建筑咨询管理系统服务器迁移教程

Author:superzhaoyang

1.迁移数据库

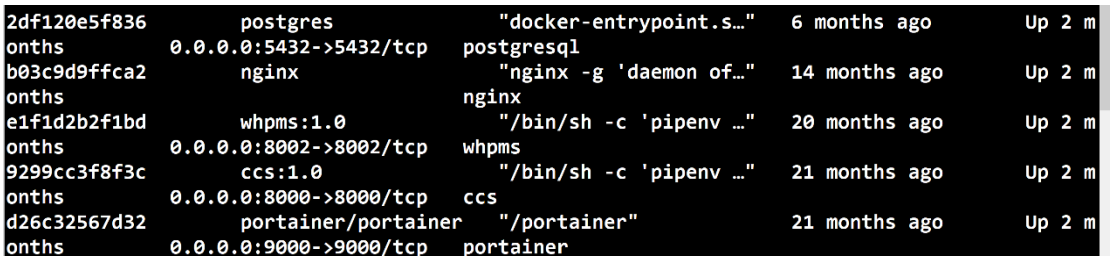
首先在新的服务器上拉取 **postgresql**,版本为 **11.2。**

拉取命令：**docker pull postgres:11.2-alipine**,

运行命令为:

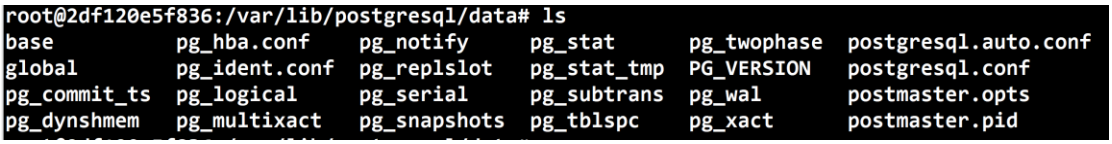
```
docker run \
    --name postgresql \
    -v /app/postgresql/data:/var/lib/postgresql/data \
    -e POSTGRES_PASSWORD=3390tsl \
    --restart=always \
    -d \
    -p 5432:5432 \
    Postgres
```

接下来进入旧的服务器上运行命令 **docker ps**，找到 **postgresql**，如下图所示



运行命令 **docker exec -it postgresql /bin/bash**,进入容器,

运行命令 **cd /var/lib/postgresql/data**,将如下图所示文件夹下的全部文件复制到新的服务器的 **postgresql** 的容器的对应位置中。



至此，数据库迁移完成。

2.代码迁移

首先将最新的代码复制进新的服务器，然后利用 **docker build .**(注意有个点)，命令构建镜像文件，然后运行命令:

```
docker run -d \
    --name ccs \
    -p 8000:8000 \
    --restart=always \
    -v /app/ccs/config/:/config/ \
    -v
/app/ccs/ConstructionConsultationSystem/:/ConstructionConsultationSystem/ \
    ccs:1.0
```

至此，**Django 已经启动**

运行 **docker logs ccs**，查看是否有报错;

如果有报错，则运行 **docker exec -it ccs /bin/bash**，再运行

pipenv shell,再运行 **pip install django==2.2.0**,退出容器，运行重启命令：**docker restart ccs**,应该可以解决报错问题。

3.部署 uwsgi 和 Nginx

复制原服务器的 **config** 目录，里面有个 **uwsgi.ini**，要限制一下最大进程数，否则会吃很多内存。

接下来，运行 **sudo apt-get install nginx,**

修改 nginx 配置文件如下：

```
server {
    listen 8081;
    server_name 60.217.64.234;
    charset UTF-8;
    client_max_body_size 75M;

    location /static {
        alias /home/yuda/CCS/ConstructionConsultationSystem/static;
    }
    location /media {
        alias /home/yuda/CCS/ConstructionConsultationSystem/media;
    }
    location / {
        uwsgi_pass 127.0.0.1:8000;
        include uwsgi_params;
    }
}
```

然后执行 **nginx -s reload**

最后要进入 **ccs**,再 **pipenv shell**，运行 **python manage.py collectstatic**(此操作是为了收集静态文件)。

项目工作逻辑：nginx 作为一个 http 和反向代理服务器，通过监听外部的 8081 端口，将外部的 8081 端口的请求转发至 8000 端口，uwsgi 将请求再转发给服务器上的 8000 端口，服务器上的 8000 端口再映射到 docker 内部的 8000 端口。

WSGI

WSGI 是一种 WEB 服务器网关接口。 是一个 Web 服务器（如 nginx）与应用服务器（如 uWSGI）通信的一种规范（协议）。

在生产环境中使用 WSGI 作为 python web 的服务器。Python Web 服务器网关接口，是 Python 应用程序或框架和 Web 服务器之间的一种接口，被广泛接受。WSGI 没有官方的实现，因为 WSGI 更像一个协议，只要遵照这些协议，WSGI 应用(Application)都可以在任何服务器(Server)上运行。

uWSGI

uWSGI 实现了 WSGI 的所有接口，是一个快速、自我修复、开发人员和系统管理员友好的服务器。uWSGI 代码完全用 C 编写，效率高、性能稳定。

uwsgi 是一种线路协议而不是通信协议，在此常用于在 uWSGI 服务器与其他网络服务器的数据通信。uwsgi 协议是一个 uWSGI 服务器自有的协议，它用于定义传输信息的类型。

作用

Django 是一个 Web 框架，框架的作用在于处理 request 和 reponse，其他的不是框架所关心的内容。所以怎么部署 Django 不是 Django 所需要关心的。

Django 所提供的是一个开发服务器，这个开发服务器，没有经过安全测试，而且使用的是 Python 自带的 simple HTTPServer 创建的，在安全性和效率上都是不行的

而 uWSGI 是一个全功能的 HTTP 服务器，他要做的就是把 HTTP 协议转化成语言支持的网络协议。比如把 HTTP 协议转化成 WSGI 协议，让 Python 可以直接使用。提升安全性和效率。uwsgi 是一种 uWSGI 的内部协议，使用二进制方式和其他应用程序进行通信。

注意：

1. 部署过程绝对没有我所写的那么简单，但是这个是一个比较准确的绝大部分流程，你要根据服务器的实际目录对 docker 命令进行修改和相应的 conf 文件进行配置。

2. 如果你没学过 nginx 和 docker 以及 django，看这个教程就有点像听天书。因为不能深刻地理解项目各组件之间的关系。所以我我建议部署之前储备一些相关知识，理清项目结构。