**问卷星球**

[部署文档]

|  |  |
| --- | --- |
| **成员** | **张锴睿** |
| **成员** | **高嘉轩** |
| **成员** | **魏润田** |
| **成员** | **胡斌** |
| **成员** | **刘兆丰** |

## **1. 引言**

### **1.1项目背景**

随着互联网的发展，我们对高效的数据收集处理的需求正在不断增加。作为一名大学生，日常的调研、志愿、科研活动中，我们常常需要信息收集和处理，来为我们提供数据和分析结果；软件开发时需要对用户的需求和市场情况进行调查，来为软件开发提供灵感和方向以及制定相关的营销策略。因此，拥有一个高效的数据收集处理软件是时代发展的必然趋势。应对于学习工作中数据收集处理的需求，我们决定开发一个问卷设计与发布平台，高效地解决用户收集数据难，处理数据慢的问题。

### **1.2编写目的**

本文档是问卷星球的用户使用说明书，本文档的主要目的旨在对软件的各个专有名词、模块、系统、页面等进行详细介绍，让用户了解软件的各项细节，辅助用户使用软件。

### **1.3涉及名词**

[1] 管理员：本系统的后台管理者，可以对问卷信息以及相应的答卷信息进行管理，确保内容正确合法。

[2] 游客：没有注册的网站用户，只能进行问卷填写的操作，以ip地址唯一确定身份并可以暂存相关数据，注册后可成为已注册用户。

[3] 已注册用户：已注册的网站用户，具有游客除了注册之外的所有功能，还可以进行问卷创建、问卷管理、数据分析、管理个人信息等操作。

### **1.4项目使用者**

本平台的实际使用者为需要进行调研、志愿、科研活动的在校大学生用户以及需要进行信息收集和调研各种公司企业用户。

系统维护人员为项目开发团队，对于此系统的相关部分比较熟悉，团队内部具有对数据库、计算机、网络较为熟悉的人员。

管理员为开发团队指定的人选，需要有一定的问卷信息和答卷信息的审核经验。

### **1.5参考资料**

[1]吕云翔，王昕鹏，邱玉龙. 软件工程——理论与实践. 北京：人民邮电出版社，2012.

[2]吕云翔，刘浩，王昕鹏，周建. 软件工程课程设计. 北京：机械工业出版社，2009.

[3]吕云翔编著. 软件工程实用教程. 北京：清华大学出版社，2015.

### **1.6相关文档**

[1] 《需求规格说明书》

[2] 《软件设计说明书》

[3] 《软件开发计划书》

[4] 《测试报告》

[5] 《用户使用说明书》

## 系统设置

### 2.1硬件运行环境

(1)服务器

处理器型号:阿里云轻量应用服务器

内存容量:1核2GB

网络配置:1Mbps

(2) Web浏览PC机

处理器型号: AMD或Imtel 1.6 GHz以上

内存要求:256MB以上

网络配置:100M带宽网络适配器

### 2.2软件运行环境

(1)服务器

操作系统: Ubuntu 20.04或以上版本系统

数据库: MySQL 8.0以上

(2)客户机

操作系统:无限制

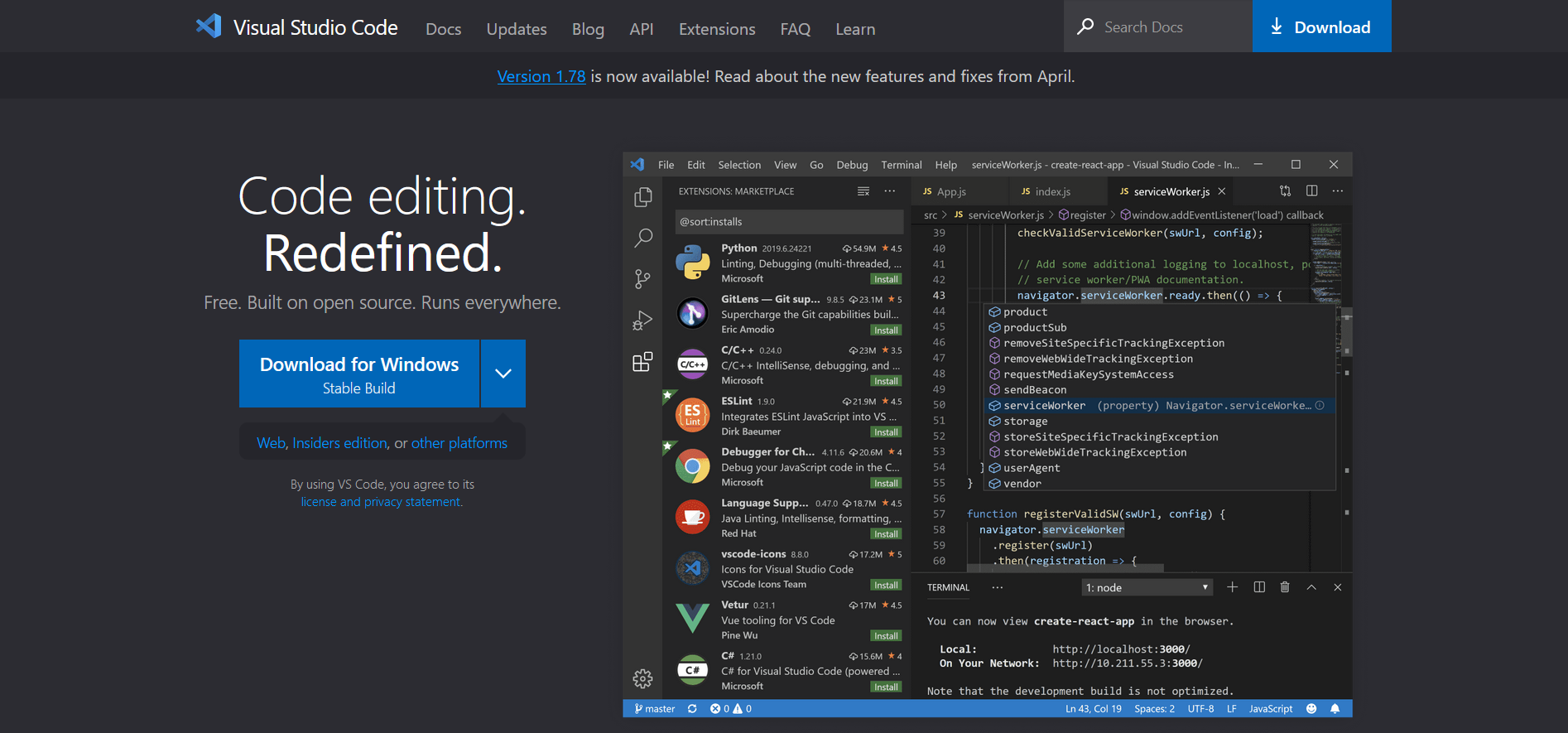
浏览器:支持所有主流浏览器，建议使用Chrome或Edge

网络环境:客户机应能够访问互联网

## 环境搭建和项目运行

### 3.1前端开发环境搭建

进入Visual Studio Code官网：https://code.visualstudio.com/，根据自己的系统进行相应的下载。官网页面如图1所示。



**图1 Visual Studio Code**

### 3.2前端vue环境搭建

#### 3.2.1下载Node.js

通过链接https://github.com/coreybutler/nvm-windows/releases，下载nvm-setup.zip并进行解压安装。

安装完成之后修改相应的配置，将 npm 包的下载源地址更改为淘宝镜像。打开 nvm 安装文件夹，在文件夹下的 settings.txt 文件中添加如下语句

node\_mirror: http://npm.taobao.org/mirrors/node/npm\_mirror: https://npm.taobao.org/mirrors/npm/

打开命令行输入nvm install v16.14.0与nvm use v16.14.0完成node.js的安装。

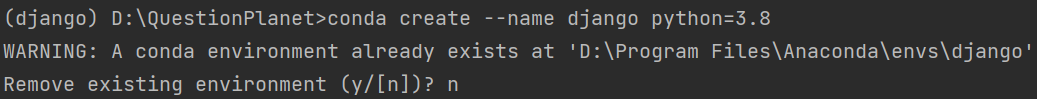
#### 3.2.2安装vue环境

在命令行中输入npm install vue -gnpm install vue-router -gnpm install -g @vue/cli 来将vue.js、vue-router、vue-cli 脚手架到 global 全局目录，之后利用语句vue create my-project 完成vue项目的创建。

### 3.3后端开发环境搭建

#### 3.3.1后端本地依赖环境搭建

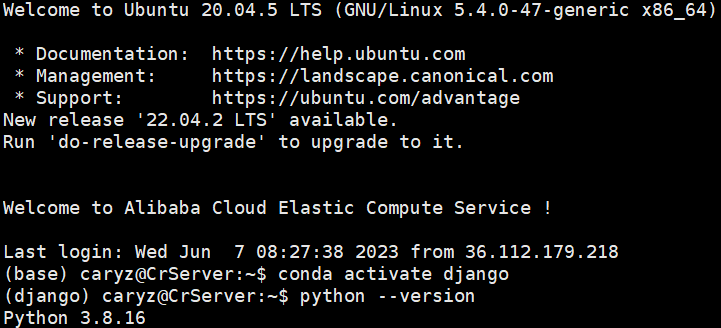
Django框架是基于python环境的后端框架，所以我们需要先进行python环境的安装，使用anaconda配置Django的虚拟环境，在初始化时设置python版本。创建虚拟环境页面如图2所示。



**图2 anaconda创建django虚拟环境**

#### 3.3.2服务器依赖环境搭建

我们使用的是阿里云的轻量级应用服务器，使用的操作系统为Ubuntu20.04.5，使用miniconda3配置了Django环境所需的python版本，查看python版本如图3所示。



**图3 服务器查看python版本**

### 3.4后端Django环境的搭建

#### 3.4.1本地Django安装

在安装完python并将它添加到系统环境变量之后，我们需要安装Django框架。进入Django虚拟环境后在终端输入：pip install django。即可进行Django的安装。其安装过程如图4所示。

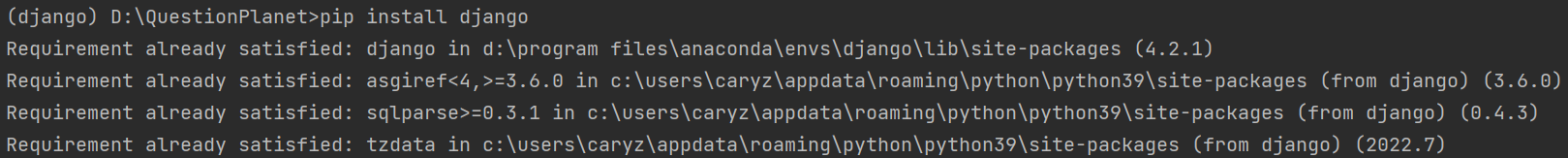


图4 本地安装Django过程

安装完毕之后，在终端中输入 pip list 查看 Django 是否安装成功和安装的版本号。本地查看Django安装结果图5所示。

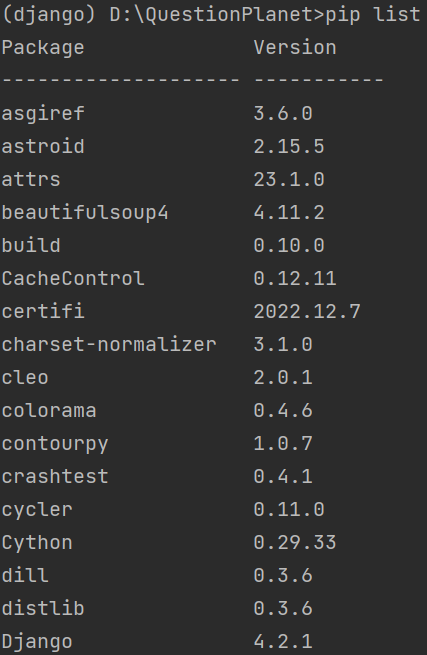


图5 本地查看Django安装结果

#### 3.4.2服务器Django安装

在查看完服务器的 python 版本号之后，进入 Django ，使用 pip install django 下载安装 Django，其安装过程如图6所示。

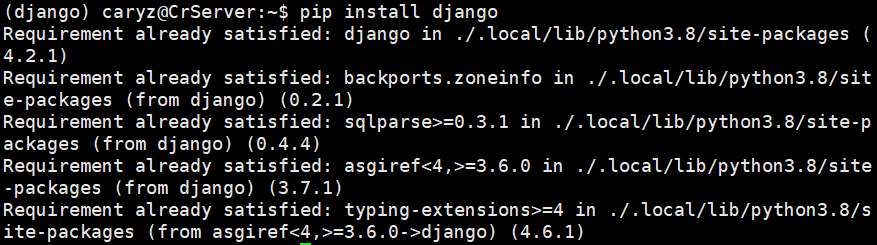


图6 服务器安装 Django 过程

安装完毕之后，可以输入 pip list 查看 Django 是否安装成功和安装的版本号。其结果如图7所示。

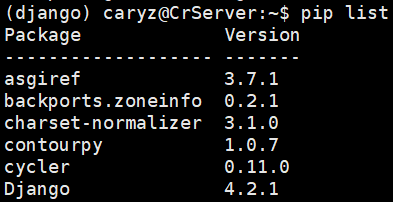


图7 服务器查看 Django 版本结果

## 项目部署

### 4.1后端部署

#### 4.1.1连接云服务器

使用Xshell连接阿里云服务器，点击新建连接后使用账号密码登录服务器，连接成功进入服务器界面如图8所示。

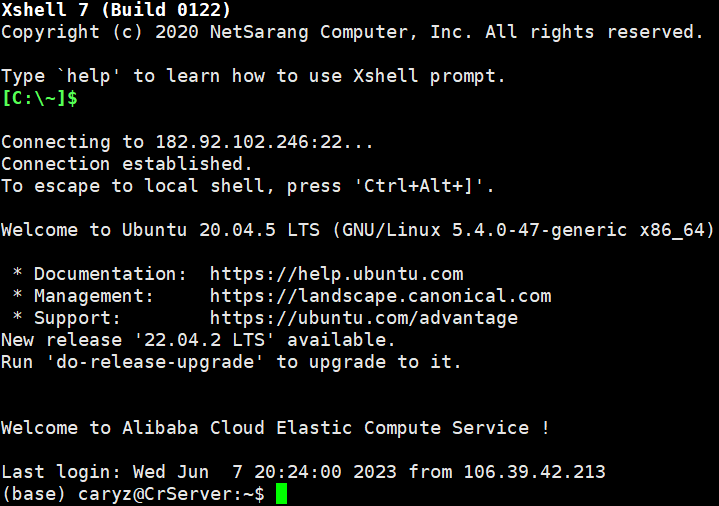


图8 Xshell连接服务器界面

#### 4.1.2下载项目

在服务器上创建好虚拟环境后，创建目录作为项目根目录，进入目录后使用 git clone 获取项目文件夹，后续更新使用 git pull 更新文件。更新项目文件如图9所示。

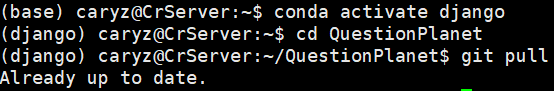


图9 更新项目

#### 4.1.3配置uswgi启动Django项目

在服务器上打开至刚才下载好的项目的目录，配置uwsgi.ini文件，具体内容需要根据当前项目存放位置来编写。配置成功后的uwsgi.ini启动文件内容如下编写uwsgi.ini启动文件如图10所示。

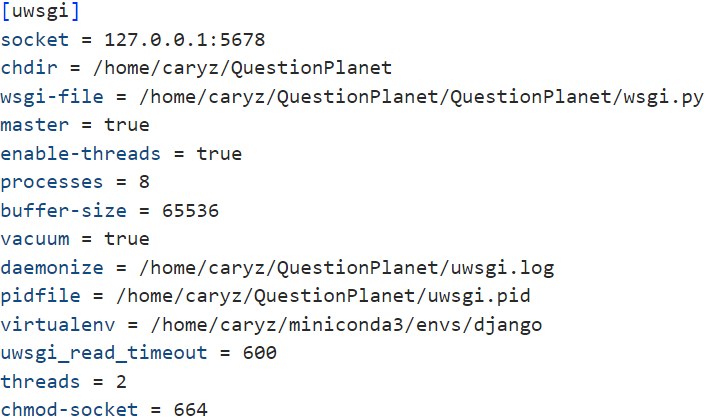


图10 编写好的 uwsgi.ini 配置文件

然后在uwsgi.ini启动文件的目录下输入uwsgi --ini uwsgi.ini 进行uwsgi启动。启动结果如图11所示。

descript

图11 启动 uwsgi

此时可以通过指令 ps aux|grep 'uwsgi' 查看是否存在已经启动的uwsgi进程。结果如图12所示。

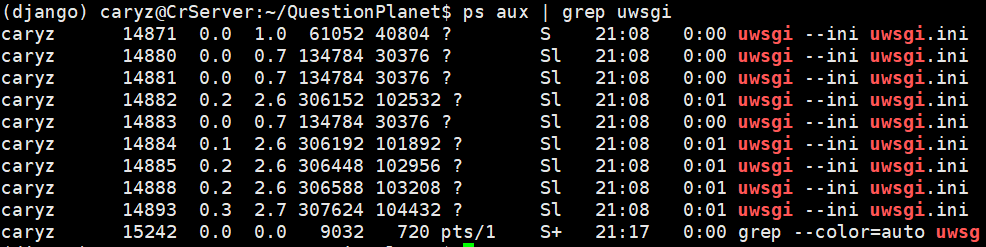


图12 查看 uwsgi 是否启动成功

#### 4.1.4配置Nginx完成部署

首先在服务器上安装nginx，输入sudo apt-get install nginx进行安装。结果如下图所示，由于该服务器之前已经安装过nginx，所以下方显示的是已安装状态。下载安装nginx如图13所示。

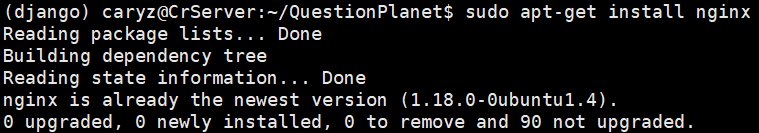


图13 下载安装nginx

然后通过nginx -V查询下载安装的nginx安装是否成功以及版本号。查看版本号如图14所示。

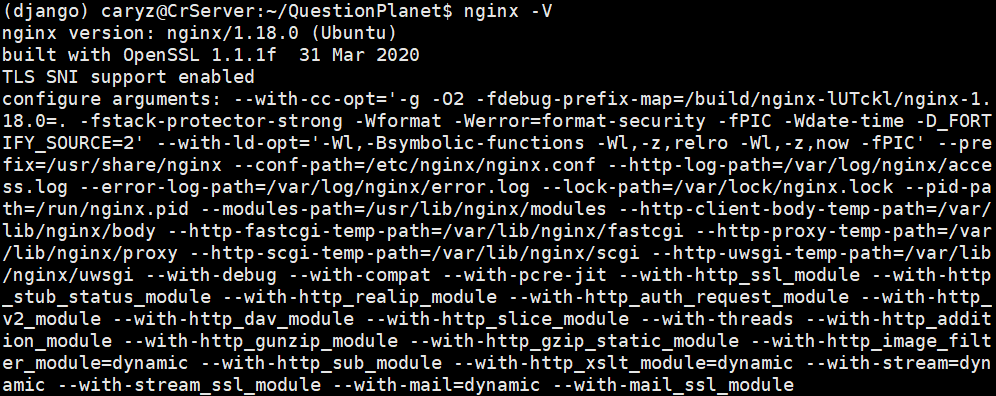


图14 查看nginx版本号

然后进入/etc/nginx/conf.d目录下配置相关的文件，定义项目的工作目录以及设置跨域访问的各种限制以及uwsgi的端口。写好的配置文件如图15所示。



图15 nginx配置文件

使用 sudo nginx -t 查看配置文件编写是否有误，最后输入 sudo systemctl start nginx 启动 nginx 服务。查看过程如图16所示。

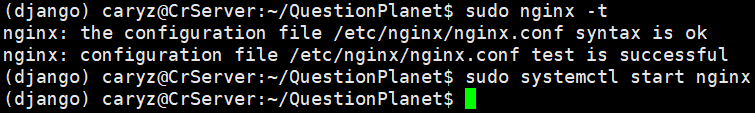


图16 查看nginx配置是否正确并启动

这时服务器端的配置基本已经完成。本项目还需要连接一个远程MySQL数据库，数据库的配置文件已经写入到了settings.py文件中，不需要手动修改。

至此，后端已成功部署到服务器。

### 4.2前端vue项目打包上传

此部分在4.3中实现。

### 4.3使用github actions工作流自动部署

在仓库的dev分支中创建.github/workflows/deploy.yml文件，使用工作流实现检查代码、通过ssh访问服务器、拉取项目文件、下载依赖、重新部署、打包前端项目、复制到指定目录、重启代理服务器等操作，实现自动部署。deploy.yml文件如图17所示。

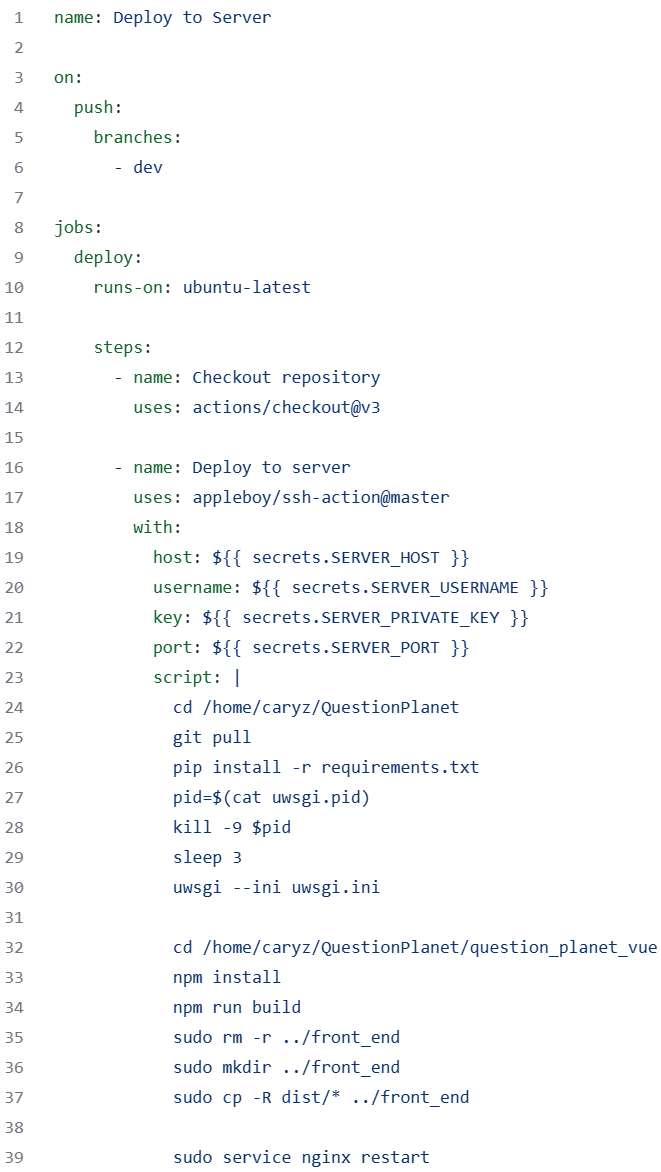


图17 使用deploy.yml实现自动部署

当 push 到 dev 分支时，会在 github actions 上自动执行指令，实现自动部署。部署成功结果如图18所示。

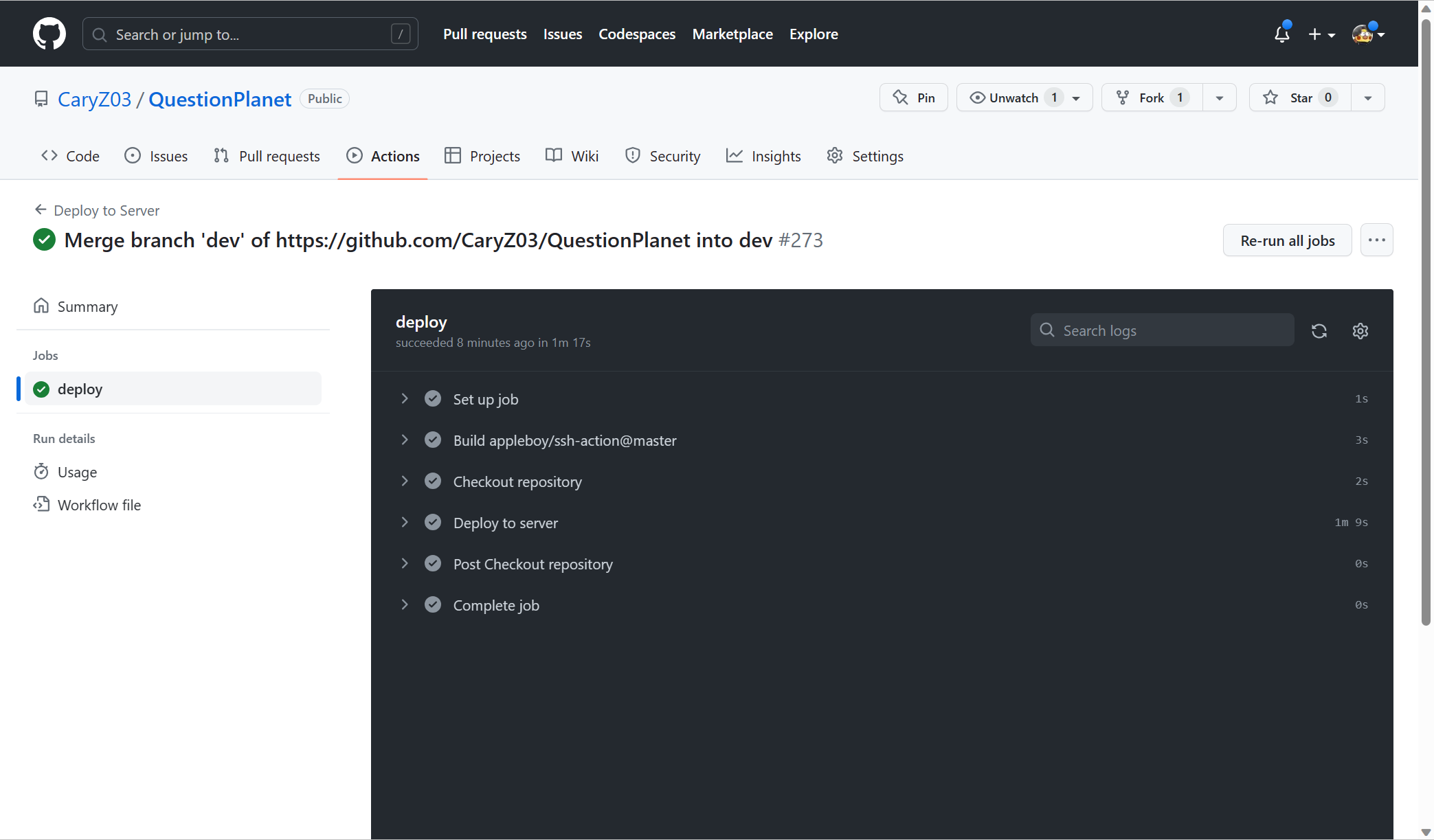


图18 部署成功结果