

在阿里云上部署一个 flask 项目

发件人: 春天花开开<no-reply@yinxiang.com>

时 间: 2019年7月7日(星期天) 晚上7:27

收件人: 春天花开开<1663177102@qq.com>

部署hello world到阿里云服务器

在阿里云上部署flask项目，需要三步：

- 1.配置阿里云
- 2.配置gunicorn
- 3.配置nginx

阿里云的配置

基于centos-7.4系统（阿里云实例），使用 Gunicorn + Nginx 进行部署，云服务器为阿里云

你的阿里云服务器会有一个实例，需要做一些配置

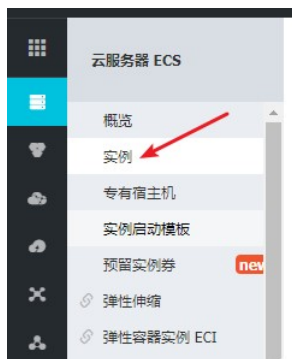
登录阿里云

如果忘了你的密码或者用户名，就给阿里云的客服打电话吧，他会给你安排的明明白白的。

点击控制台



找到你的实例（如果你之前买好了，注意你在哪个区）



配置安全组

我们部署的时候，nginx会用到80端口，gunicorn会用到5000端口，所以需要提前配置一下



sg-wz92ty70orrb2lpnmfm... / vpc-wz98wm47h73gm7dji3r26

教我设置 返回 添加安全组规则

入方向		出方向								导入规则
<input type="checkbox"/>	授权策略	协议类型	端口范围	授权类型(全部)	授权对象	描述	优先级	创建时间		
<input type="checkbox"/>	允许	自定义 TCP	80/80	IPv4地址段访问	0.0.0.0/0	-	1	2019年3月31日 09:50	修改	
<input type="checkbox"/>	允许	自定义 TCP	5000/5000	IPv4地址段访问	0.0.0.0/0	-	1	2019年3月15日 20:39	修改	
<input type="checkbox"/>	允许	自定义 TCP	3389/3389	IPv4地址段访问	0.0.0.0/0	System created rule.	110	2018年10月20日 11:20	修改	
<input type="checkbox"/>	允许	全部 ICMP(IPv4)	-1/-1	IPv4地址段访问	0.0.0.0/0	System created rule.	110	2018年10月20日 11:20	修改	
<input type="checkbox"/>	允许	自定义 TCP	22/22	IPv4地址段访问	0.0.0.0/0	System created rule.	110	2018年10月20日 11:20	修改	

添加安全组规则

网卡类型: 内网

规则方向: 入方向

授权策略: 允许

协议类型: 自定义 TCP

* 端口范围: 5000/5000

优先级: 1

授权类型: 地址段访问

* 授权对象: 0.0.0.0/0

描述:

长度为2-256个字符, 不能以http://或https://开头。

确定 取消

https://blog.csdn.net/qq_43920774

按照这个格式把80和5000端口配置好

在自己的虚拟机上搭建一个小项目

工程项目代码

helloworld.py

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return 'hello world'

if __name__ == '__main__':
    app.run()
```

gunicorn

搭建好以后跑一下, 没问题的话就可以。但是这是基于falsk自带的web服务器, 所以在阿里云上要用gunicorn来代替, 它能让你的程序跑起来。

安装

```
pip install gunicorn
```

使用gunicorn 启动工程

通过命令行启动它

```
gunicorn -w worker数量 -b ip:端口号 运行文件名: flask实例名
例:
gunicorn -w 3 -b 127.0.0.1:8000 helloworld:app
```

这里说明一下

后台运行

```
gunicorn -D -w 3 -b 127.0.0.1:8000 helloworld:app
```

gunicorn命令的参数:

- D 表示后台运行
- w 表示有3 个 工作线程 (感觉有些类似 nginx 的 master-worker 模型)
- b 指定ip 和端口
- 这里采用本机访问, 主要是为了使用nginx 进行代理, 方便管理
- application 表存放 写着全局变量 app 的那个工程文件夹
- 在我们的这个工程中, 即包含 init.py 的那个文件

- app 为全局变量 (app = Flask(__name__))
- ps : 在下面图示文件夹中运行 gunicorn 指令

到达这一步, 可以在linux服务器本地上访问(127.0.0.1:8000)

注意: 如果想通过本地Windows访问的话, 还需安装nginx, 把ip改为阿里云的公网的ip.

nginx

详细流程参考 <https://www.runoob.com/linux/nginx-install-setup.html>

nginx安装

```
cd /usr/local/src
wget http://nginx.org/download/nginx-1.1.10.tar.gz
tar -zxvf nginx-1.1.10.tar.gz
cd nginx-1.1.10
./configure
make
make install
```

nginx配置

1. 安装完的nginx 在 /usr/local/nginx 目录下, 我们可以找到conf文件夹下的 nginx.conf 文件, 将其修改

2. 添加nginx 需要监听的端口信息

```
server {
    listen 8001;
    server_name localhost;
    location /{
        proxy_pass http://127.0.0.1:8000; }
}
```

```
server {
    listen      8088;
    server_name localhost;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    location / {
        #root    html;
        #index  index.html index.htm;
        #请求转发到gunicorn服务器
        proxy_pass http://127.0.0.1:5000;
        #设置请求头, 并将头信息传递给服务器端
        proxy_set_header Host $host;
    }
}
```

实际上, 我们将外部通过 8001 端口发送过来的请求, 代理给了 127.0.0.1:8000 也就是我们的 flask 项目部署项目时 请求流程

Client <====> Nginx <====> gunicorn <====> Flask服务

记得修改完配置文件, 需要重新载入配置文件! !!

检查配置文件nginx.conf是否正确

```
$ /usr/local/nginx/sbin/nginx -t
```

启动nginx

```
$ /usr/local/nginx/sbin/nginx
```

nginx其他命令

```
$ /usr/local/nginx/sbin/nginx -s reload # 重新载入配置文件
$ /usr/local/nginx/sbin/nginx -s reopen # 重启nginx
$ /usr/local/nginx/sbin/nginx -s stop # 停止nginx
```

nginx快速停止

```
[root@mylinux nginx]# ps -ef | grep nginx
root      25276      1  0 17:19 ?        00:00:00 nginx: master process /usr/local
nobody    25277 25276   0 17:19 ?        00:00:00 nginx: worker process
root      25363 25179   0 18:41 pts/4    00:00:00 grep --color=auto nginx
[root@mylinux nginx]#
```

\$ ps -ef | grep nginx #查看进程号

\$ kill -9 25276 #杀死进程

\$ pkill -9 nginx # 强制停止

小结

将要部署的项目上传至服务器，安装gunicorn,开启gunicorn,在服务器本地上调通url。然后在阿里云上安装nginx，编辑配置文件，开启nginx，通过公网ip访问一下看看是否成功。