# 云服务器ECS环境搭建及配置

云服务器：

https://home.console.aliyun.com/#/

## 系统基本信息

Ubuntu 12.04

CPU 1核

内存 512M 带宽1M

所在可用区： 青岛可用区B

# 部署前的准备

## 一 安装Xshell

登陆服务器进行服务的部署

首先检查 python版本

Python –v

创建目录

Mkdir /home/kuibu/

Mkdir /kuibu/images/

Mkdir /kuibu/images/user\_photos/

Mkdir /kuibu/images/topic/

Mkdir /kuibu/images/cover

Mkdir /kuibu/images/thumbnail/

Mkdir /kuibu/log/

二、安装WINSCP (或者其他FTP客户端)，服务端需安装配置vsftp(具体见附录一)

登陆用来上传文件

还可以用来查看文件结构

## 三 安装mysql 及mysqldb

Sduo Apt-get update

Sudo apt-get mysql-server

Sudo apt-get python-mysqldb

四 安装nginx

sudo apt-get nginx

sudo /etc/init.d/nginx start

Nginx的配置文件是/etc/nginx/nginx.conf

具体的配置信息见后

安装后在本地测试是否成功

http://115.28.200.109/

## 四 Mysql 服务

客户端连接服务端的问题

安装完mysql 后，为客户端创建一个用户

Use mysql ;

CREATE USER kuibu IDENTIFIED BY “123”

123为远端的登录密码

Grant all privileges on \*.\* to ‘ kuibu’@’%’ WITH GRANT OPTION

kuibu 表示用户 % 表示远端的任意主机 如果主机IP知道了就不要这么设置了

# 搭建环境

首先下载virtualenv.py

(这里使用已经下载好的，后面的可以略过)

通过命令

$sudo easy\_install virtualenv

或者

$sudo pip install virtualenv

Unbuntu 下可以通过软件包管理

Sudo apt-get install python-virtualenv

## 创建虚拟环境

Cd 到 /home/kuibu/

$ virtualenv flask 或者使用下载好的virtualenv.py 创建 python virtualenv.py flask ;

每次使用时首先激活虚拟环境

$ source ./flask/bin/activate

创建项目目录

Mkdir /flask/apps/ 其中存放我们服务端项目的代码

## 安装flask

完成上面的步骤，接下来就来安装flask等其附属的第三方模块：

$ flask/bin/pip install flask

$ flask/bin/pip install flask-sqlalchemy

$ flask/bin/pip install flask-httpauth

$ flask/bin/pip install redis

$ flask/bin/pip install MySQL-python #mysqldb

$ flask/bin/pip install passlib

$flask/bin/pip install flask-socketio

$flask/bin/pip install PIL (具体看附录的PIL安装)

## nginx配置

sudo apt-get nginx 这里需要>1.4的稳定版

或许最新版稳定版参见

http://www.kaijia.me/2013/05/ubuntu-latest-nginx-repo-collection/

sudo /etc/init.d/nginx start

配置文件 /etc/nginx/sites-avaliable/default

nginx的优化配置文件在

/etc/nginx/nginx.conf

# 部署

基本架构 nginx + flask + gunicorn + gevent

## Nginx反向代理及配置

跬步使用到socketio， 部署要使用 [gunicorn](http://gunicorn.org/)  作为web服务容器，所以首先要安装

先切换到虚拟环境，然后pip install gunicorn 即可，详见<http://gunicorn.org/>

注意这里需要加上版本号 pip install gunicorn==18.0

Gunicorn 作为web服务容器，前端使用nginx做反向代理，配置如下

server {

listen 80;

server\_name localhost;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

location /socket.io {

proxy\_pass http://127.0.0.1:5000/socket.io;

proxy\_redirect off;

proxy\_buffering off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

location ~ ^/(images|javascript|js|css|flash|media|static|files)/{

root /home/kuibu/ ;

expires 30d ;

}

}

重启nginx .

## 在项目目录下编写配置文件gunicorn.conf

先切换到项目目录 cd apps ;

编写配置文件gunicorn.conf

workers = 4

backlog = 2048

bind = '127.0.0.1:5000'

worker\_class="socketio.sgunicorn.GeventSocketIOWorker" #sync,gevent,meinheld,socketio.sgunicorn.GeventSocketIOWorker

proc\_name='kuibu'

pidfile= '/tmp/kuibu.pid'

logfile = '/home/kuibu/log/debug.log'

debug=True

daemon=True

errorlog='/home/kuibu/log/error.log'

..bin/gunicorn --config gunicorn.conf run:app

启动成功后，通过命令查看后台进程

ps -ef | grep kuibu

### 参考链接

http://flask-socketio.readthedocs.org/en/latest/

http://www.open-open.com/lib/view/open1423107543014.html

## 使用 supervisor 管理gunicorn 进程

基本用法

**##sudo安装** sudo apt-**get** install supervisor

**##启动服务** sudo service supervisor start

**##终止服务** sudo service supervisor stop

**##也可以直接kill pid** ps -A | grep supervisor

参考配置

配置示例

*##文件内容*

[program:dylan]

*##注意项目目录和gunicorn的配置文件地址*

command=/home/bob/dylan/bin/gunicorn runserver:app -c /home/bob/dylan/gunicorn.conf directory=/home/bob/dylan

user=root

autostart=true

autorestart=true

*##log文件的位置*

stdout\_logfile=/home/bob/logs/gunicorn\_supervisor.log

##重新读取配置

su**do** supervisorctl reread

##启动服务 sudo supervisorctl **start** kuibu

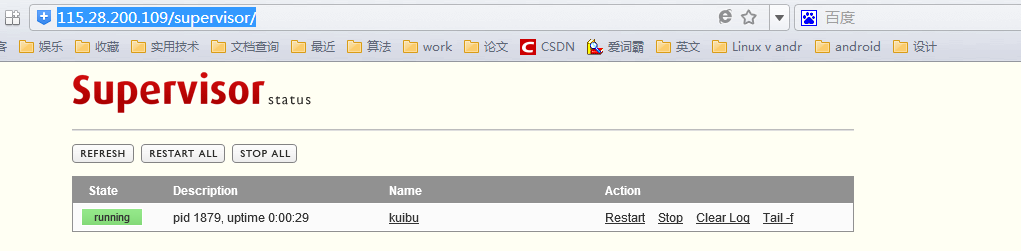
##停止服务 sudo supervisorctl stop kuibu

##有问题就重启supervisor的总服务

sudo service supervisor stop

sudo service supervisor **start**

**通过浏览器管理supervisor**



配置/etc/supervisor/supervisord.conf

添加以下内容

[unix\_http\_server]

file=/var/run//supervisor.sock ; (the path to the socket file)

chmod=0700 ; sockef file mode (default 0700)

chown=nobody:nogroup

username = esli

password = 123

[inet\_http\_server]

port = 127.0.0.1:9001

username = esli

password = 123

详见：<http://supervisord.org/configuration.html#inet-http-server-section-settings>

在对在nginx添加配置

location /supervisor {

proxy\_pass http://127.0.0.1:9001/;

proxy\_redirect off;

proxy\_buffering off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

}

这样我们就可以远程对其进行管理。

# 测试

安装ab测试工具

sudo apt-get install apache2-utils

简单测试

$time ab -n 200 -c 200 <http://115.28.200.109/caddy/api/v1.0/test>

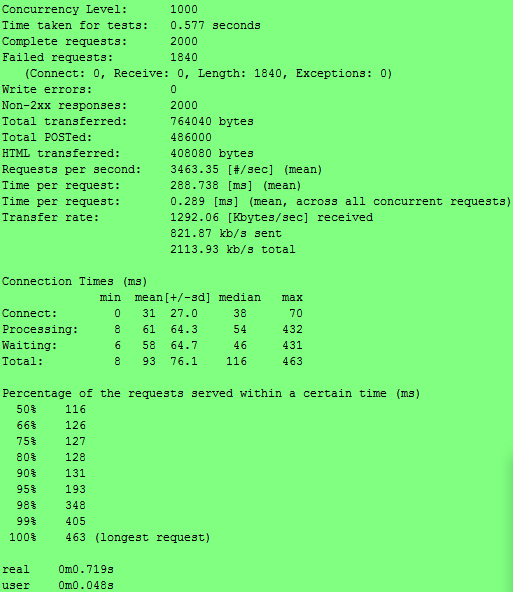
实际测试

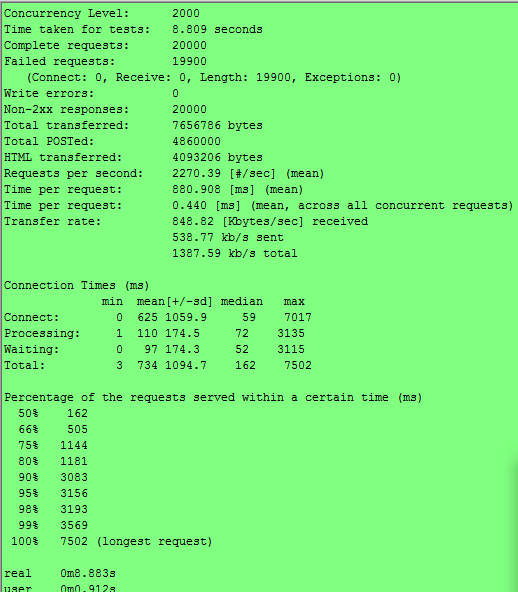
time ab -n 2000 -c 1000 -H keywords:dt -p /home/kuibu/param.conf -T 'application/json; charset=utf8' <http://115.28.200.109/caddy/api/v1.0/get_collections>

其中params.conf 的内容如下：

data\_type=HOME\_LIST&uid=9&action=INIT&threshold=-1

几次测试结果





**在远程对web服务器进行压力测试，往往效果不理想（因为网络延时过大），建议使用内网的另一台或者多台服务器通过内网进行测试，这样得出的数据，准确度会高很多。如果只有单独的一台服务器，可以直接本地测试，比远程测试效果要准确。**

**关于ab的使用，参考http://www.ha97.com/4617.html**

# 附录一 安装其他模块

1、Ubuntu/Debian安装vsftp软件

1.更新软件源

首先需要更新系统的软件源，便捷工具下载地址：http://help.aliyun.com/manual?spm=0.0.0.0.zJ3dBU&helpId=1692

2.安装vsftp

使用apt-get命令安装vsftp

#apt-get install vsftpd –y

3.添加ftp帐号和目录

先检查一下nologin的位置，通常在/usr/sbin/nologin或者/sbin/nologin下。

使用下面的命令创建帐户，该命令指定了/alidata/www/wwwroot为用户pwftp的家目录，您可以自己定义帐户名和目录：

#useradd -d /alidata/www/wwwroot -s /sbin/nologin pwftp

修改该帐户密码:

#passwd pwftp

修改指定目录的权限

#chown -R pwftp.pwftp /alidata/www/wwwroot

4.配置vsftp

编辑vsftp配置文件，命令如下：

#vi /etc/vsftpd.conf

将配置文件中”anonymous\_enable=YES “改为 “anonymous\_enable=NO”

取消如下配置前的注释符号：

local\_enable=YES

write\_enable=YES

chroot\_local\_user=YES

chroot\_list\_enable=YES

chroot\_list\_file=/etc/vsftpd.chroot\_list

保存退出

编辑/etc/vsftpd.chroot\_list文件，将ftp的帐户名添加进去，保存退出

5.修改shell配置

vi编辑/etc/shells，如果该文件里没有/usr/sbin/nologin或者/sbin/nologin（具体看当前系统配置）则追加进去

6.重启vsftp服务并测试登录

使用命令启动vsftp服务：

#service vsftpd restart

然后用帐号pwftp测试下是否可以登陆ftp。目录是/alidata/www/wwwroot。

2.PIL安装

具体参见

<http://jinvan.com/post/install-pil-on-ubuntu>

安装依赖包

sudo apt-get install libjpeg8-dev zlib1g-dev libfreetype6-dev

创建链接，因为PIP安装软件包时，并没有安装在系统路径里，因此做一个软连接。

ln -s /usr/lib/`uname -i`-linux-gnu/libfreetype.so /usr/lib/

ln -s /usr/lib/`uname -i`-linux-gnu/libjpeg.so /usr/lib/

ln -s /usr/lib/`uname -i`-linux-gnu/libz.so /usr/lib/

准备工作做好了，安装PIL

sudo pip install pil

sudo pip install PIL --allow-external PIL --allow-unverified PIL

# 附录二 Error 汇总

1. 安装Mysql-python 报错

Python.h: No such file or directory

需要 $ sudo apt-get install python-dev

1. IOError: [Errno 26] Text file busy: 'flask/bin/python'

Lsof

unlink flask/bin/python

1. greenlet.h:8:20: fatal error: Python.h: No such file or directory

解决 sudo apt-get install python-dev

1. 安装MySQL-python EnvironmentError: mysql\_config not found

sudo apt-get install libmysqlclient-dev

1. 关于使用flask - socketio

../bin/gunicorn -c gunicorn.conf run:app

使用socketio.sgunicorn.GeventSocketIOWorker 代替了 gevent

报错AttributeError: 'socket' object has no attribute 'cfg\_addr'

这是由于gunicorn 版本问题引起的，当前版本如果>18.0 修会出现这个问题，

所以需要将当前版本回归到18.0及以下

https://github.com/miguelgrinberg/Flask-SocketIO/issues/93

# 附录三 Linux命令

1. 杀死服务进程

ps -ef | grep kuibu

kill -s 9 24578

1. 查看目录权限

Ll

1. 查看内存使用

Free –m

top

# 附录四 服务端配置

#coding:utf-8

'''

Created on 2015/4/22

@author: ThinkPad

'''

#server-config.py

db\_config={

'host':'localhost',

'user':'root',

'passwd':'ddshlmstzczx',

'db':'caddy\_v1',

'charset':'utf8'

};

#server domain ip  ,can config in db .

SERVER\_BASE\_URL = r'http://115.28.200.109:80';

#coding:utf-8

'''

Created on 2015/4/22

@author: ThinkPad

'''

#server-config.py

db\_config={

'host':'localhost',

'user':'root',

'passwd':'ddshlmstzczx',

'db':'caddy\_v1',

'charset':'utf8'

};

#server domain ip  ,can config in db .

SERVER\_BASE\_URL = r'http://115.28.200.109:80';

#coding:utf-8

'''

Created on 2015/4/22

@author: ThinkPad

'''

#server-config.py

db\_config={

'host':'localhost',

'user':'root',

'passwd':'ddshlmstzczx',

   'db':'caddy\_v1',

'charset':'utf8'

};

#server domain ip  ,can config in db .

SERVER\_BASE\_URL = r'http://115.28.200.109:80';

#db connection url

MYSQL\_DATABASE\_CONN\_URI='mysql+mysqldb://%s:%s@%s/%s?charset=%s' ;

#nginx image server url

NGINX\_IMAGE\_SERVER\_URI=SERVER\_BASE\_URL+'/images/';

NGINX\_IMAGE\_SERVER\_THUMBNAIL = SERVER\_BASE\_URL+'/images/thumbnail/'

NGINX\_IMAGE\_SERVER\_COVER = SERVER\_BASE\_URL+'/images/cover/';

#uploadimage directory

UPLOAD\_IAMGE\_ROOT\_PATH = r'/home/kuibu/images/';

UPLOAD\_IMAGE\_THUMBNAIL\_PATH = r'/home/kuibu/images/thumbnail/';

UPLOAD\_IMAGE\_COVER\_PATH = r'/home/kuibu/images/cover/';

UPLOAD\_LOG\_PATH = r'/home/kuibu/flask/apps/debug/exception/';

#user photo base path

UPLOAD\_USERPHOTO\_ROOT\_PATH = UPLOAD\_IAMGE\_ROOT\_PATH+r'user\_photos/src/';