# 1. 部署后端

购买华为云主机，选择的 Linux 版本为CentOS 7.6，规格为2vCPUs | 4GB。

## 1.1 MySQL

### （1）安装

这里我将选择安装 MySQL 社区版：

[root@CentOS ~]*# wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm*

[root@CentOS ~]*# sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm*

[root@CentOS ~]*# yum install -y mysql-server*

[root@CentOS ~]*# systemctl start mysqld*

[root@CentOS ~]*# systemctl enable mysqld*

### （2）创建 Flask 应用需要的数据库

使用 root 用户登录 MySQL：

[root@CentOS ~]*# mysql -u root -p*

Enter password:

Welcome to the MySQL monitor. Commands end with ; or **\g**.

Your MySQL connection id is 15

Server version: 5.6.47 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' **for** help. Type '\c' to clear the current input statement.

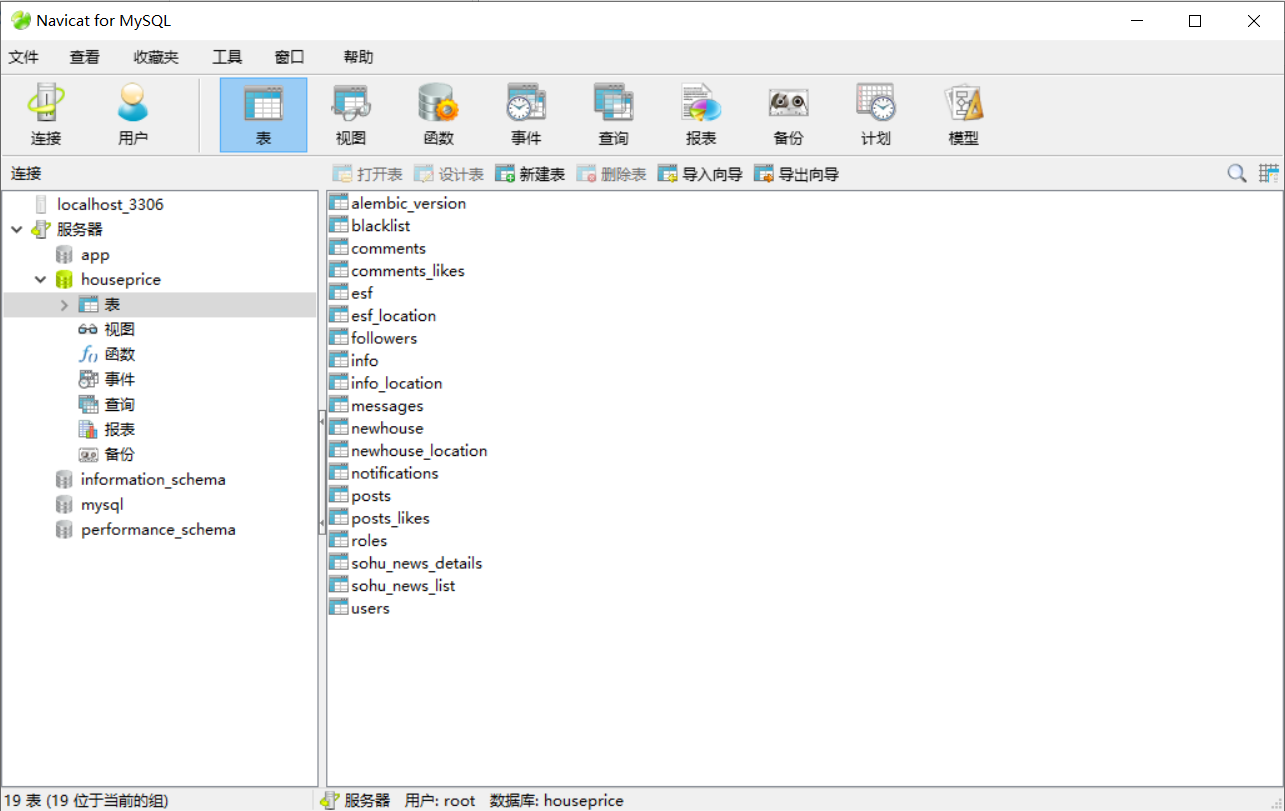
mysql>

创建名为 houseprice 的数据库：

mysql> create database houseprice character set utf8 collate utf8\_general\_ci;

Query OK, 1 row affected (0.00 sec)

用Navicat for MySQL连接云服务器的数据库，在本地将数据的.sql文件导入到 houseprice 数据库中：



## 1.2 Python 3

1. 准备编译环境

[root@CentOS ~]*# yum -y install gcc make readline-devel sqlite-devel openssl openssl-devel libffi-devel zlib\**

2. 编译安装

[root@CentOS ~]*# wget -P /root http://python.org/ftp/python/3.6.4/Python-3.6.4.tar.xz*

[root@CentOS ~]*# tar xf Python-3.6.4.tar.xz*

[root@CentOS ~]*# cd Python-3.6.4/*

[root@CentOS ~]*# ./configure --prefix=/usr/local/python-3.6*

[root@CentOS ~]*# make && make install*

[root@CentOS ~]*# ln -s /usr/local/python-3.6/bin/python3.6 /usr/bin/python3*

[root@CentOS ~]*# ln -s /usr/local/python-3.6/bin/pip3.6 /usr/bin/pip3*

更改 pip 安装源为国内的源（比如 aliyun）：

[root@CentOS ~]*# mkdir ~/.pip*

[root@CentOS ~]*# vi ~/.pip/pip.conf*

添加内容如下：

[global]

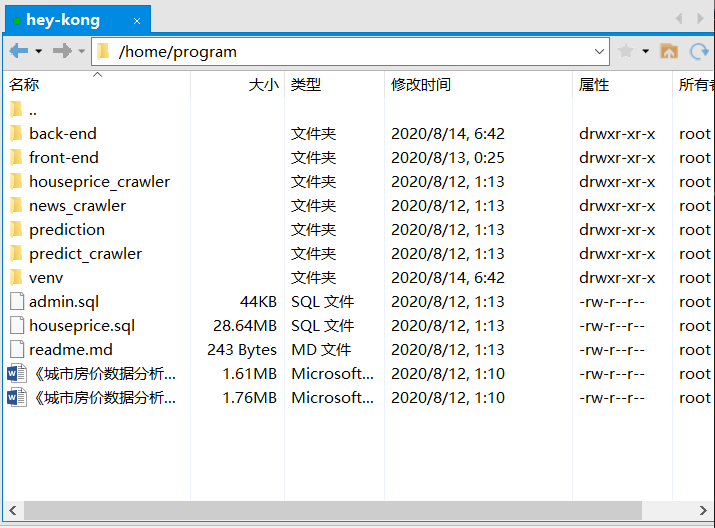
index-url = http://mirrors.aliyun.com/pypi/simple/

[install]

trusted-host=mirrors.aliyun.com

## 1.3 尝试启动 Flask 应用

你可以使用 Xftp 将代码整个目录上传到服务器上，放在 /home/program 位置：



执行如下命令：

[root@CentOS ~]*# cd /home*

[root@CentOS home]*# python -m venv venv*

[root@CentOS home]*# source venv/bin/activate*

(venv3) [root@CentOS home]*# cd program/back-end/*

(venv3) [root@CentOS back-end]*# pip install -r requirements.txt*

(venv3) [root@CentOS back-end]*# pip install pymysql*

(venv3) [root@CentOS back-end]*# flask db upgrade*

(venv3) [root@CentOS back-end]*# flask deploy*

(venv3) [root@CentOS back-end]*# flask run -h 0.0.0.0 -p 5000*

然后在笔记本中，打开浏览器，访问 http://121.36.253.244:5000/api/ping，如果返回 "Pong!" 则说明正常，先停止应用

**1.4 Gunicorn**

此时，可以切换到 Gunicorn 来启动 Flask 应用了：

(venv3) [root@CentOS back-end]*# pip install gunicorn*

使用 gunicorn 命令行来启动 Flask 非常简单：

(venv3) [root@CentOS back-end]*# gunicorn -w 3 main:app -b 0.0.0.0:5000*

说明：

* -w 3： 表示将启动 3 个 Gunicorn 进程，建议是 CPU 核数 \* 2 + 1
* main:app： 表示将要启动的应用为，当前目录下的 main.py 模块中的 app 对象

先关闭应用，接下来通过指定 gunicorn 配置文件来启动 Flask：

(venv3) [root@CentOS back-end]*# mkdir deploy*

(venv3) [root@CentOS back-end]*# vim deploy/gunicorn.conf.py*

内容如下:

import multiprocessing

bind = '0.0.0.0:5000'

workers = multiprocessing.cpu\_count() \* 2 + 1

*# daemon = True*

pidfile = '/run/gunicorn.pid'

loglevel = 'info'

errorlog = '/tmp/gunicorn-error.log'

accesslog = '/tmp/gunicorn-access.log'

access\_log\_format = '%(h)s %(l)s %(u)s %(t)s "%(r)s" %(s)s %(b)s "%(f)s" "%(a)s"'

更详细的配置可以参考 [Gunicorn](https://github.com/benoitc/gunicorn/blob/master/examples/example_config.py) 官方示例

**1.5 Supervisor**

现在想把用 Gunicorn 启动 Flask 应用的命令，放到 CentOS 后台执行，可以使用 [Supervisor](http://www.supervisord.org/)

[root@CentOS ~]*# yum install -y supervisor*

[root@CentOS ~]*# systemctl start supervisord.service*

[root@CentOS ~]*# systemctl enable supervisord.service*

然后再提供相关配置文件即可，请参考： <http://docs.gunicorn.org/en/stable/deploy.html#supervisor>

[root@CentOS ~]*# vim /etc/supervisord.d/gunicorn.ini*

内容如下:

[program:gunicorn]

command=/home/venv/bin/gunicorn main:app -c /home/program/back-end/deploy/gunicorn.conf.py

directory=/home/program/back-end

user=root

autostart=true

autorestart=true

redirect\_stderr=true

说明：

* command： 即启动 gunicorn 的命令，此处要写 绝对路径
* directory： 项目部署目录，不然没办法知道 command 中 main.py 模块在哪

有了配置文件，就可以通过 Supervisor 来管理 Gunicorn 了：

1. 增加配置文件后，需要更新

[root@CentOS ~]*# supervisorctl reread*

[root@CentOS ~]*# supervisorctl update*

2. 查看状态

[root@CentOS ~]*# supervisorctl status*

3. 启动/停止

[root@CentOS ~]*# supervisorctl start gunicorn*

[root@CentOS ~]*# supervisorctl stop gunicorn*

[root@CentOS ~]*# supervisorctl restart gunicorn*

此时，可以查看进程 ps -ef | grep gunicorn，会发现 Gunicorn 进程已经被 Supervisor 启动起来了：



# 2. 部署前端

## 2.1 npm run build

要先修改 front-end/src/http.js，指定后端 API 的地址为云服务器 IP：

*// 基础配置*

**if** (process.env.NODE\_ENV === 'production') {

axios.defaults.baseURL = 'http://121.36.253.244:5000';

} **else** {

axios.defaults.baseURL = 'http://127.0.0.1:5000';

}

然后，将前端代码打包成正式环境运行所需的静态文件：

front-end> npm run build

## 2.2 Nginx

前端代码等会打包后，就是一堆静态文件，用 Nginx 提供静态资源服务：

1. 安装

[root@CentOS ~]*# yum install -y nginx*

2. 启动

[root@CentOS ~]*# systemctl start nginx*

[root@CentOS ~]*# systemctl enable nginx*

将新生成的 dist/ 目录下 **所有文件** 上传到服务器的 /usr/share/nginx/html/ 目录下（压缩成单个 zip 文件再上传会比较快），云服务器上此目录最终文件列表如下：

[root@CentOS html]*# pwd*

/usr/share/nginx/html

[root@CentOS html]*# ls -l*

total 20

-rw-r--r-- 1 root root 3650 Oct 3 2019 404.html

-rw-r--r-- 1 root root 3693 Oct 3 2019 50x.html

lrwxrwxrwx 1 root root 20 Aug 12 02:43 en-US -> ../../doc/HTML/en-US

drwxr-xr-x 2 root root 4096 Aug 12 02:43 icons

lrwxrwxrwx 1 root root 18 Aug 12 02:43 img -> ../../doc/HTML/img

lrwxrwxrwx 1 root root 25 Aug 12 02:43 index.html -> ../../doc/HTML/index.html

-rw-r--r-- 1 root root 368 Oct 3 2019 nginx-logo.png

lrwxrwxrwx 1 root root 14 Aug 12 02:43 poweredby.png -> nginx-logo.png

drwxr-xr-x 6 root root 4096 Aug 14 06:47 static

我的服务器 IP 为 121.36.253.244，所以在浏览器访问 http://121.36.253.244：



成功！