



后端基础架构

准备工作

- 1.服务器计算机
 - 公司服务器
 - 云服务器
 -
- 2.虚拟环境
- 3.数据存储
 - Redis
 - Mysql
 -
- 4.框架程序
 - Flask
 - Django
 -
- 5.服务器程序
 - Gunicorn
 - uwsgi
 -
- 6.代理服务器
 - Nginx
 -

服务器计算机

- 1.目的
 - 运行后端程序的地方
- 2.使用
 - 远程云服务器地址 **101.200.170.171**
 - ssh 用户名@IP地址 **ssh zhangjie@101.200.170.171**

虚拟环境

- 1.安装虚拟环境
 - pip install virtualenv
 - pip install virtualenvwrapper
- 2.编辑~/.bashrc文件
 - 1.目的
 - 使安装的virtualenvwrapper生效
 - 2.内容
 - export WORKON_HOME=\$HOME/.virtualenvs
 - export PROJECT_HOME=\$HOME/workspace
 - source /usr/local/bin/virtualenvwrapper.sh
 - 3.执行
 - source ~/.bashrc
- 3.创建虚拟环境
 - mkvirtualenv -p python3 py3_flask
- 4.同步项目依赖包到服务器
 - 1.收集依赖包
 - workon py3_flask
 - pip freeze > requirements.txt
 - 2.远程拷贝
 - scp -r 本地文件路径 zhangjie@101.200.170.171:远程保存路径
 - 3.服务器安装
 - cd 到 requirements.txt文件路径
 - pip install -r requirements.txt

数据库

- 0.更新apt库
 - sudo apt-get update
- 1.安装MySQL
 - apt-get install mysql-server
 - apt-get install libmysqlclient-dev
- 2.安装Redis
 - sudo apt-get install redis-server

框架程序

- 1.目的
 - 将开发者电脑上的服务器程序同步到服务器
 - 遵守wsgi协议
 - 执行业务逻辑的程序
- 2.方式一
 - scp -r 本地项目路径 zhangjie@101.200.170.171:远程保存路径
- 3.方式二
 - 1.远程服务器安装git
 - 2.git clone

新闻数据准备

- 1.创建数据库
 - create database information_sy09 charset=utf8;
- 2.迁移建表
 - python manage.py db init
 - python manage.py db migrate
 - python manage.py db upgrade
- 3.将新闻分类和新闻数据拷贝到远程服务器
 - scp -r 本地文件路径 zhangjie@101.200.170.171: 远程保存路径
 - information_info_category.sql
 - information_info_news.sql

服务器程序

- 1.目的
 - 接受和转发请求报文给框架程序
 - 接受和转发响应报文给客户端、Nginx
 - 遵守wsgi协议
- 2.安装
 - workon到虚拟环境
 - pip install 服务器程序名字
 - pip install gunicorn
- 3.启动框架程序
 - 1.进入到框架程序目录
 - 2.启动
 - gunicorn 指定进程数 绑定IP:PORT 项目入口文件名:Flask程序实例名
 - gunicorn -w 2 -b 127.0.0.1:5000 项目入口文件名:Flask程序实例名

代理服务器

- 1.目的
 - 分流
 - 转发
 - 负载均衡
 - 2.安装
 - sudo apt-get install nginx
 - 3.配置
 - sudo vim /etc/nginx/sites-available/default
 - # 如果是多台服务器的话,则在此配置,并修改 location 节点下面的 proxy_pass

```
upstream flask {
    server 127.0.0.1:5000;
    server 127.0.0.1:5001;
}
server {
    # 监听80端口
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # 请求转发到gunicorn服务器
        proxy_pass http://flask;
        # 请求转发到多个gunicorn服务器
        # proxy_pass http://flask;
        # 设置请求头,并将头信息传递给服务器端
        proxy_set_header Host $host;
        # 设置请求头,传递原始请求ip给 gunicorn 服务器
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```
- 4.开启和停止
 - /etc/init.d/nginx start #启动
 - /etc/init.d/nginx stop #停止

用户访问

- Nginx服务器IP:端口
 - 端口默认是80,可以省略

部署