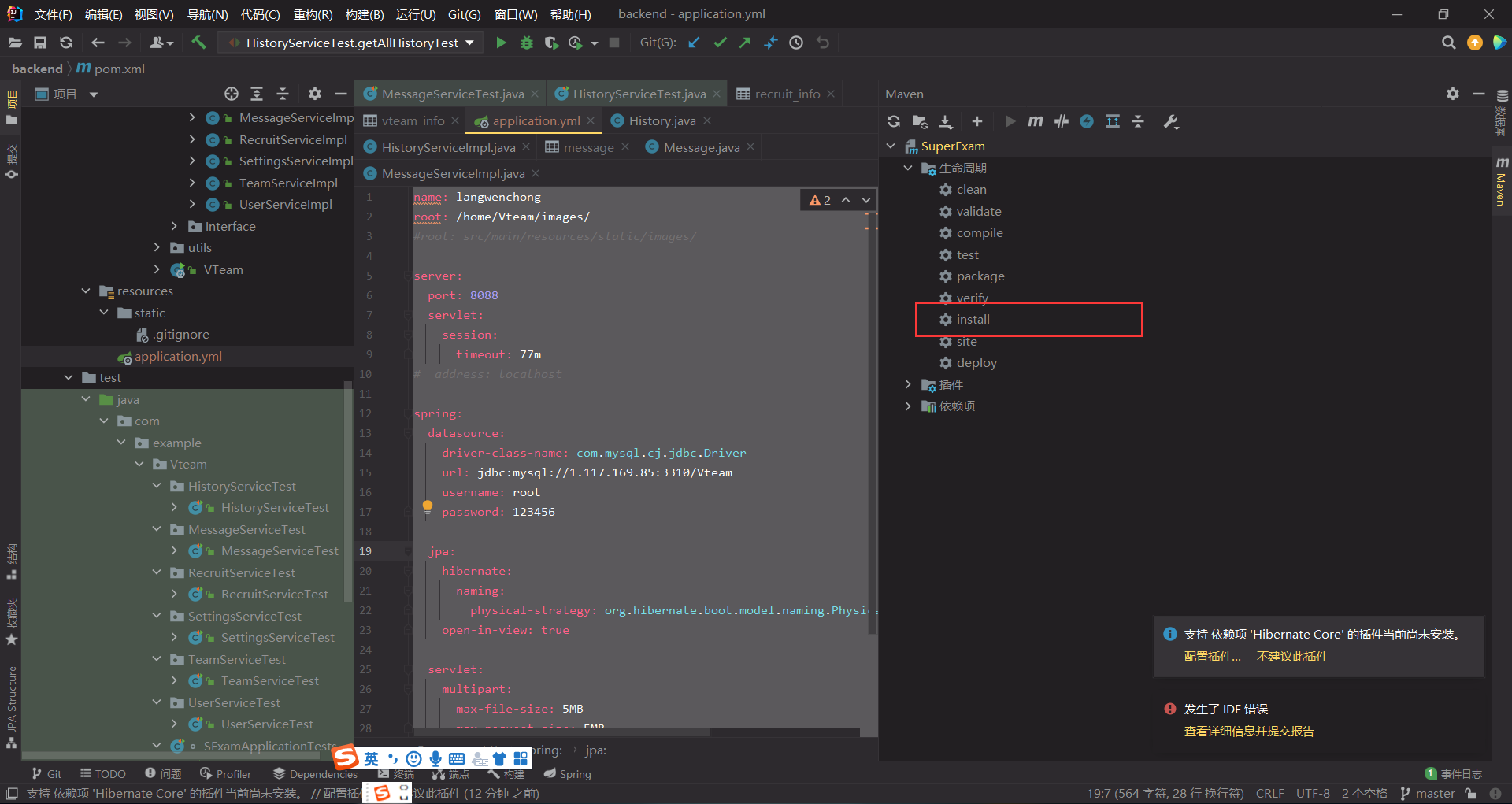
|  |
| --- |
| name: langwenchong  root: /home/Vteam/images/ #部署远程服务器时素材存放处  #root: src/main/resources/static/images/ #本地调试时素材存放处  server: #运行端口    port: 8088    servlet:      session:        timeout: 77m  # address: localhost #如果本地启动内网ip请填写ipconfig下对应的ip  spring: # 使用的是mysql驱动进行数据库连接    datasource:      driver-class-name: com.mysql.cj.jdbc.Driver      url: jdbc:mysql://1.117.169.85:3310/Vteam      username: root      password: 123456    jpa: # 使用jpa进行数据持久化存储      hibernate:        naming:          physical-strategy: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl      open-in-view: true    servlet: # 定义最大传输文件容量      multipart:        max-file-size: 5MB        max-request-size: 5MB |

配置完成并待索引加载完毕以后，可以直接点击绿色箭头进行本地的项目构建和调试了。

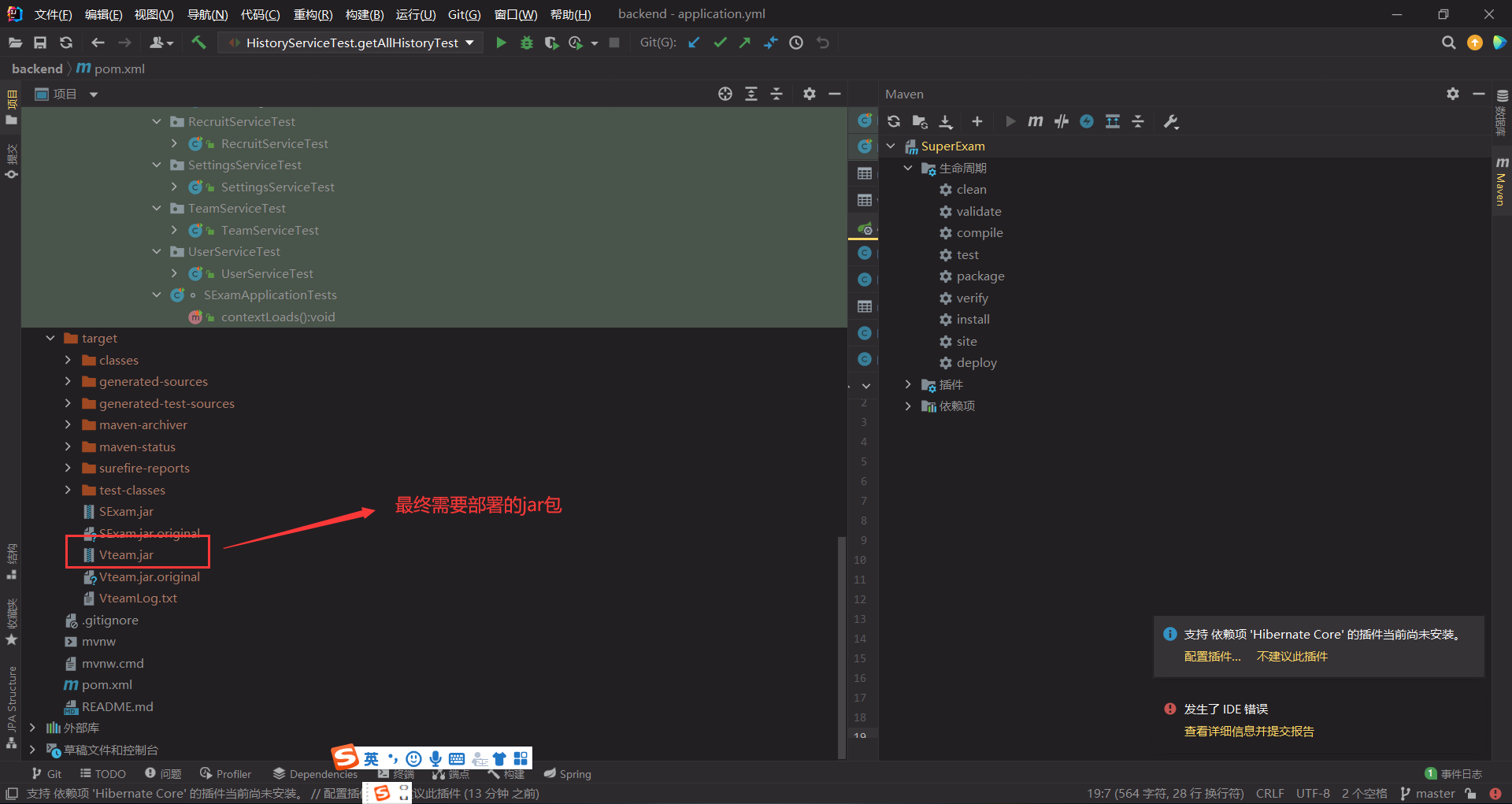
## 项目打包和部署

### 项目打包

点击右侧的maven，然后在生命周期中选择install：

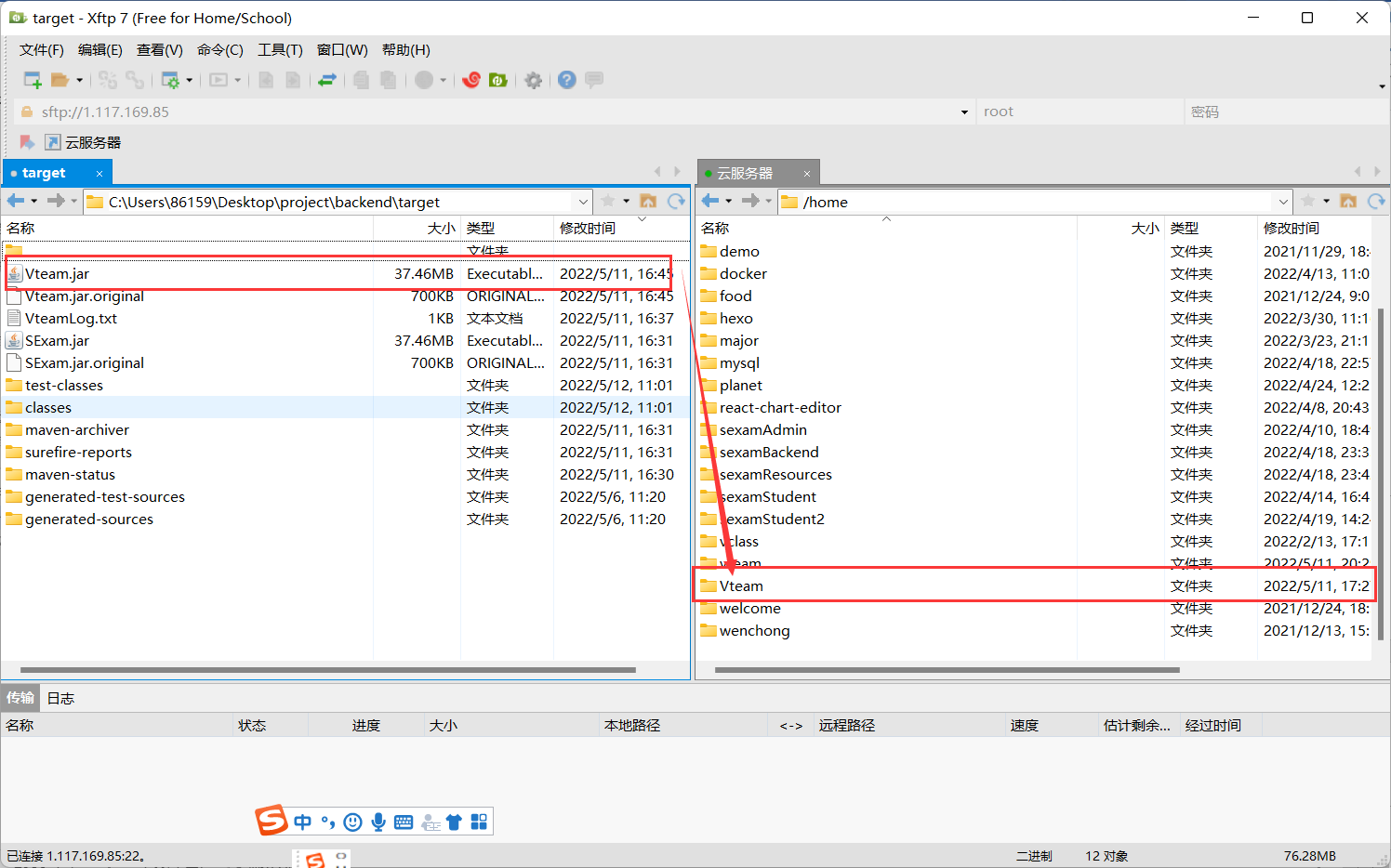


打包完成后即可以在终端中查看到Build Success字样同时target生成如下文件：



### Xftp部署项目并运行

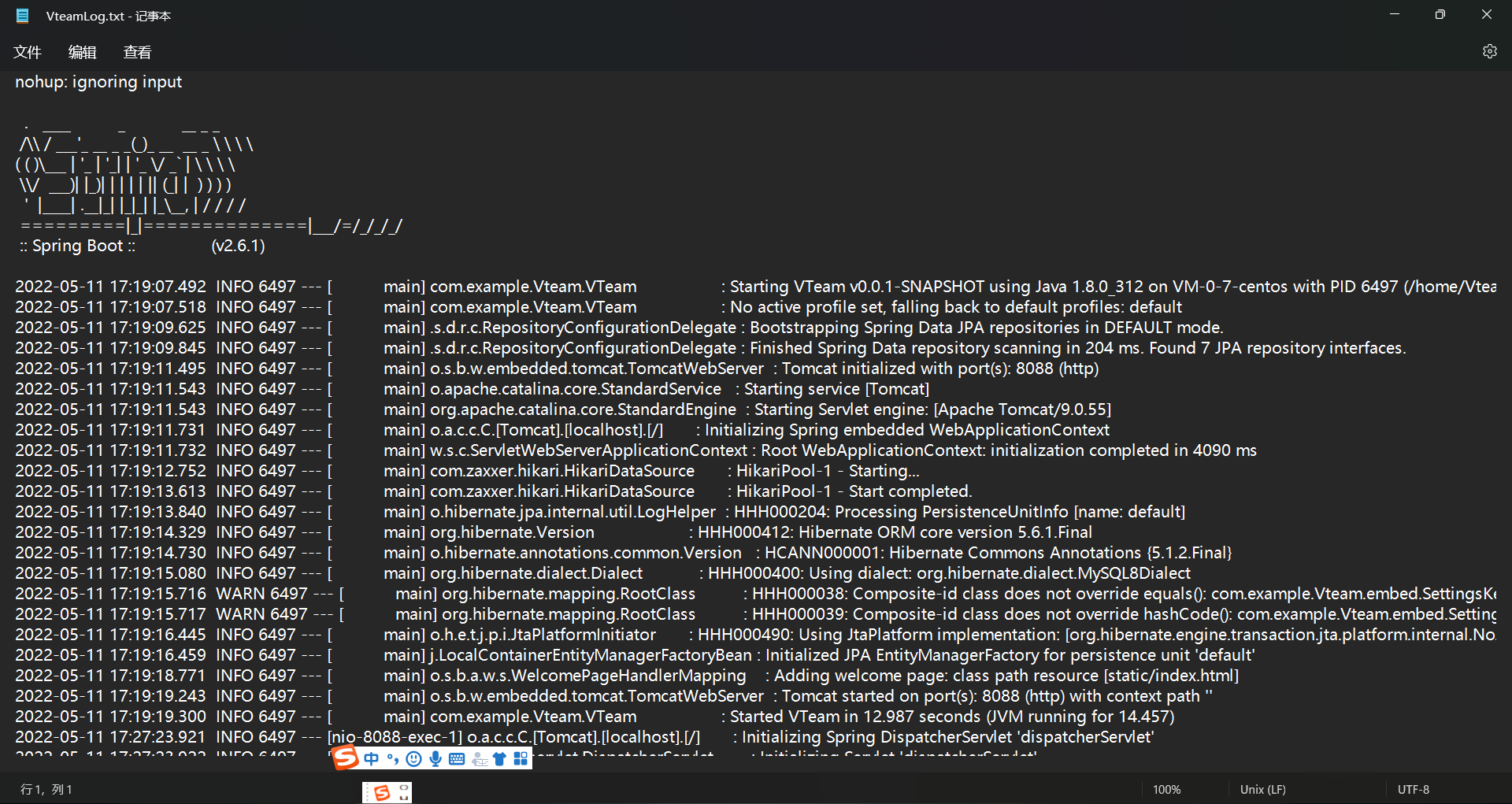
打开xftp将jar包上传到/home/Vteam路径下：



进入Vteam文件夹输入如下运行jar文件来启动后台服务：

|  |
| --- |
| nohup java -jar Vteam.jar >VteamLog.txt 2>&1 & |

如果正常启动可以在日志输出文件VteamLog.txt中看到启动时的输出：



### 配置接口域名绑定以及第三方资源路径代理

虽然开启了后台的服务，但是此时如果前台要请求接口，需要输入ip+value的路径形式进行请求。为了方便，我们为这个服务绑定一个二级域名即vteamapi.coolchong.cn。还是参照上面去申请https格式的二级域名，然后打开nginx.conf文件配置nginx：

|  |
| --- |
| server {          listen      443 ssl;          server\_name vteamapi.coolchong.cn;            ssl\_certificate /etc/ssl/1\_vteamapi.coolchong.cn\_bundle.crt;          ssl\_certificate\_key /etc/ssl/2\_vteamapi.coolchong.cn.key;          ssl\_session\_cache shared:SSL:1m;          ssl\_session\_timeout  10m;          ssl\_ciphers PROFILE=SYSTEM;          ssl\_prefer\_server\_ciphers on;          location / {              proxy\_pass http://localhost:8088;          }      } |

然后再配置一个vteamresources.coolchong.cn二级域名来代理指向存储图片的路径：

|  |
| --- |
| server {          listen      443 ssl;          server\_name vteamresources.coolchong.cn;             ssl\_certificate /etc/ssl/1\_vteamresources.coolchong.cn\_bundle.crt;          ssl\_certificate\_key /etc/ssl/2\_vteamresources.coolchong.cn.key;          ssl\_session\_cache shared:SSL:1m;          ssl\_session\_timeout  10m;          ssl\_ciphers PROFILE=SYSTEM;          ssl\_prefer\_server\_ciphers on;          location / {            root    /home/Vteam/images;          }       } |

配置基本完成，接下来输入如下指令重启nginx：

|  |
| --- |
| Systemctl restart nginx.service |

至此前后台项目的打包部署就完成了，可以通过https://vteam.coolchong.cn访问项目界面，然后界面会通过https://vteamapi.coolchong.cn调用后台服务，最后会使用https://vteamresources.coolchong.cn+图片相对路径，来获取存储在后台的静态资源。