一、项目部署

参考博客：

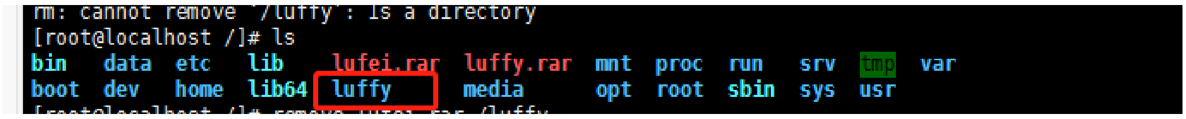
```
https://www.cnblogs.com/lvye001/p/10631276.html
```

第一步：上传项目到centos

```
rz   然后选择压缩包上传
```

第二步：创建luffy文件夹

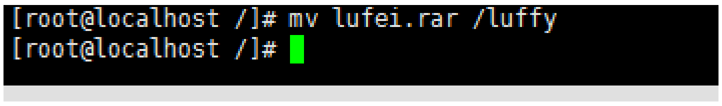
```
mkdir luffy
```



```
rm: cannot remove '/luffy': Is a directory
[root@localhost /]# ls
bin   data  etc   lib    lufei.rar  luffy.rar  mnt  proc  run   srv  tmp  var
boot  dev   home  lib64  luffy      media      opt  root  sbin  sys  usr
[root@localhost /]# remove lufei.rar /luffy
```

001、将路飞的项目前后端的项目压缩文件移动到luffy文件夹下

```
mv lufei.rar /luffy
mv luffy.rar /luffy
```

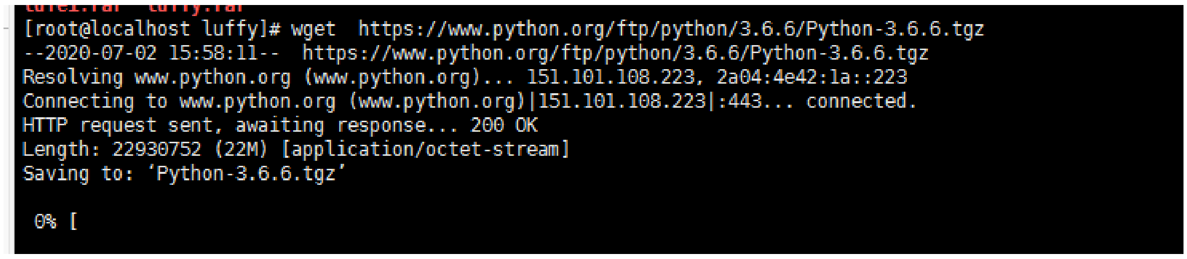


```
[root@localhost /]# mv lufei.rar /luffy
[root@localhost /]# 
```

第三步：安装python3，在centos界面使用python3说明安装成功

```

```

```
    wget  https://www.python.org/ftp/python/3.6.6/Python-3.6.6.tgz
```

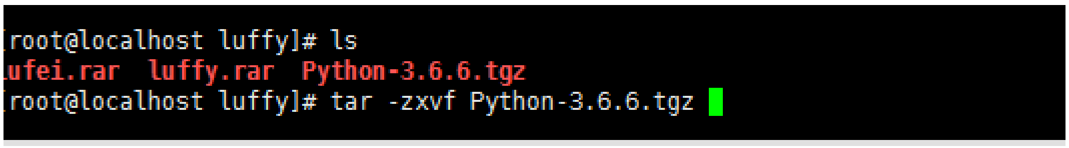


```
lufei.rar  luffy.rar
[root@localhost luffy]# wget  https://www.python.org/ftp/python/3.6.6/Python-3.6.6.tgz
--2020-07-02 15:58:11--  https://www.python.org/ftp/python/3.6.6/Python-3.6.6.tgz
Resolving www.python.org (www.python.org)... 151.101.108.223, 2a04:4e42:1a::223
Connecting to www.python.org (www.python.org)|151.101.108.223|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22930752 (22M) [application/octet-stream]
Saving to: 'Python-3.6.6.tgz'

 0% [
```

```
python
```

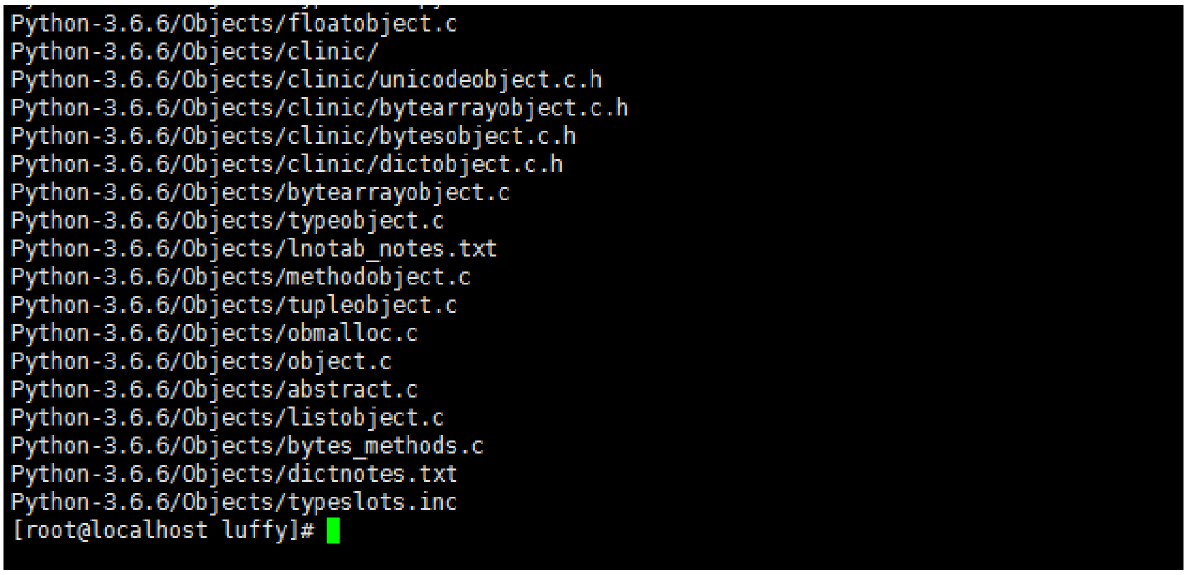
002、解压python压缩文件

```
tar -zxvf Python-3.6.6.tgz
```



```
root@localhost luffy]# ls
ufei.rar  luffy.rar  Python-3.6.6.tgz
root@localhost luffy]# tar -zxvf Python-3.6.6.tgz
```

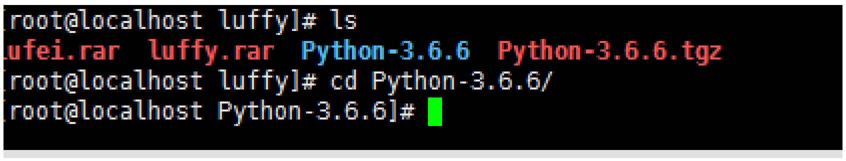
解压完成



```
Python-3.6.6/Objects/floatobject.c
Python-3.6.6/Objects/clinic/
Python-3.6.6/Objects/clinic/unicodeobject.c.h
Python-3.6.6/Objects/clinic/bytearrayobject.c.h
Python-3.6.6/Objects/clinic/bytesobject.c.h
Python-3.6.6/Objects/clinic/dictobject.c.h
Python-3.6.6/Objects/bytearrayobject.c
Python-3.6.6/Objects/typeobject.c
Python-3.6.6/Objects/lnotab_notes.txt
Python-3.6.6/Objects/methodobject.c
Python-3.6.6/Objects/tupleobject.c
Python-3.6.6/Objects/obmalloc.c
Python-3.6.6/Objects/object.c
Python-3.6.6/Objects/abstract.c
Python-3.6.6/Objects/listobject.c
Python-3.6.6/Objects/bytes_methods.c
Python-3.6.6/Objects/dictnotes.txt
Python-3.6.6/Objects/typeslots.inc
[root@localhost luffy]#
```

003、进入解压的python文件夹

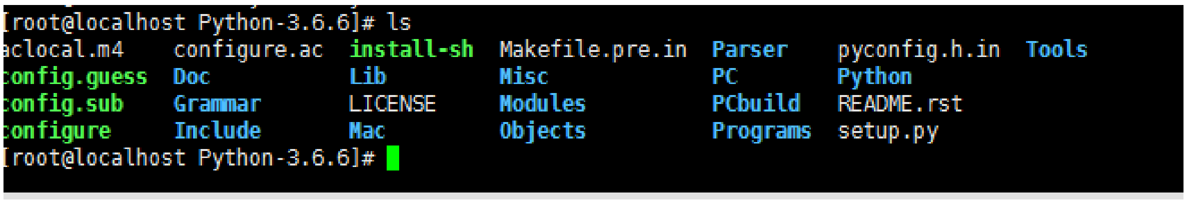
```
cd Python-3.6.6
```



```
root@localhost luffy]# ls
ufei.rar  luffy.rar  Python-3.6.6  Python-3.6.6.tgz
root@localhost luffy]# cd Python-3.6.6/
root@localhost Python-3.6.6]#
```

解压后的文件

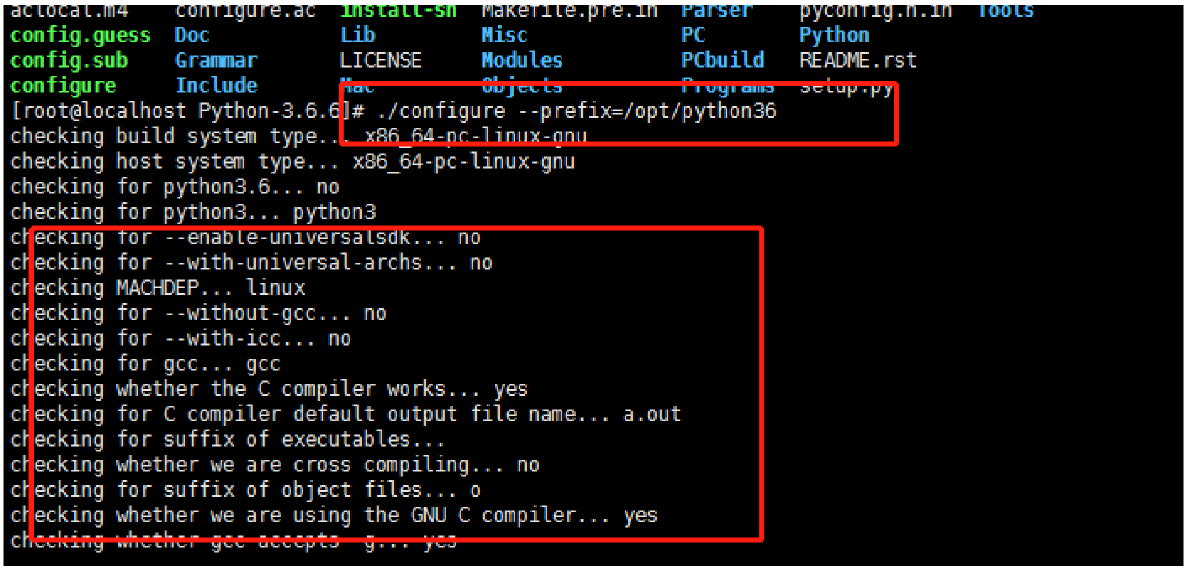
```
ls
```



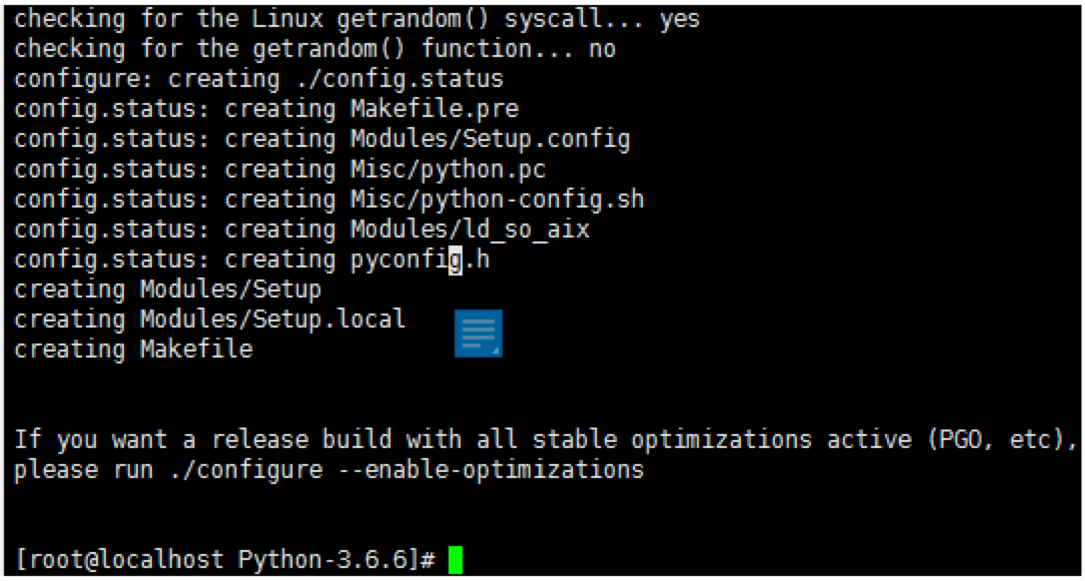
```
[root@localhost Python-3.6.6]# ls
aclocal.m4     configure.ac  install-sh  Makefile.pre.in  Parser    pyconfig.h.in  Tools
config.guess   Doc           Lib         Misc             PC        Python
config.sub     Grammar       LICENSE     Modules          PCbuild   README.rst
configure      Include       Mac         Objects          Programs  setup.py
[root@localhost Python-3.6.6]#
```

004、编译安装，先指定安装目录为指定安装目录为/opt/python36

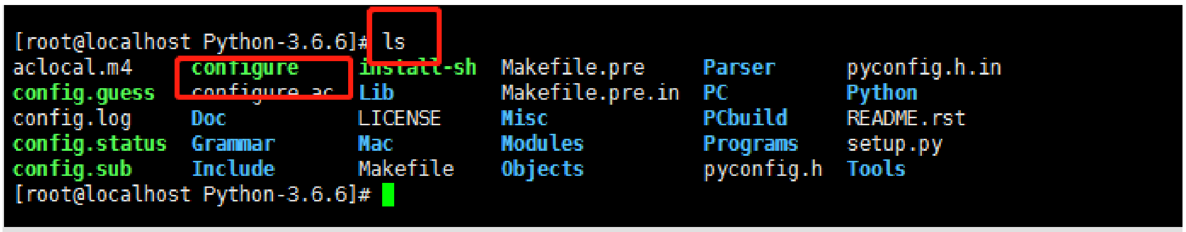
```
./configure --prefix=/opt/python36    # 指定安装目录为/opt/python36
```



```
aclocal.m4     configure.ac   install-sh     Makefile.pre.in  Parser     pyconfig.h.in  Tools
config.guess   Doc            Lib            Misc             PC         Python
config.sub     Grammar        LICENSE        Modules          PCbuild    README.rst
configure      Include        Mac            Objects          Programs   setup.py
[root@localhost Python-3.6.6]# ./configure --prefix=/opt/python36
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for python3.6... no
checking for python3... python3
checking for --enable-universalsdk... no
checking for --with-universal-archs... no
checking MACHDEP... linux
checking for --without-gcc... no
checking for --with-icc... no
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
```



```
checking for the Linux getrandom() syscall... yes
checking for the getrandom() function... no
configure: creating ./config.status
config.status: creating Makefile.pre
config.status: creating Modules/Setup.config
config.status: creating Misc/python.pc
config.status: creating Misc/python-config.sh
config.status: creating Modules/ld_so_aix
config.status: creating pyconfig.h
creating Modules/Setup
creating Modules/Setup.local
creating Makefile


If you want a release build with all stable optimizations active (PGO, etc),
please run ./configure --enable-optimizations


[root@localhost Python-3.6.6]#
```



```
[root@localhost Python-3.6.6]# ls
aclocal.m4     configure      install-sh  Makefile.pre     Parser     pyconfig.h.in
config.guess   configure.ac   Lib         Makefile.pre.in  PC         Python
config.log     Doc            LICENSE     Misc             PCbuild    README.rst
config.status  Grammar        Mac         Modules          Programs   setup.py
config.sub     Include        Makefile    Objects          pyconfig.h Tools
[root@localhost Python-3.6.6]#
```

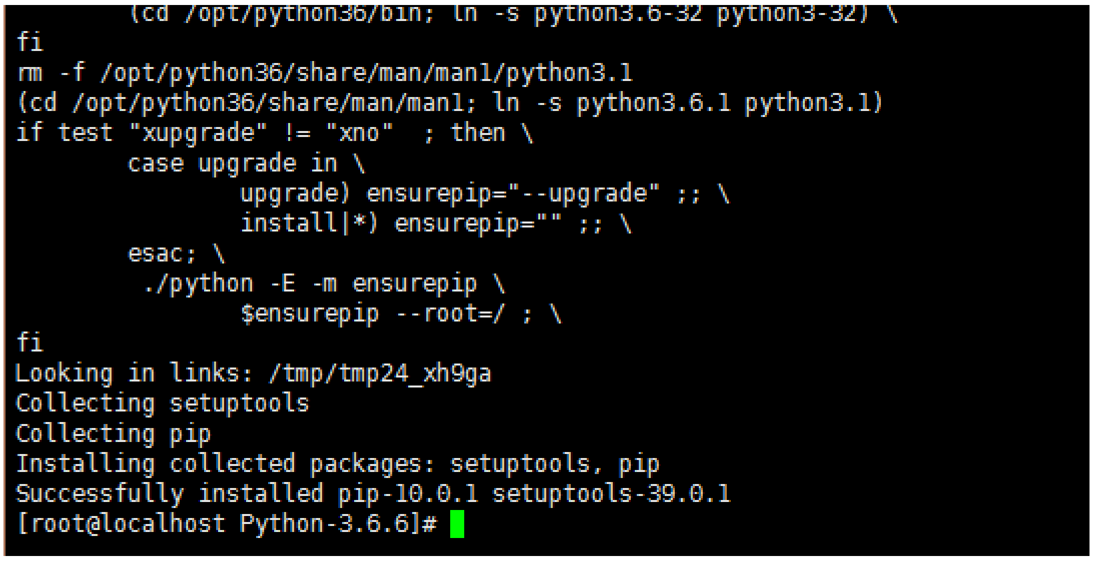
005、编译

```
make    # 相当于把源码包里面的代码编译成linux服务器可以识别的代码
```

006、5编译安装,此步才会最终生成 /opt/python36/

```
make install
```

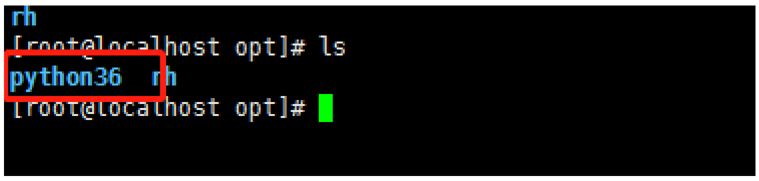
```
5 6 步骤可以合并:
make && make install  # 先进行make，执行完成之后，才开始执行make install
```



```
        (cd /opt/python36/bin; ln -s python3.6-32 python3-32) \
fi
rm -f /opt/python36/share/man/man1/python3.1
(cd /opt/python36/share/man/man1; ln -s python3.6.1 python3.1)
if test "xupgrade" != "xno"  ; then \
        case upgrade in \
                upgrade) ensurepip="--upgrade" ;; \
                install|*) ensurepip="" ;; \
        esac; \
         ./python -E -m ensurepip \
                $ensurepip --root=/ ; \
fi
Looking in links: /tmp/tmp24_xh9ga
Collecting setuptools
Collecting pip
Installing collected packages: setuptools, pip
Successfully installed pip-10.0.1 setuptools-39.0.1
[root@localhost Python-3.6.6]#
```

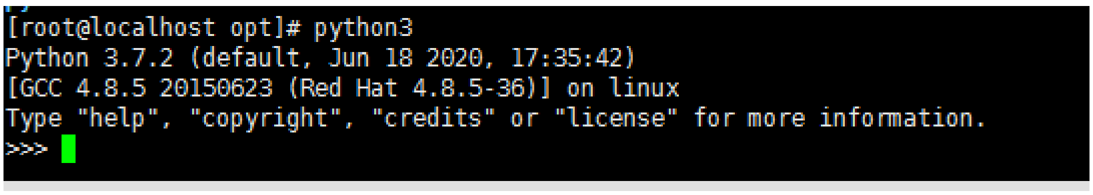
此时查看/opt/

```
cd /opt  # 可以看到出现了python36的文件夹
```



```
rh
[root@localhost opt]# ls
python36  rh
[root@localhost opt]#
```

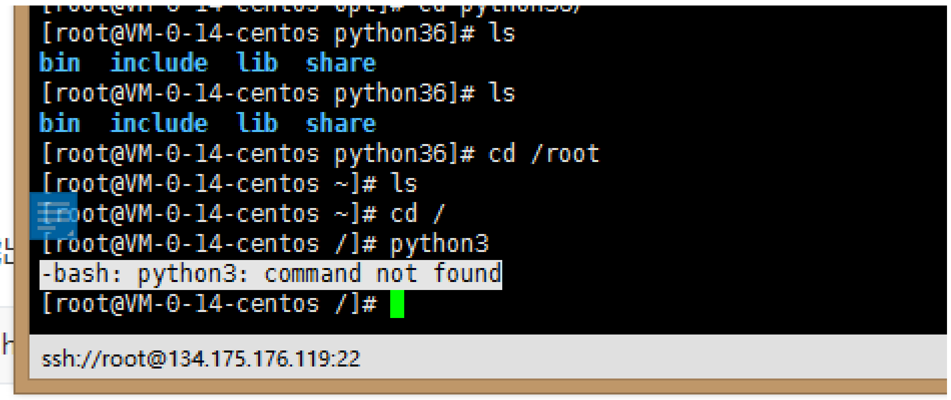
输入python3，说明安装成功了



```
[root@localhost opt]# python3
Python 3.7.2 (default, Jun 18 2020, 17:35:42)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-36)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

也可能出现：

```
-bash: python3: command not found
那就需要安装各种依赖包，百度一下就能找到,然后再重新编译安装
```



```
[root@VM-0-14-centos opt]# cd python36/
[root@VM-0-14-centos python36]# ls
bin  include  lib  share
[root@VM-0-14-centos python36]# ls
bin  include  lib  share
[root@VM-0-14-centos python36]# cd /root
[root@VM-0-14-centos ~]# ls
[root@VM-0-14-centos ~]# cd /
[root@VM-0-14-centos /]# python3
-bash: python3: command not found
[root@VM-0-14-centos /]#
```

ssh://root@134.175.176.119:22

007、将安装路径添加到centos的path下

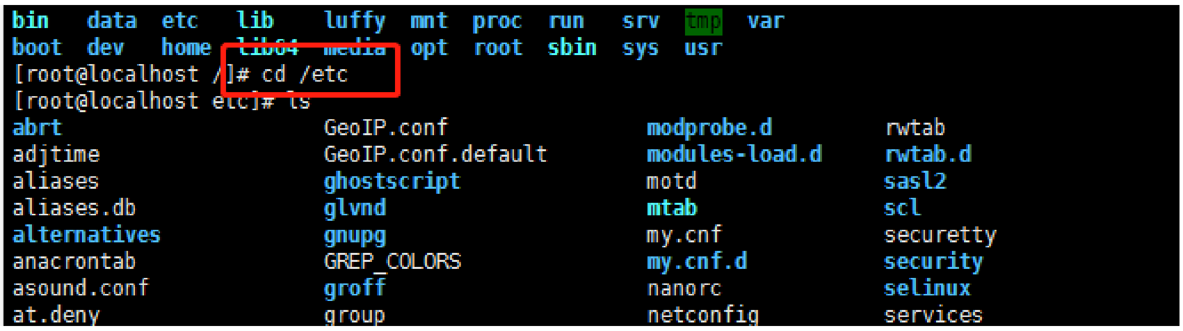
```
~/.bash_profile  # 这个是用户环境变量配置文件
/etc/profile  # 系统环境变量配置文件
```

修改/etc/profile系统环境变量配置文件，

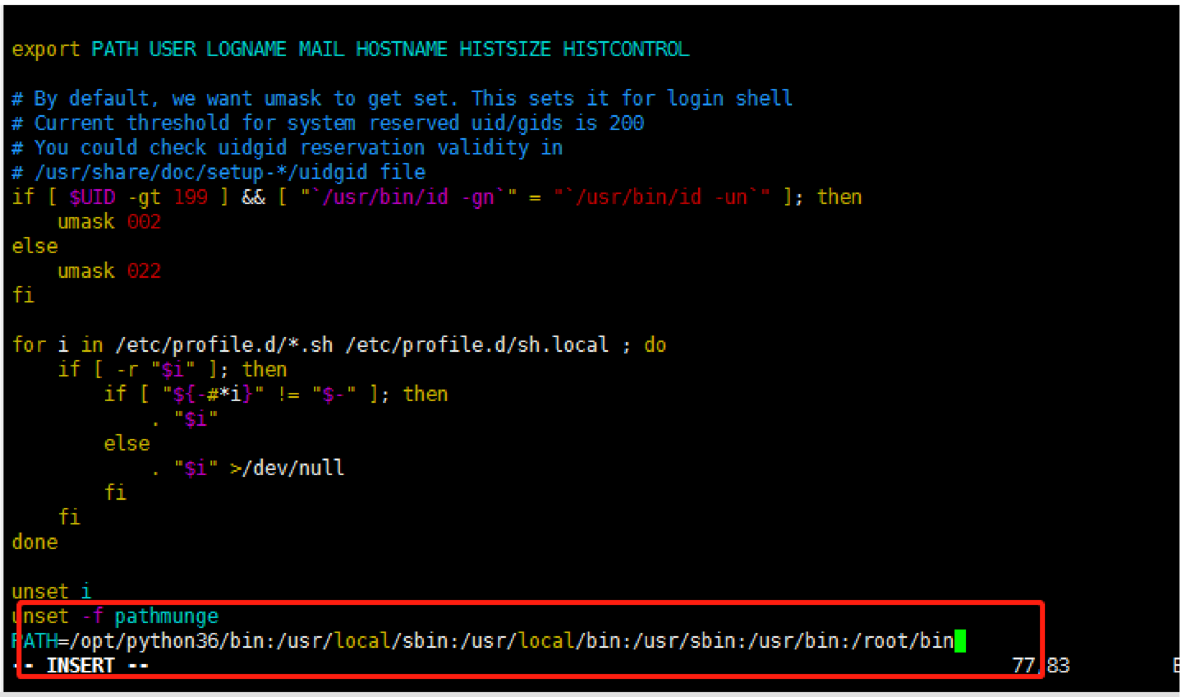
```
cd /etc
vim profile
```

添加以下内容

```
export
PATH=$path:/opt/python36/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
root/bin
```



```
bin   data  etc   lib    luffy  mnt  proc  run   srv  tmp  var
boot  dev   home  lib64  media  opt  root  sbin  sys  usr
[root@localhost /]# cd /etc
[root@localhost etc]# ls
abrt                      GeoIP.conf               modprobe.d         rwtab
adjtime                   GeoIP.conf.default       modules-load.d     rwtab.d
aliases                   ghostscript              motd               sasl2
aliases.db                glvnd                    mtab               scl
alternatives              gnupg                    my.cnf             securetty
anacrontab                GREP_COLORS              my.cnf.d           security
asound.conf               groff                    nanorc             selinux
at.deny                   group                    netconfig          services
```



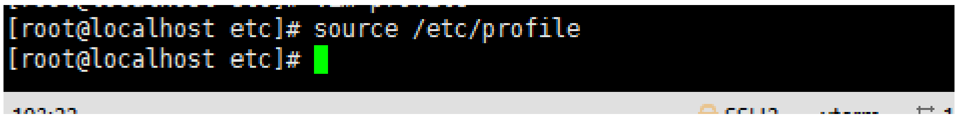
```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
PATH=/opt/python36/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
-- INSERT --                                                                77,83
```

注意： 一定要将python3的目录放在第一位

- 注意： 一定要将python3的目录放在第一位
- 注意： 一定要将python3的目录放在第一位
-

为了永久生效path设置，需要,重载配置文件/etc/profile

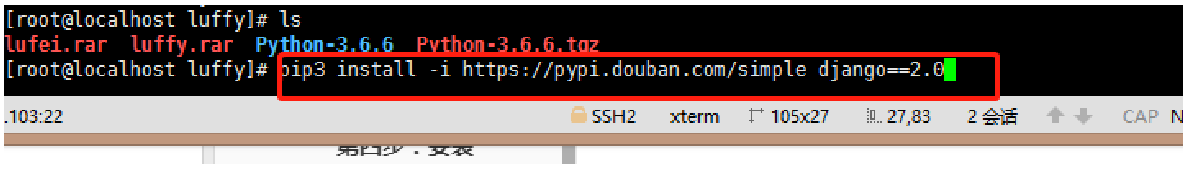
```
source /etc/profile
```



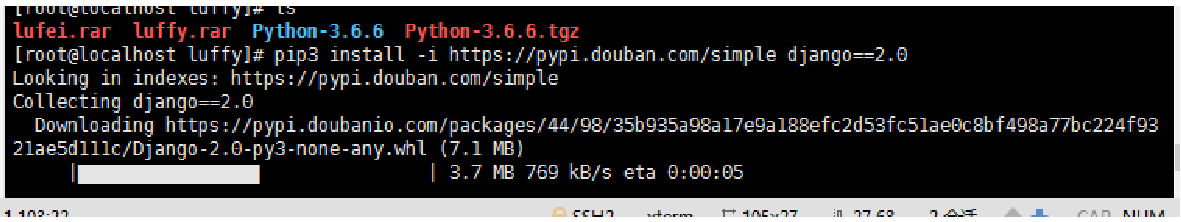
```
[root@localhost etc]# source /etc/profile
[root@localhost etc]# 
```

第四步：安装Django，这里安装的是django2.0

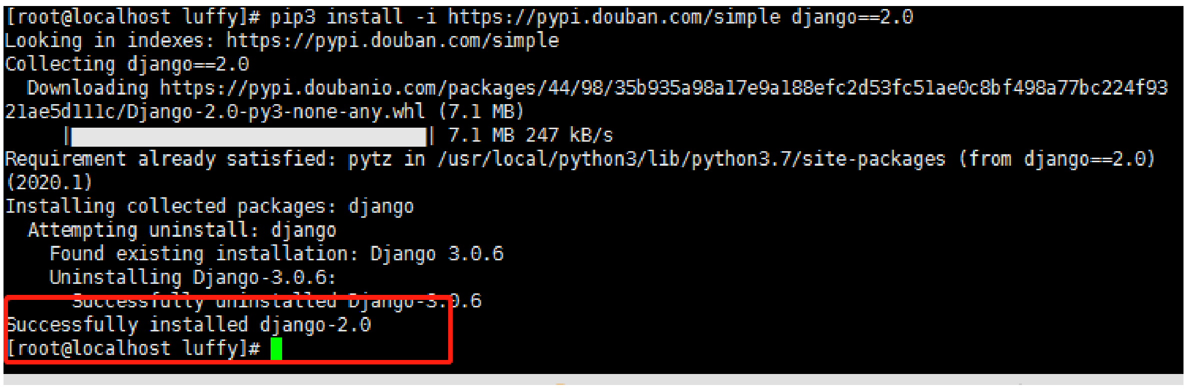
```
pip3 install -i https://pypi.douban.com/simple django==2.0
```



```
[root@localhost luffy]# ls
lufei.rar  luffy.rar  Python-3.6.6  Python-3.6.6.tgz
[root@localhost luffy]# pip3 install -i https://pypi.douban.com/simple django==2.0
```



```
lufei.rar  luffy.rar  Python-3.6.6  Python-3.6.6.tgz
[root@localhost luffy]# pip3 install -i https://pypi.douban.com/simple django==2.0
Looking in indexes: https://pypi.douban.com/simple
Collecting django==2.0
  Downloading https://pypi.doubanio.com/packages/44/98/35b935a98a17e9a188efc2d53fc51ae0c8bf498a77bc224f93
21ae5d111c/Django-2.0-py3-none-any.whl (7.1 MB)
     |█████████                        | 3.7 MB 769 kB/s eta 0:00:05
```



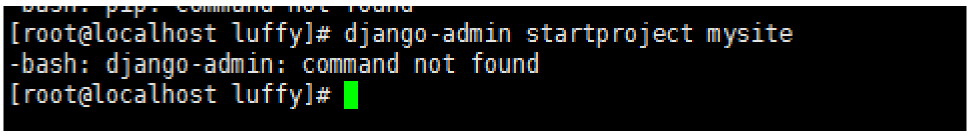
```
[root@localhost luffy]# pip3 install -i https://pypi.douban.com/simple django==2.0
Looking in indexes: https://pypi.douban.com/simple
Collecting django==2.0
  Downloading https://pypi.doubanio.com/packages/44/98/35b935a98a17e9a188efc2d53fc51ae0c8bf498a77bc224f93
21ae5d111c/Django-2.0-py3-none-any.whl (7.1 MB)
     |████████████████████████████████| 7.1 MB 247 kB/s
Requirement already satisfied: pytz in /usr/local/python3/lib/python3.7/site-packages (from django==2.0)
(2020.1)
Installing collected packages: django
  Attempting uninstall: django
    Found existing installation: Django 3.0.6
    Uninstalling Django-3.0.6:
      Successfully uninstalled Django-3.0.6
Successfully installed django-2.0
[root@localhost luffy]#
```

pip3 list,可能需要升级pip

```
pip3 install --upgrade pip
```

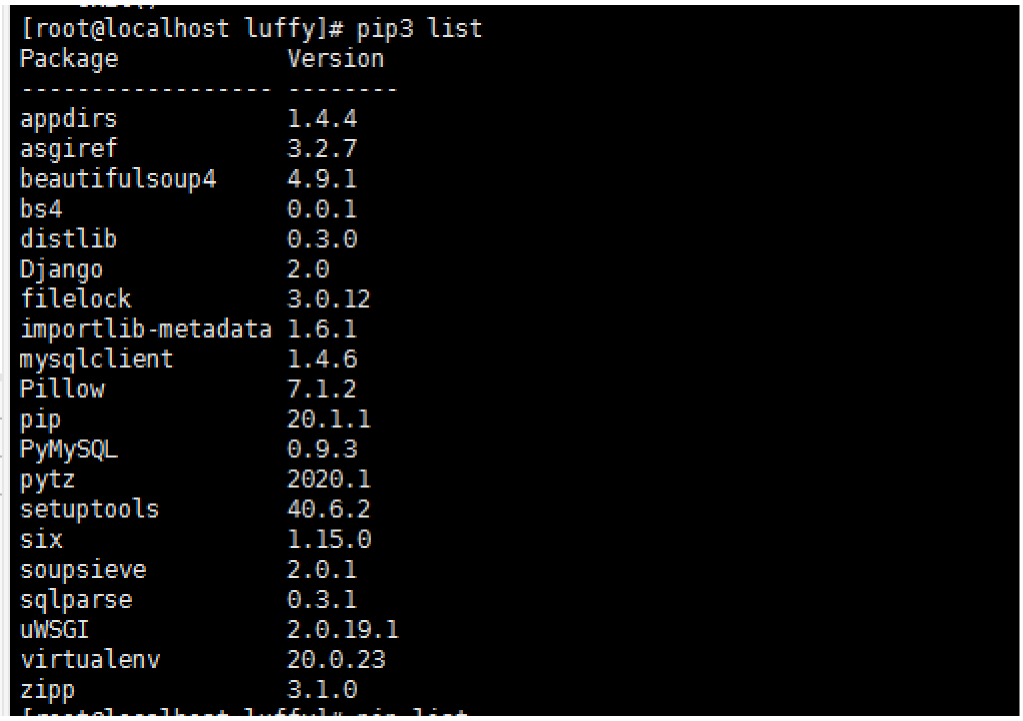
安装了python3，但是centos 里面是默认安装的python2.在python3种我安装了django，但是使用

```
django-admin startproject mysite  提示命令行不存在，这个时候可能是python的路径有问题，上名的路径我是又修改了的，所以应该不会再出现这个问题
```



```
[root@localhost luffy]# django-admin startproject mysite
-bash: django-admin: command not found
[root@localhost luffy]#
```

pip3 list

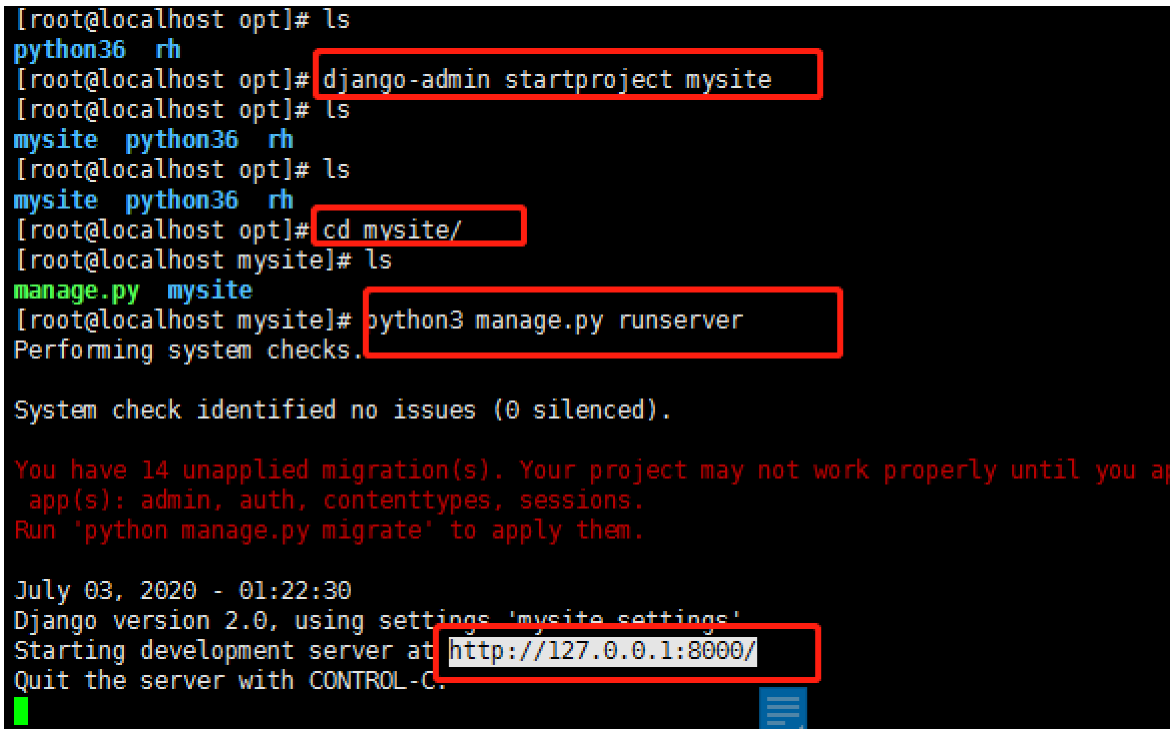


```
[root@localhost luffy]# pip3 list
Package            Version
------------------ --------
appdirs            1.4.4
asgiref            3.2.7
beautifulsoup4     4.9.1
bs4                0.0.1
distlib            0.3.0
Django             2.0
filelock           3.0.12
importlib-metadata 1.6.1
mysqlclient        1.4.6
Pillow             7.1.2
pip                20.1.1
PyMySQL            0.9.3
pytz               2020.1
setuptools         40.6.2
six                1.15.0
soupsieve          2.0.1
sqlparse           0.3.1
uWSGI              2.0.19.1
virtualenv         20.0.23
zipp               3.1.0
```

为了验证django是否安装成功了，我在 /opt 目录下新建一个djang项目：

在root/目录的下的opt目录下新建一个django的项目，然后启动该项目。

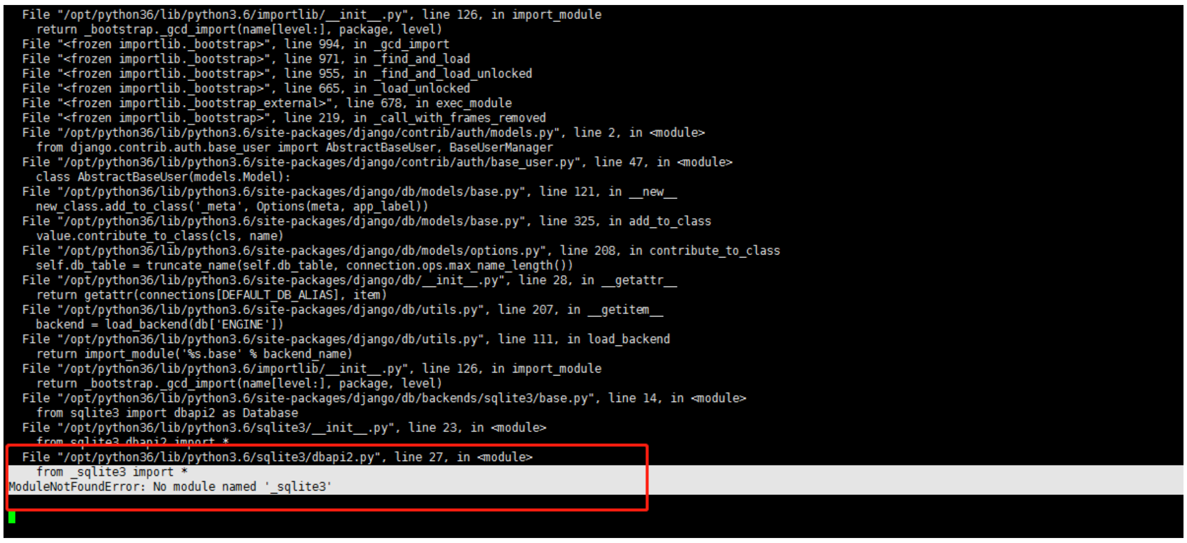
```
[root@tom opt]# django-admin startproject mysite
[root@tom opt]# ls
mysite  python36  rh
[root@tom opt]# cd mysite
[root@tom mysite]# ls
manage.py  mysite
# 启动一定要指定0.0.0.0: 8000 同时要修改settings.py文件的host 和关闭防火墙

关闭防火墙 ： systemctl stop firewalld.service

[root@tom mysite]# python3 manage.py runserver 0.0.0.0:8000
```



```
[root@localhost opt]# ls
python36  rh
[root@localhost opt]# django-admin startproject mysite
[root@localhost opt]# ls
mysite  python36  rh
[root@localhost opt]# ls
mysite  python36  rh
[root@localhost opt]# cd mysite/
[root@localhost mysite]# ls
manage.py  mysite
[root@localhost mysite]# python3 manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you a
 app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

July 03, 2020 - 01:22:30
Django version 2.0, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

启动时还报错：

```
    from _sqlite3 import *
ModuleNotFoundError: No module named '_sqlite3'
```



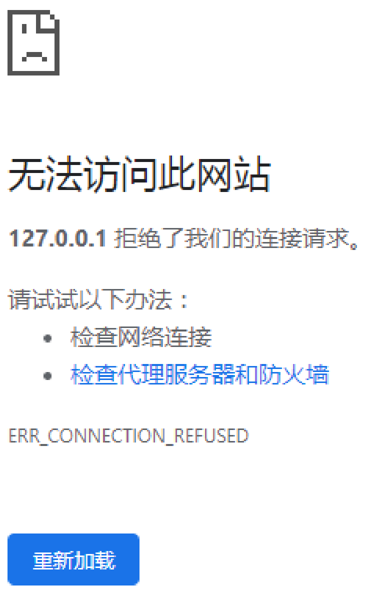
```
  File "/opt/python36/lib/python3.6/importlib/__init__.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 994, in _gcd_import
  File "<frozen importlib._bootstrap>", line 971, in _find_and_load
  File "<frozen importlib._bootstrap>", line 955, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 665, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 678, in exec_module
  File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
  File "/opt/python36/lib/python3.6/site-packages/django/contrib/auth/models.py", line 2, in <module>
    from django.contrib.auth.base_user import AbstractBaseUser, BaseUserManager
  File "/opt/python36/lib/python3.6/site-packages/django/contrib/auth/base_user.py", line 47, in <module>
    class AbstractBaseUser(models.Model):
  File "/opt/python36/lib/python3.6/site-packages/django/db/models/base.py", line 121, in __new__
    new_class.add_to_class('_meta', Options(meta, app_label))
  File "/opt/python36/lib/python3.6/site-packages/django/db/models/base.py", line 325, in add_to_class
    value.contribute_to_class(cls, name)
  File "/opt/python36/lib/python3.6/site-packages/django/db/models/options.py", line 208, in contribute_to_class
    self.db_table = truncate_name(self.db_table, connection.ops.max_name_length())
  File "/opt/python36/lib/python3.6/site-packages/django/db/__init__.py", line 28, in __getattr__
    return getattr(connections[DEFAULT_DB_ALIAS], item)
  File "/opt/python36/lib/python3.6/site-packages/django/db/utils.py", line 207, in __getitem__
    backend = load_backend(db['ENGINE'])
  File "/opt/python36/lib/python3.6/site-packages/django/db/utils.py", line 111, in load_backend
    return import_module('%s.base' % backend_name)
  File "/opt/python36/lib/python3.6/importlib/__init__.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "/opt/python36/lib/python3.6/site-packages/django/db/backends/sqlite3/base.py", line 14, in <module>
    from sqlite3 import dbapi2 as Database
  File "/opt/python36/lib/python3.6/sqlite3/__init__.py", line 23, in <module>
    from sqlite3.dbapi2 import *
  File "/opt/python36/lib/python3.6/sqlite3/dbapi2.py", line 27, in <module>
    from _sqlite3 import *
ModuleNotFoundError: No module named '_sqlite3'
```

出现原因：可能是python版本问题，因为我装了python3，将其设置为默认解释其了

```
我重新默认了解释器为python2 并进行安装  yum install sqlite-devel
```

```
File "/usr/libexec/urlgrabber-ext-down", line 28
https://blog.csdn.net/qq_34272964/article/details/87178724

出现上面 问题是我在安装python3.7版本，让python2和python3共存，没有修改/usr/libexec/urlgrabber-ext-down文件，只需要将#! /usr/bin/python2 修改成 #! /usr/bin/python2.7即可。
```

按照提示的 网站进行用浏览器进行访问，发现访问不到：



无法访问此网站

127.0.0.1 拒绝了我们的连接请求。

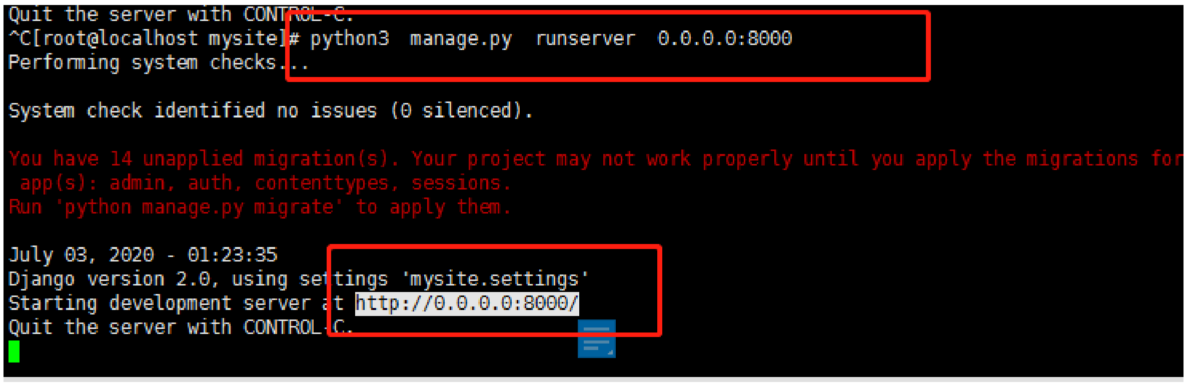
请试试以下办法：

- 检查网络连接
- 检查代理服务器和防火墙

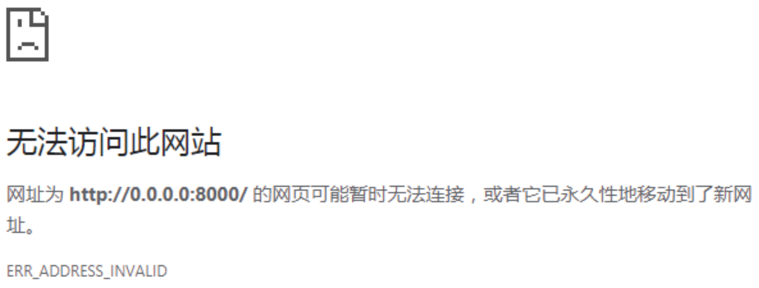
ERR_CONNECTION_REFUSED

于是我们给其指定一个ip

python3 manage.py runserver 0.0.0.0:8000 运行后



```
Quit the server with CONTROL-C.
^C[root@localhost mysite]# python3  manage.py  runserver  0.0.0.0:8000
Performing system checks...

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations for
 app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

July 03, 2020 - 01:23:35
Django version 2.0, using settings 'mysite.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```



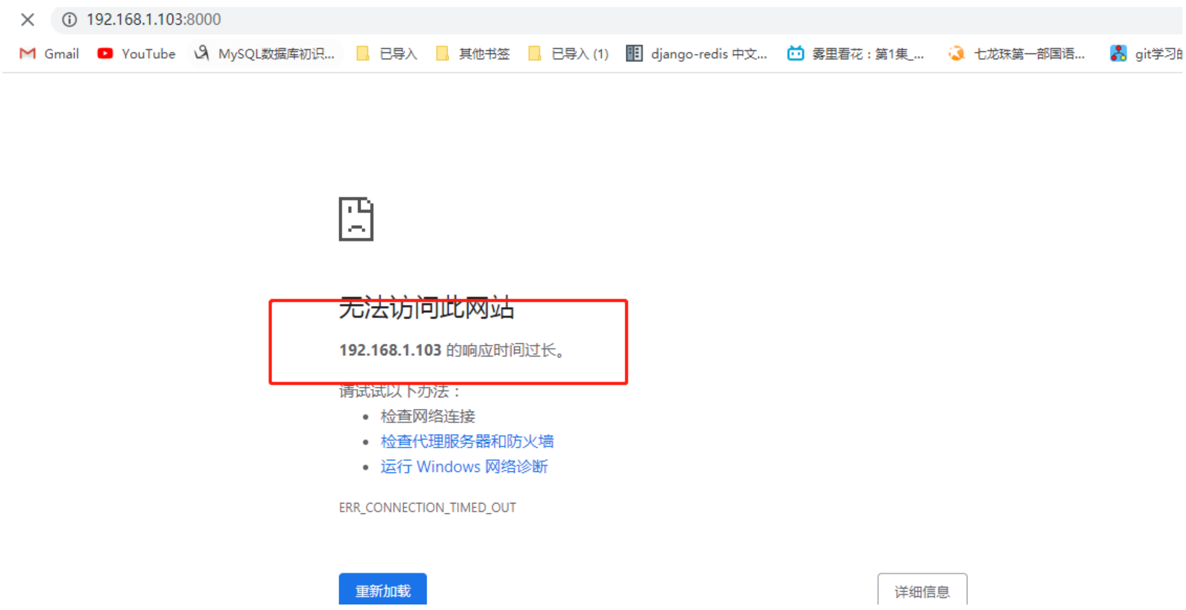
无法访问此网站

网址为 **http://0.0.0.0:8000/** 的网页可能暂时无法连接，或者它已永久性地移动到了新网址。

ERR_ADDRESS_INVALID

然后再访问这个显示的地址 本机的ip加上指定的端口号 http://192.168.130.68:8000/

```
127.0.0.1 本机环回地址
192.168.12.56 本机对外IP地址
0.0.0.0 本机环回地址 127.0.0.1加上192.168.12.56
```



192.168.1.103:8000

无法访问此网站

192.168.1.103 的响应时间过长。

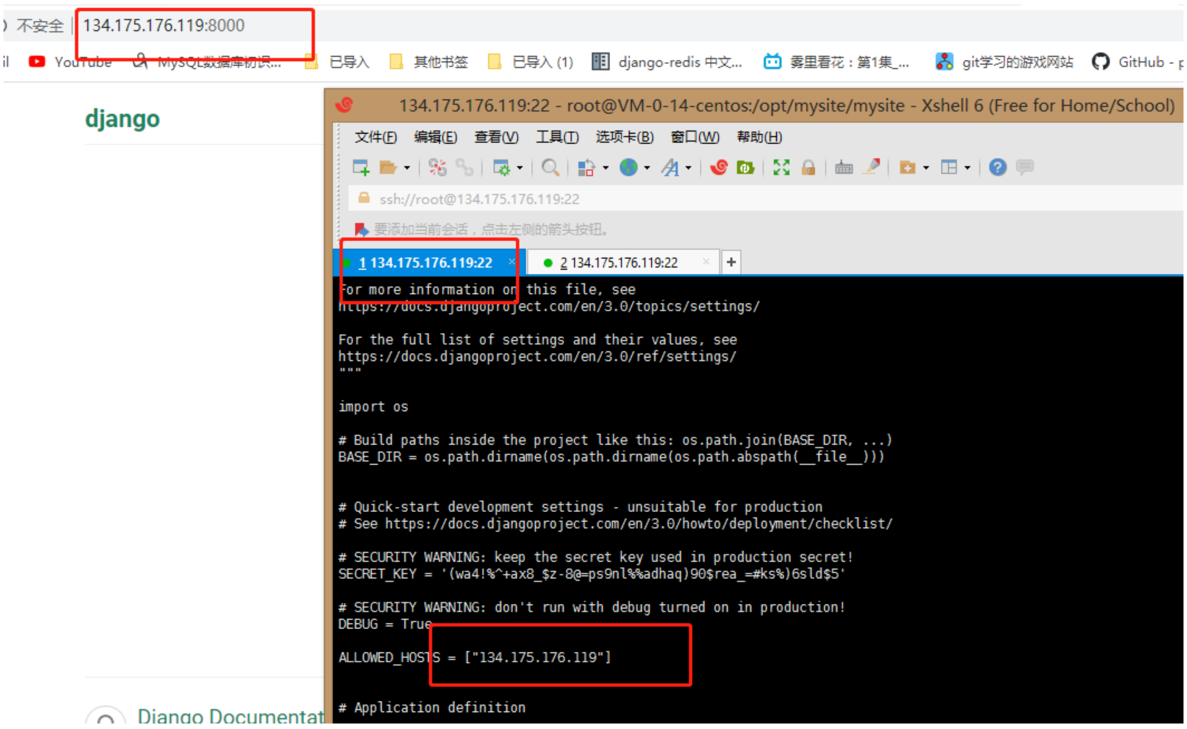
请试试以下办法：

- 检查网络连接
- 检查代理服务器和防火墙
- 运行 Windows 网络诊断

ERR_CONNECTION_TIMED_OUT

发现还是访问不到，出现错误的原因是：Django项目的settings下需要设置host:

如果是在线的腾讯云服务器，则需要些服务器的ip

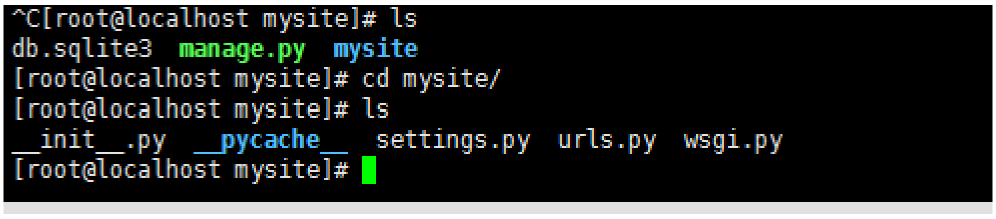


134.175.176.119:8000

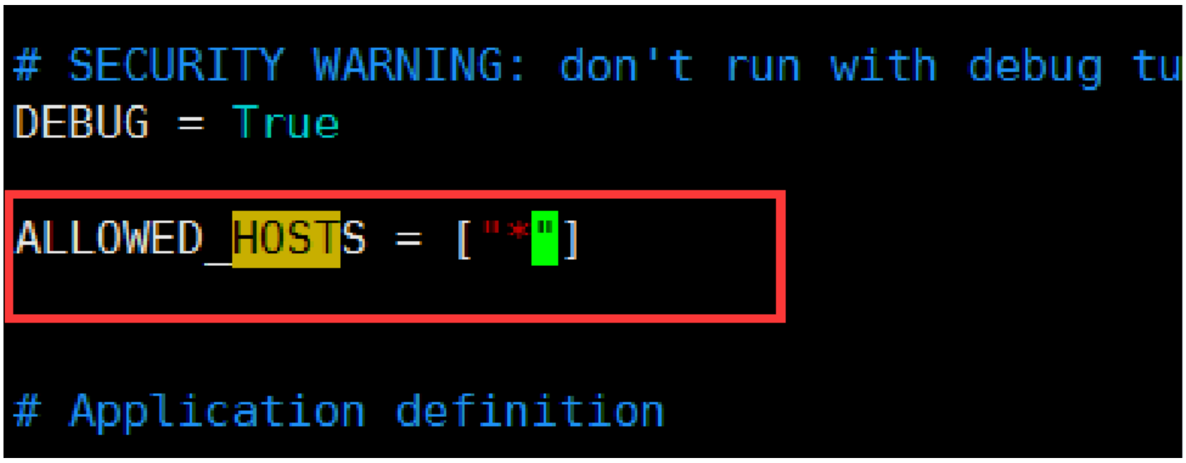
134.175.176.119:22 - root@VM-0-14-centos:/opt/mysite/mysite - Xshell 6 (Free for Home/School)

ssh://root@134.175.176.119:22

```
For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '(wa4!%^+ax8_$z-8@=ps9nl%%adhaq)90$rea_=#ks%)6sld$5'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["134.175.176.119"]

# Application definition
```

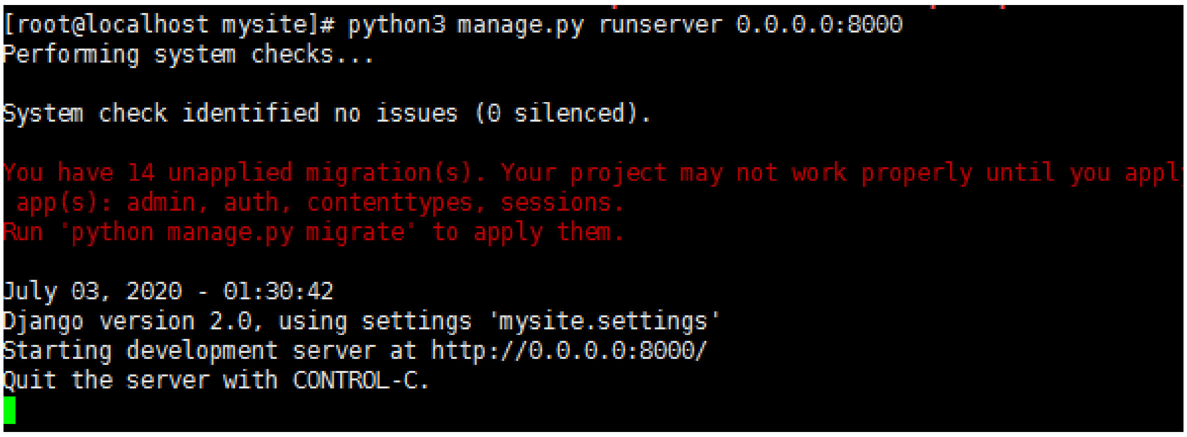


```
^C[root@localhost mysite]# ls
db.sqlite3  manage.py  mysite
[root@localhost mysite]# cd mysite/
[root@localhost mysite]# ls
__init__.py  __pycache__  settings.py  urls.py  wsgi.py
[root@localhost mysite]# 
```



```
# SECURITY WARNING: don't run with debug tu
DEBUG = True

ALLOWED_HOSTS = ["*"]

# Application definition
```

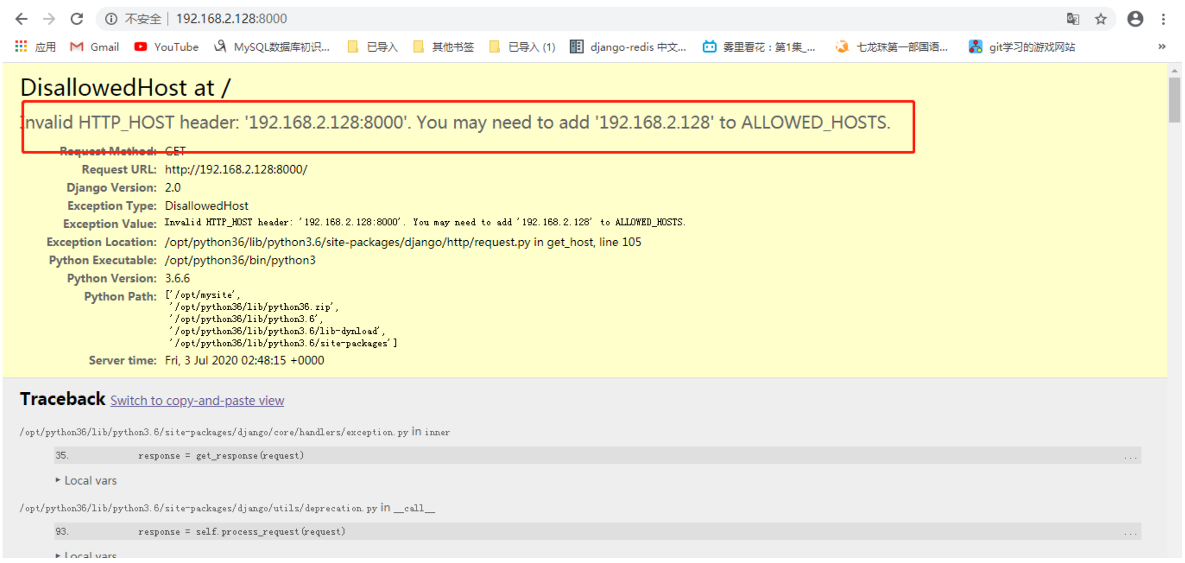
启动：



```
[root@localhost mysite]# python3 manage.py runserver 0.0.0.0:8000
Performing system checks...

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you appl
 app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

July 03, 2020 - 01:30:42
Django version 2.0, using settings 'mysite.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

然后再使用本机地址加上指定端口8000进行访问，发现事实访问不到，此时可能防火墙还未关闭，关闭防火墙，

关闭防火墙后再重启：出现下面的页面，说明host文件忘记添加了



DisallowedHost at /

Invalid HTTP_HOST header: '192.168.2.128:8000'. You may need to add '192.168.2.128' to ALLOWED_HOSTS.

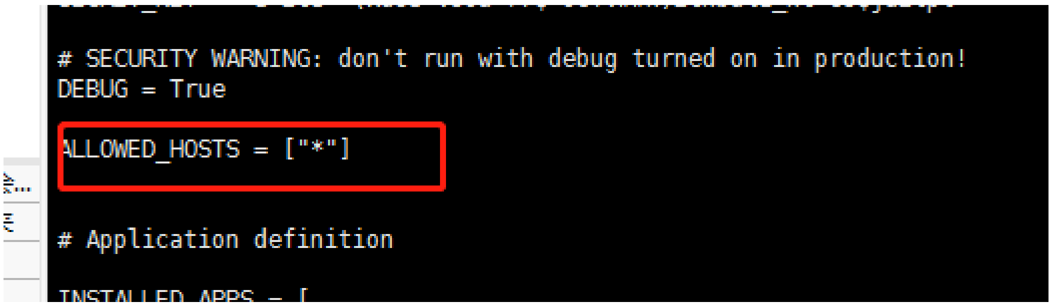
```
   Request Method: GET
      Request URL: http://192.168.2.128:8000/
   Django Version: 2.0
   Exception Type: DisallowedHost
  Exception Value: Invalid HTTP_HOST header: '192.168.2.128:8000'. You may need to add '192.168.2.128' to ALLOWED_HOSTS.
Exception Location: /opt/python36/lib/python3.6/site-packages/django/http/request.py in get_host, line 105
Python Executable: /opt/python36/bin/python3
   Python Version: 3.6.6
      Python Path: ['/opt/mysite',
                    '/opt/python36/lib/python36.zip',
                    '/opt/python36/lib/python3.6',
                    '/opt/python36/lib/python3.6/lib-dynload',
                    '/opt/python36/lib/python3.6/site-packages']
      Server time: Fri, 3 Jul 2020 02:48:15 +0000
```

Traceback Switch to copy-and-paste view

```
/opt/python36/lib/python3.6/site-packages/django/core/handlers/exception.py in inner
    35.             response = get_response(request)                                   ...
  ▸ Local vars
/opt/python36/lib/python3.6/site-packages/django/utils/deprecation.py in __call__
    93.             response = self.process_request(request)                           ...
  ▸ Local vars
```

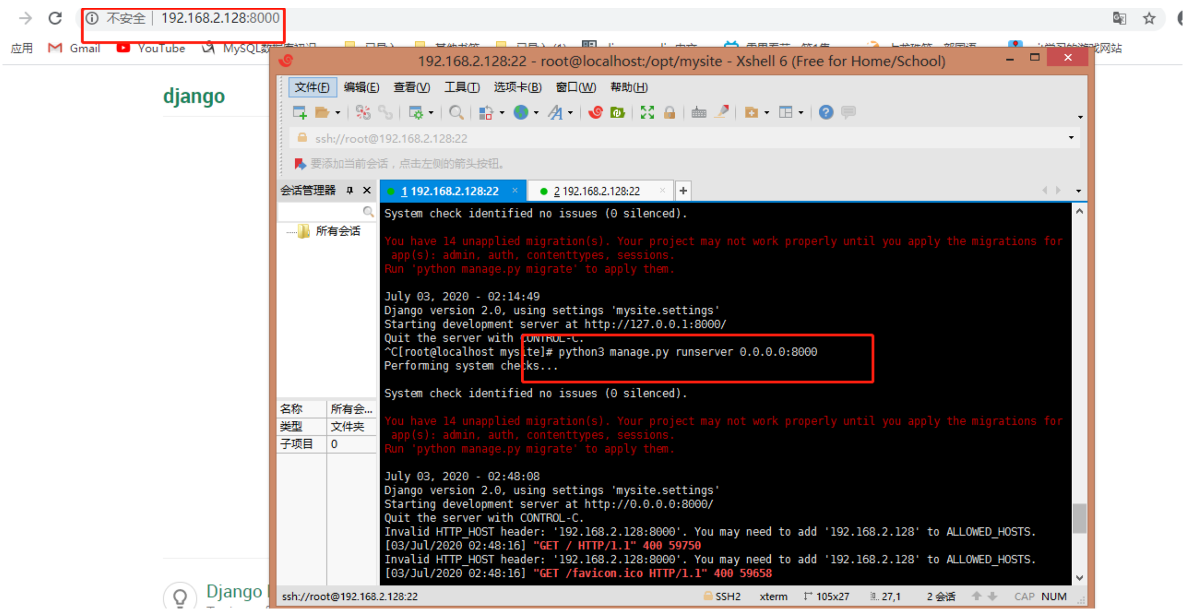
```
之前我已经设置了，为vmware为NAT模式，
然后关闭防火墙：systemctl stop firewalld.service
```

此时再检查setttins.py文件的host



```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["*"]


# Application definition

INSTALLED_APPS = [
```

再次重启运行：可以运行了

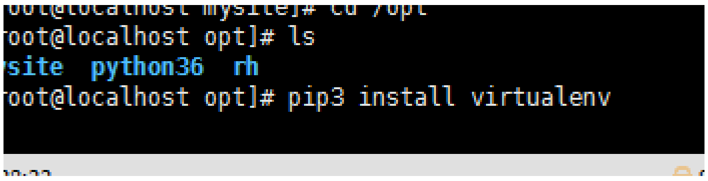


不安全 | 192.168.2.128:8000
django
192.168.2.128:22 - root@localhost:/opt/mysite - Xshell 6 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root@192.168.2.128:22
要添加当前会话，点击左侧的箭头按钮。
会话管理器
1 192.168.2.128:22
2 192.168.2.128:22
所有会话

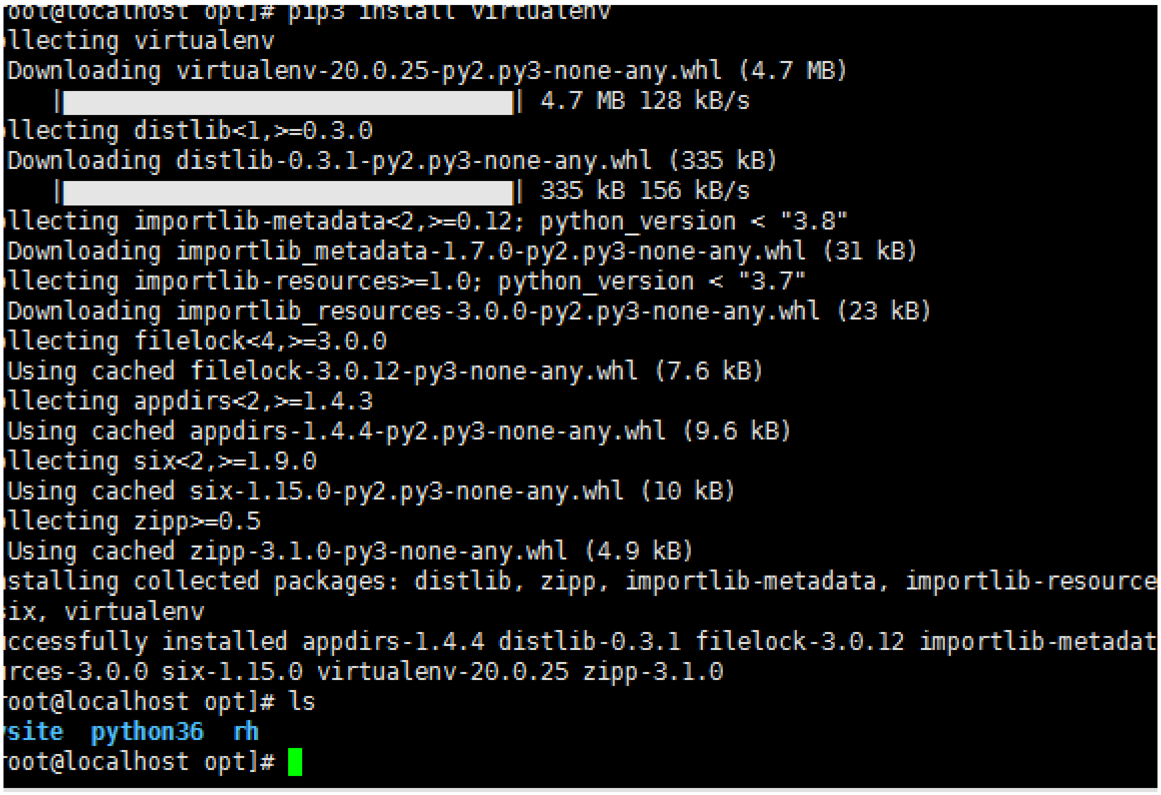
```
System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations for
 app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

July 03, 2020 - 02:14:49
Django version 2.0, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
^C[root@localhost mysite]# python3 manage.py runserver 0.0.0.0:8000
Performing system checks...

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations for
 app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

July 03, 2020 - 02:48:08
Django version 2.0, using settings 'mysite.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
Invalid HTTP_HOST header: '192.168.2.128:8000'. You may need to add '192.168.2.128' to ALLOWED_HOSTS.
[03/Jul/2020 02:48:16] "GET / HTTP/1.1" 400 59750
Invalid HTTP_HOST header: '192.168.2.128:8000'. You may need to add '192.168.2.128' to ALLOWED_HOSTS.
[03/Jul/2020 02:48:16] "GET /favicon.ico HTTP/1.1" 400 59658
```

第五步：安装mkvirtualenv 虚拟环境

```
cd /opt   # opt目录就是存储安装包的
pip3 install virtualenv
pip3 install --upgrade virtualenv==16.7.9 # 指定安装这个版本，不然会报错 package。。
```



```
oot@localhost mysite]# cd /opt
oot@localhost opt]# ls
site  python36  rh
oot@localhost opt]# pip3 install virtualenv
```



```
oot@localhost opt]# pip3 install virtualenv
llecting virtualenv
Downloading virtualenv-20.0.25-py2.py3-none-any.whl (4.7 MB)
    |████████████████████████████████| 4.7 MB 128 kB/s
llecting distlib<1,>=0.3.0
Downloading distlib-0.3.1-py2.py3-none-any.whl (335 kB)
    |████████████████████████████████| 335 kB 156 kB/s
llecting importlib-metadata<2,>=0.12; python_version < "3.8"
Downloading importlib_metadata-1.7.0-py2.py3-none-any.whl (31 kB)
llecting importlib-resources>=1.0; python_version < "3.7"
Downloading importlib_resources-3.0.0-py2.py3-none-any.whl (23 kB)
llecting filelock<4,>=3.0.0
Using cached filelock-3.0.12-py3-none-any.whl (7.6 kB)
llecting appdirs<2,>=1.4.3
Using cached appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
llecting six<2,>=1.9.0
Using cached six-1.15.0-py2.py3-none-any.whl (10 kB)
llecting zipp>=0.5
Using cached zipp-3.1.0-py3-none-any.whl (4.9 kB)
stalling collected packages: distlib, zipp, importlib-metadata, importlib-resource
ix, virtualenv
ccessfully installed appdirs-1.4.4 distlib-0.3.1 filelock-3.0.12 importlib-metadat
rces-3.0.0 six-1.15.0 virtualenv-20.0.25 zipp-3.1.0
oot@localhost opt]# ls
site  python36  rh
oot@localhost opt]# 
```

此时 我在 root目录下创建一个文件夹 Myproject

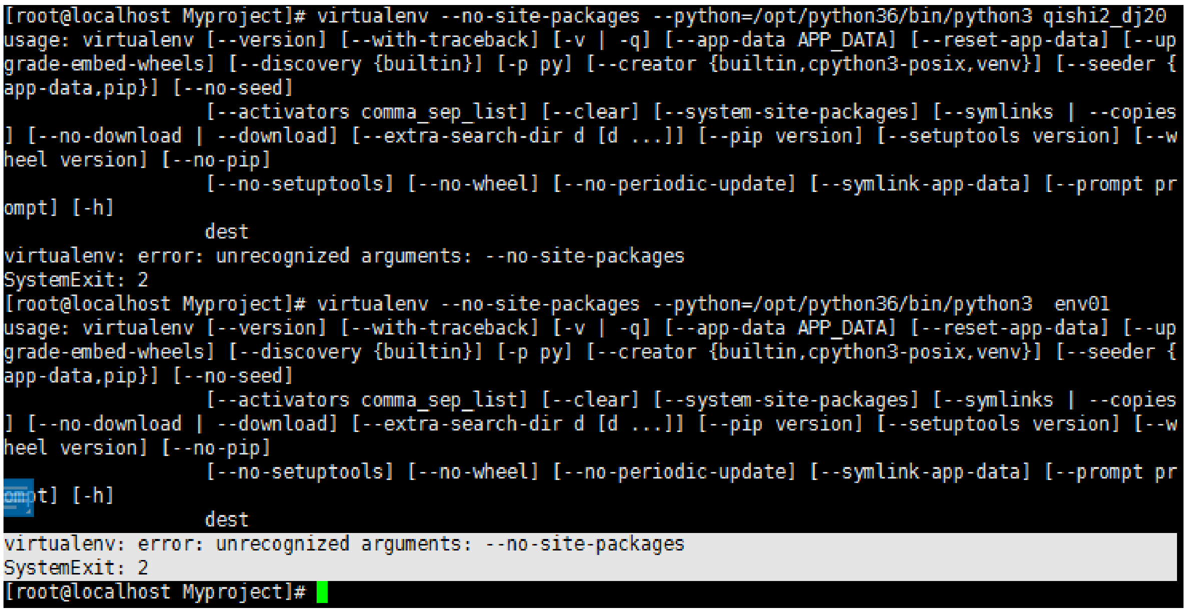
```
cd /
mkdir Myproject
```

然后：

```
virtualenv --no-site-packages --python=/opt/python36/bin/python3  env01  #得到独立第三方包的环境，并且指定解释器是python3
```

报错：

```
virtualenv: error: unrecognized arguments: --no-site-packages
SystemExit: 2
```



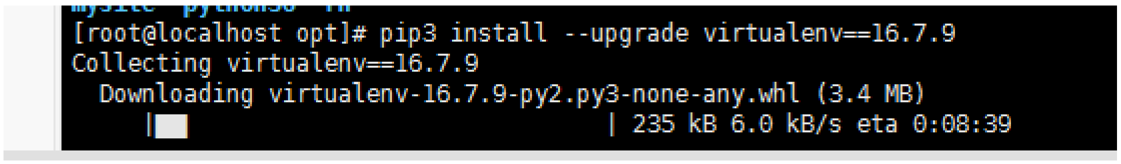
```
[root@localhost Myproject]# virtualenv --no-site-packages --python=/opt/python36/bin/python3 qishi2_dj20
usage: virtualenv [--version] [--with-traceback] [-v | -q] [--app-data APP_DATA] [--reset-app-data] [--up
grade-embed-wheels] [--discovery {builtin}] [-p py] [--creator {builtin,cpython3-posix,venv}] [--seeder {
app-data,pip}] [--no-seed]
                  [--activators comma_sep_list] [--clear] [--system-site-packages] [--symlinks | --copies
] [--no-download | --download] [--extra-search-dir d [d ...]] [--pip version] [--setuptools version] [--w
heel version] [--no-pip]
                  [--no-setuptools] [--no-wheel] [--no-periodic-update] [--symlink-app-data] [--prompt pr
ompt] [-h]
                  dest
virtualenv: error: unrecognized arguments: --no-site-packages
SystemExit: 2
[root@localhost Myproject]# virtualenv --no-site-packages --python=/opt/python36/bin/python3  env01
usage: virtualenv [--version] [--with-traceback] [-v | -q] [--app-data APP_DATA] [--reset-app-data] [--up
grade-embed-wheels] [--discovery {builtin}] [-p py] [--creator {builtin,cpython3-posix,venv}] [--seeder {
app-data,pip}] [--no-seed]
                  [--activators comma_sep_list] [--clear] [--system-site-packages] [--symlinks | --copies
] [--no-download | --download] [--extra-search-dir d [d ...]] [--pip version] [--setuptools version] [--w
heel version] [--no-pip]
                  [--no-setuptools] [--no-wheel] [--no-periodic-update] [--symlink-app-data] [--prompt pr
ompt] [-h]
                  dest
virtualenv: error: unrecognized arguments: --no-site-packages
SystemExit: 2
[root@localhost Myproject]# 
```

原因：其实就死版本问题

```
--no-site-packages   没有这个参数，是virtualenv版本问题
```

安装指定版本：

```
pip3 install --upgrade virtualenv==16.7.9
```



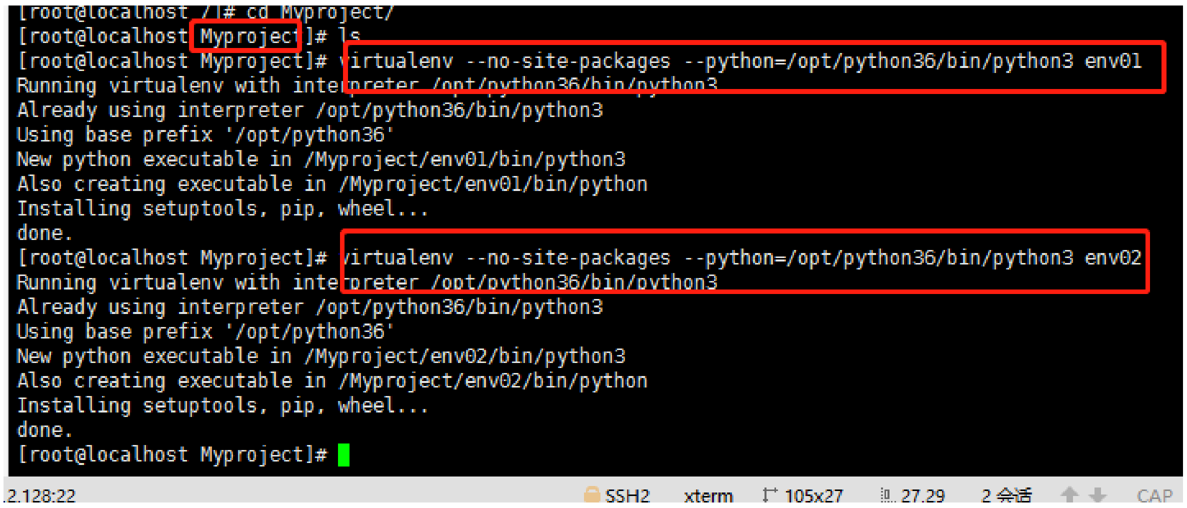
```
[root@localhost opt]# pip3 install --upgrade virtualenv==16.7.9
Collecting virtualenv==16.7.9
  Downloading virtualenv-16.7.9-py2.py3-none-any.whl (3.4 MB)
     |█                               | 235 kB 6.0 kB/s eta 0:08:39
```

然后进行虚拟机创建

```
进入创建的Myproject 目录
cd  Myproject
```

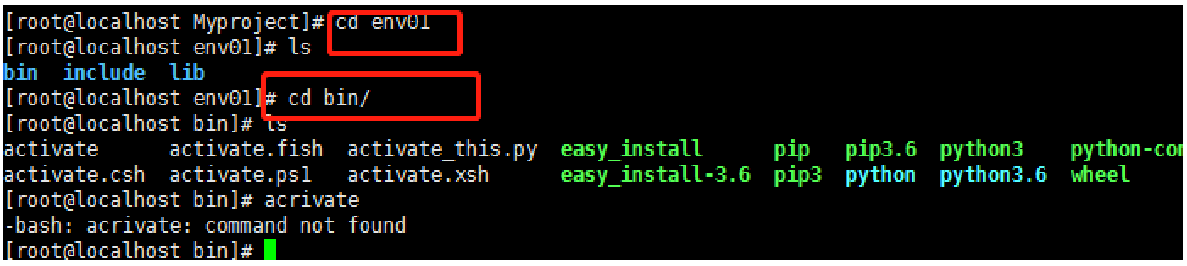
创建虚拟环境：

进入创建的Myproject目录下，可以看到之前创建的两个虚拟环境



```
[root@localhost /]# cd Myproject/
[root@localhost Myproject]# ls
[root@localhost Myproject]# virtualenv --no-site-packages --python=/opt/python36/bin/python3 env01
Running virtualenv with interpreter /opt/python36/bin/python3
Already using interpreter /opt/python36/bin/python3
Using base prefix '/opt/python36'
New python executable in /Myproject/env01/bin/python3
Also creating executable in /Myproject/env01/bin/python
Installing setuptools, pip, wheel...
done.
[root@localhost Myproject]# virtualenv --no-site-packages --python=/opt/python36/bin/python3 env02
Running virtualenv with interpreter /opt/python36/bin/python3
Already using interpreter /opt/python36/bin/python3
Using base prefix '/opt/python36'
New python executable in /Myproject/env02/bin/python3
Also creating executable in /Myproject/env02/bin/python
Installing setuptools, pip, wheel...
done.
[root@localhost Myproject]#
```

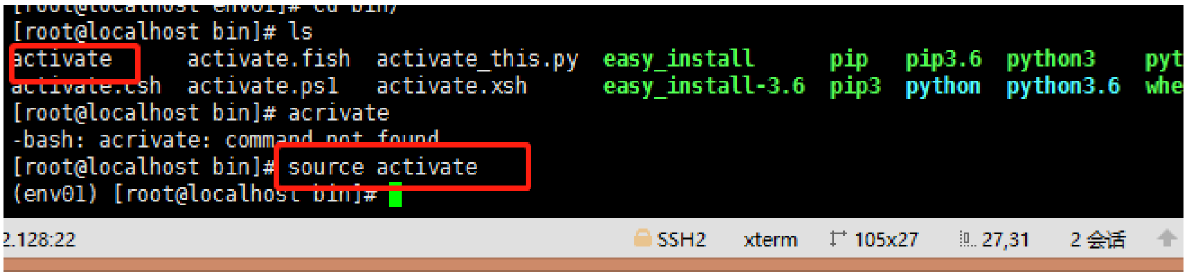
己想进入的那个虚拟环境



```
[root@localhost Myproject]# cd env01
[root@localhost env01]# ls
bin  include  lib
[root@localhost env01]# cd bin/
[root@localhost bin]# ls
activate       activate.fish  activate_this.py  easy_install      pip   pip3.6  python3    python-con
activate.csh   activate.ps1   activate.xsh      easy_install-3.6  pip3  python  python3.6  wheel
[root@localhost bin]# acrivate
-bash: acrivate: command not found
[root@localhost bin]#
```

然后执行source activate 进入虚拟环境，可以看到命令行之前有虚拟环境的名称表示进入虚拟环了。

```
cd /bin
source activate  # 启动这个虚拟环境
```



```
[root@localhost env01]# cd bin/
[root@localhost bin]# ls
activate       activate.fish  activate_this.py  easy_install      pip   pip3.6  python3    pyt
activate.csh   activate.ps1   activate.xsh      easy_install-3.6  pip3  python  python3.6  whe
[root@localhost bin]# acrivate
-bash: acrivate: command not found
[root@localhost bin]# source activate
(env01) [root@localhost bin]#
```

```
这个是是上面的简写：
source venv/bin/activate
```

source venv/bin/activate#此时进入虚拟环境(venv)Myproject

退出venv环境

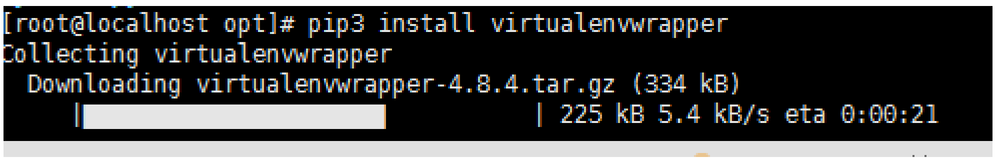
```
deactivate
```

在虚拟环境下安装python包，使用pip3

第六步 安装virtualenvwrapper，用来管理虚拟环境

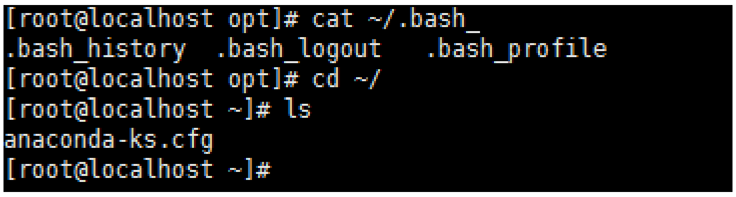
001、安装virtualenvwrapper

```
    pip3 install virtualenvwrapper
```



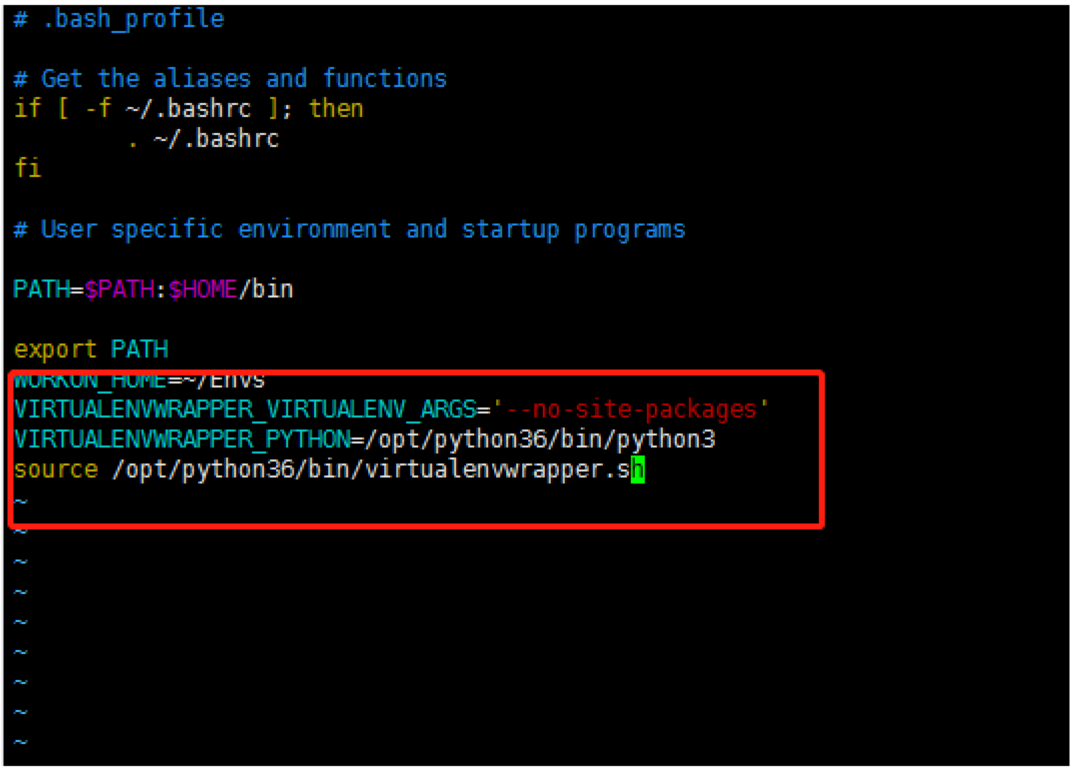
```
[root@localhost opt]# pip3 install virtualenvwrapper
Collecting virtualenvwrapper
  Downloading virtualenvwrapper-4.8.4.tar.gz (334 kB)
     |███████████                    | 225 kB 5.4 kB/s eta 0:00:21
```

002、设置Linux的用户个人配置文件



```
[root@localhost opt]# cat ~/.bash_
.bash_history  .bash_logout   .bash_profile
[root@localhost opt]# cd ~/
[root@localhost ~]# ls
anaconda-ks.cfg
[root@localhost ~]#
```

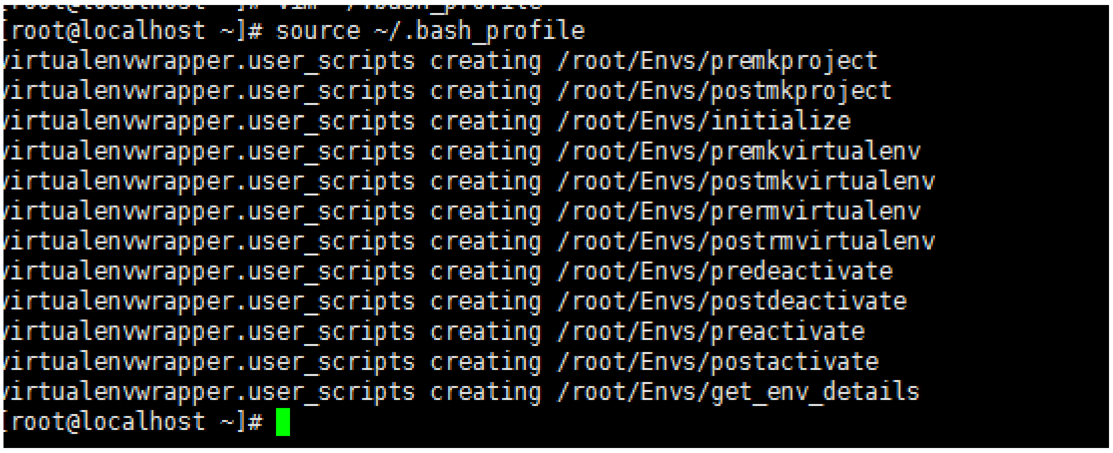
```
vim ~/.bash_profile   # 直接编辑这个文件
添加以下几行内容
WORKON_HOME=~/Envs
VIRTUALENVWRAPPER_VIRTUALENV_ARGS='--no-site-packages'
VIRTUALENVWRAPPER_PYTHON=/opt/python36/bin/python3
source /opt/python36/bin/virtualenvwrapper.sh
```



```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
WORKON_HOME=~/Envs
VIRTUALENVWRAPPER_VIRTUALENV_ARGS='--no-site-packages'
VIRTUALENVWRAPPER_PYTHON=/opt/python36/bin/python3
source /opt/python36/bin/virtualenvwrapper.sh
~
~
~
~
~
~
~
~
~
```

003、重新读取此环境变量文件

```
source ~/.bash_profile
```

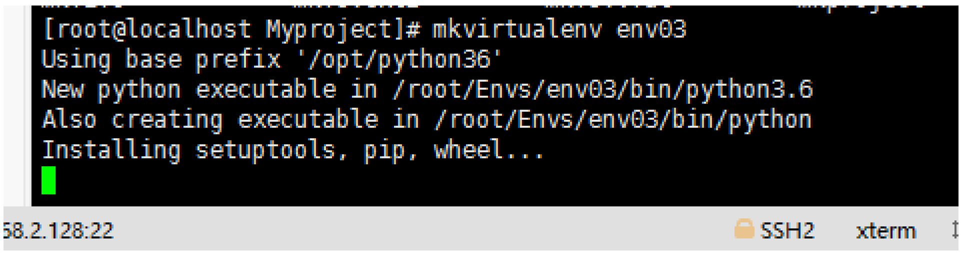


```
[root@localhost ~]# source ~/.bash_profile
virtualenvwrapper.user_scripts creating /root/Envs/premkproject
virtualenvwrapper.user_scripts creating /root/Envs/postmkproject
virtualenvwrapper.user_scripts creating /root/Envs/initialize
virtualenvwrapper.user_scripts creating /root/Envs/premkvirtualenv
virtualenvwrapper.user_scripts creating /root/Envs/postmkvirtualenv
virtualenvwrapper.user_scripts creating /root/Envs/prermvirtualenv
virtualenvwrapper.user_scripts creating /root/Envs/postrmvirtualenv
virtualenvwrapper.user_scripts creating /root/Envs/predeactivate
virtualenvwrapper.user_scripts creating /root/Envs/postdeactivate
virtualenvwrapper.user_scripts creating /root/Envs/preactivate
virtualenvwrapper.user_scripts creating /root/Envs/postactivate
virtualenvwrapper.user_scripts creating /root/Envs/get_env_details
[root@localhost ~]#
```

004、virtualenvwrapper的基本使用：

```
1. 创建一个虚拟环境：
$ mkvirtualenv my_django115
这会在 ~/Envs 中创建 my_django115 文件夹。

2. 在虚拟环境上工作：激活虚拟环境my_django115
$ workon my_django115

3. 再创建一个新的虚拟环境
$ mkvirtualenv my_django2

virtualenvwrapper 提供环境名字的tab补全功能。
当有很多环境， 并且很难记住它们的名字时，这就显得很有用。

4. workon还可以任意停止你当前的环境，可以在多个虚拟环境中来回切换
workon django1.15
workon django2.0
```

```
5. 也可以手动停止虚拟环境
deactivate

6. 删除虚拟环境，需要先退出虚拟环境
rmvirtualenv my_django115
```



```
[root@localhost Myproject]# mkvirtualenv env03
Using base prefix '/opt/python36'
New python executable in /root/Envs/env03/bin/python3.6
Also creating executable in /root/Envs/env03/bin/python
Installing setuptools, pip, wheel...
```

第七步：安装mariadb(mysql)

```
https://www.cnblogs.com/tiger666/articles/10259269.html
```

001、使用官方源安装mariadb

```
教程:
https://www.cnblogs.com/tiger666/articles/10259269.html
https://www.cnblogs.com/yhongji/p/9783065.html
```

002、 使用官方源安装mariadb，首先在 RHEL/CentOS 和 Fedora 操作系统中添加 MariaDB 的 YUM 配置文件 MariaDB.repo 文件。

刚开始时没有的，需要添加

```
vim /etc/yum.repos.d/MariaDB.repo
添加repo仓库配置内容
[mariadb]
name=MariaDB
baseurl=http://yum.mariadb.org/10.1/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

003、 当 MariaDB 仓库地址添加好后，一条命令安装官方的最新版本mariadb数据库

```
yum install MariaDB-server MariaDB-client -y
```

004、配置MariaDB

```
[root@mini ~]# systemctl start mariadb  # 开启服务
[root@mini ~]# systemctl enable mariadb  # 设置为开机自启动服务
```

005、首次安装需要进行数据库的配置，命令都和mysql的一样

```
mysql_secure_installation
```

006、配置时出现的各个选项

```
# 第一次进入还没有设置密码则直接回车
Enter current password for root (enter for none):  # 输入数据库超级管理员root的密码(注意不是系统root的密码)，第一次进入还没有设置密码则直接回车

Set root password? [Y/n]  # 设置密码，y

New password:  # 新密码
Re-enter new password:  # 再次输入密码

Remove anonymous users? [Y/n]  # 移除匿名用户， y

Disallow root login remotely? [Y/n]  # 拒绝root远程登录，n，不管y/n，都会拒绝root远程登录

Remove test database and access to it? [Y/n]  # 删除test数据库，y：删除。n：不删除，数据库中会有一个test数据库，一般不需要

Reload privilege tables now? [Y/n]  # 重新加载权限表，y。或者重启服务也许
```

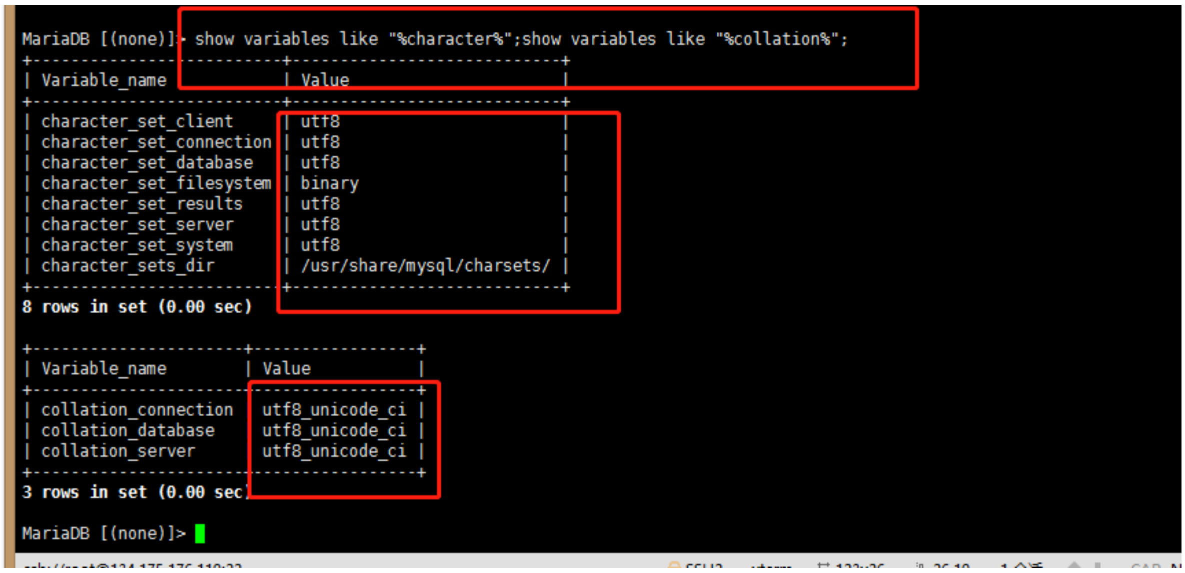
测试是否能够登录成功，出现 MariaDB [(none)]> 就表示已经能够正常登录使用MariaDB数据库了

```
[root@mini ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

007、设置MariaDB字符集为utf-8

```
一、修改配置文件/etc/my.cnf

vi /etc/my.cnf

在[mysqld]标签下添加

#character
init_connect='SET collation_connection = utf8_unicode_ci'
init_connect='SET NAMES utf8'
character-set-server=utf8
collation-server=utf8_unicode_ci
skip-character-set-client-handshake

二、配置文件/etc/my.cnf.d/client.cnf
vi /etc/my.cnf.d/client.cnf

在[client]下添加

#character
default-character-set=utf8
```

```
三、配置文件/etc/my.cnf.d/mysql-clients.cnf

 vi /etc/my.cnf.d/mysql-clients.cnf

在[mysql]中添加

#character

default-character-set=utf8


四、重启MariaDB服务

service mariadb restart


五、查看字符集
show variables like "%character%";show variables like "%collation%";
```

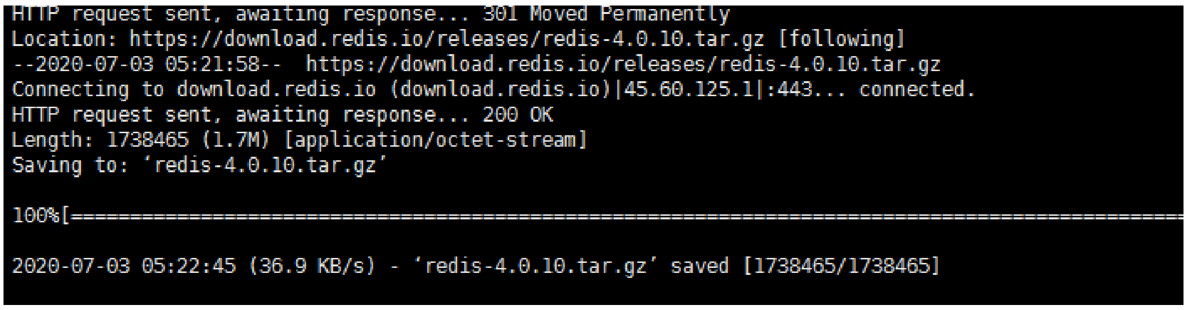


```
MariaDB [(none)]> show variables like "%character%";show variables like "%collation%";
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | utf8                       |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | utf8                       |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.00 sec)

+----------------------+-----------------+
| Variable_name        | Value           |
+----------------------+-----------------+
| collation_connection | utf8_unicode_ci |
| collation_database   | utf8_unicode_ci |
| collation_server     | utf8_unicode_ci |
+----------------------+-----------------+
3 rows in set (0.00 sec)

MariaDB [(none)]>
```

第八步：安装redis

1.下载redis源码

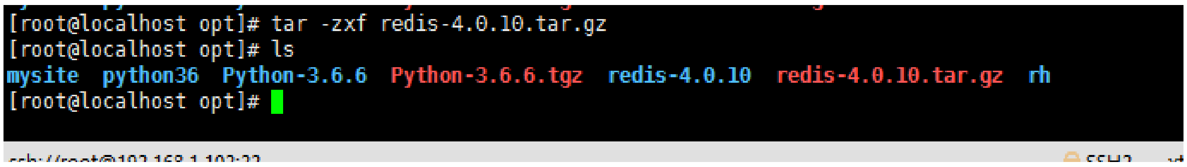
```
wget http://download.redis.io/releases/redis-4.0.10.tar.gz
```



```
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://download.redis.io/releases/redis-4.0.10.tar.gz [following]
--2020-07-03 05:21:58--  https://download.redis.io/releases/redis-4.0.10.tar.gz
Connecting to download.redis.io (download.redis.io)|45.60.125.1|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1738465 (1.7M) [application/octet-stream]
Saving to: 'redis-4.0.10.tar.gz'

100%[==========================================================================================

2020-07-03 05:22:45 (36.9 KB/s) - 'redis-4.0.10.tar.gz' saved [1738465/1738465]
```

2.解压缩

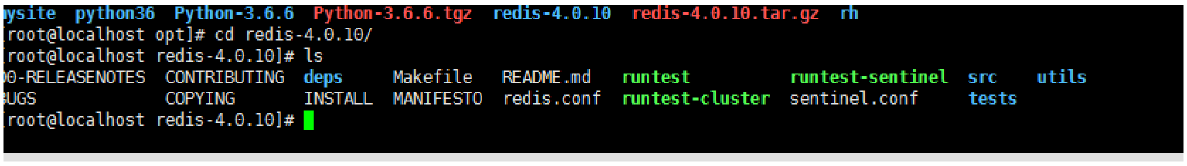
```
tar -zxf redis-4.0.10.tar.gz
```



```
[root@localhost opt]# tar -zxf redis-4.0.10.tar.gz
[root@localhost opt]# ls
mysite  python36  Python-3.6.6  Python-3.6.6.tgz  redis-4.0.10  redis-4.0.10.tar.gz  rh
[root@localhost opt]# 
```

3.切换redis源码目录

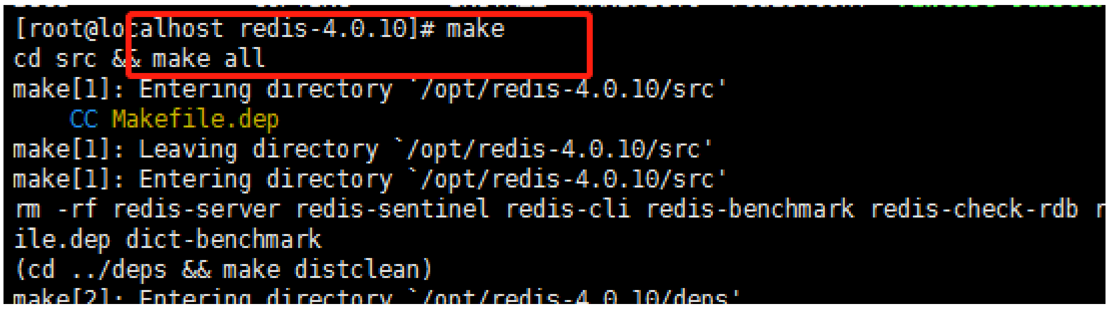
```
cd redis-4.0.10
```



```
ysite  python36  Python-3.6.6  Python-3.6.6.tgz  redis-4.0.10  redis-4.0.10.tar.gz  rh
root@localhost opt]# cd redis-4.0.10/
root@localhost redis-4.0.10]# ls
0-RELEASENOTES  CONTRIBUTING  deps     Makefile   README.md   runtest          runtest-sentinel  src    utils
UGS             COPYING       INSTALL  MANIFESTO  redis.conf  runtest-cluster  sentinel.conf     tests
root@localhost redis-4.0.10]# 
```

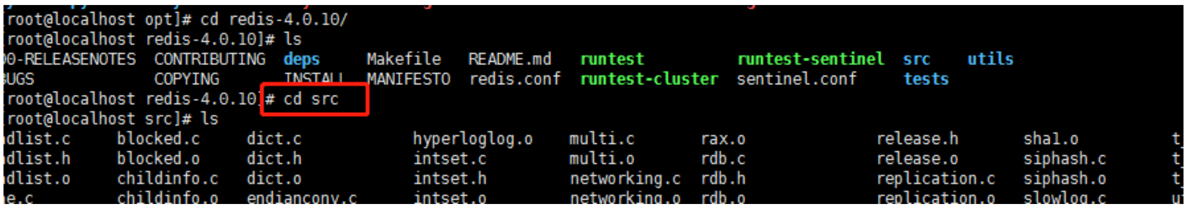
4.编译源文件

```
make
```



```
[root@localhost redis-4.0.10]# make
cd src && make all
make[1]: Entering directory `/opt/redis-4.0.10/src'
    CC Makefile.dep
make[1]: Leaving directory `/opt/redis-4.0.10/src'
make[1]: Entering directory `/opt/redis-4.0.10/src'
rm -rf redis-server redis-sentinel redis-cli redis-benchmark redis-check-rdb r
ile.dep dict-benchmark
(cd ../deps && make distclean)
make[2]: Entering directory `/opt/redis-4.0.10/deps'
```

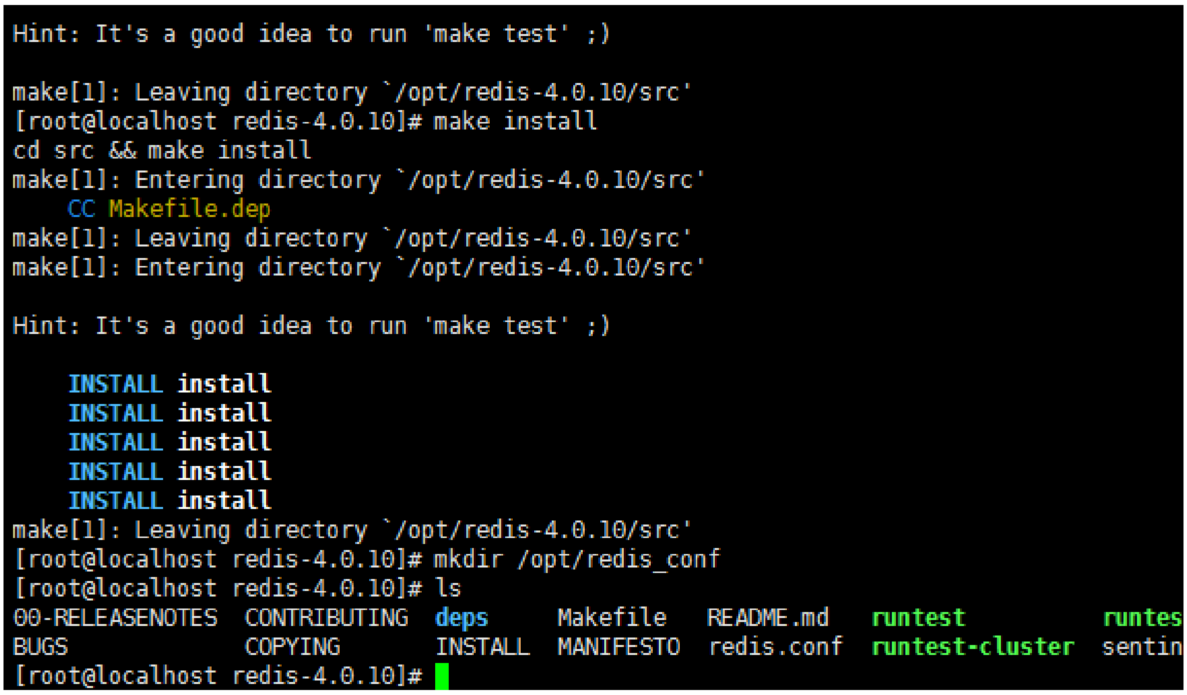
5.编译好后，src/目录下有编译好的redis指令



```
root@localhost opt]# cd redis-4.0.10/
root@localhost redis-4.0.10]# ls
0-RELEASENOTES  CONTRIBUTING  deps     Makefile   README.md   runtest          runtest-sentinel  src    utils
UGS             COPYING       INSTALL  MANIFESTO  redis.conf  runtest-cluster  sentinel.conf     tests
root@localhost redis-4.0.10]# cd src
root@localhost src]# ls
dlist.c     blocked.c     dict.c          hyperloglog.o   multi.c         rax.o         release.h       shal.o        t
dlist.h     blocked.o     dict.h          intset.c        multi.o         rdb.c         release.o       siphash.c     t
dlist.o     childinfo.c   dict.o          intset.h        networking.c    rdb.h         replication.c   siphash.o     t
e.c         childinfo.o   endianconv.c    intset.o        networking.o    rdb.o         replication.o   slowlog.c     u
```

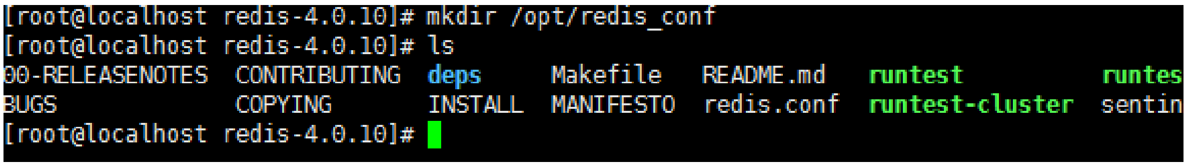
6.make install 安装到指定目录，默认在/usr/local/bin

```
make install prefix=/opt/redis4.0
```



```
Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory `/opt/redis-4.0.10/src'
[root@localhost redis-4.0.10]# make install
cd src && make install
make[1]: Entering directory `/opt/redis-4.0.10/src'
    CC Makefile.dep
make[1]: Leaving directory `/opt/redis-4.0.10/src'
make[1]: Entering directory `/opt/redis-4.0.10/src'

Hint: It's a good idea to run 'make test' ;)

    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
make[1]: Leaving directory `/opt/redis-4.0.10/src'
[root@localhost redis-4.0.10]# mkdir /opt/redis_conf
[root@localhost redis-4.0.10]# ls
00-RELEASENOTES  CONTRIBUTING  deps     Makefile   README.md   runtest          runtes
BUGS             COPYING       INSTALL  MANIFESTO  redis.conf  runtest-cluster  sentin
[root@localhost redis-4.0.10]#
```

7 创建一个专门存放redis配置文件目录



```
[root@localhost redis-4.0.10]# mkdir /opt/redis_conf
[root@localhost redis-4.0.10]# ls
00-RELEASENOTES  CONTRIBUTING  deps     Makefile   README.md   runtest          runtes
BUGS             COPYING       INSTALL  MANIFESTO  redis.conf  runtest-cluster  sentin
[root@localhost redis-4.0.10]#
```

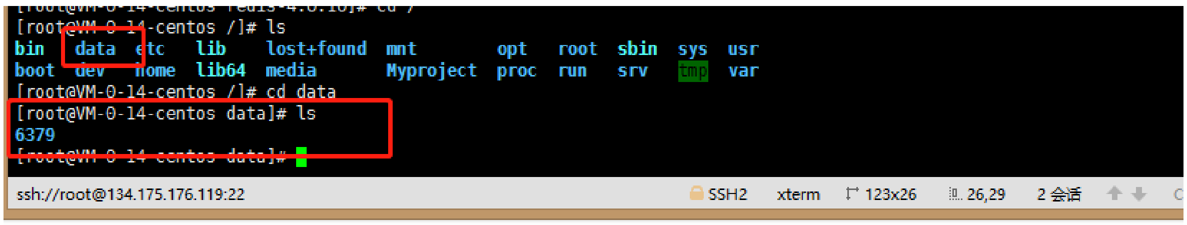
8、在/opt/redis_conf/ 下配置完上诉配置文件 vim redis-6379.conf

启动时会提示没有 /data/6379这个文件，需要手动创建，后续 在任何目录下，只要执行这个路径，redis就可以被启动了

```
mkdir /opt/redis_conf
vim redis-6379.conf
写入以下配置
port 6379
daemonize yes
pidfile /data/6379/redis.pid   #/data/6379  这个文件需要创建
loglevel notice
logfile "/data/6379/redis.log"
protected-mode yes
```

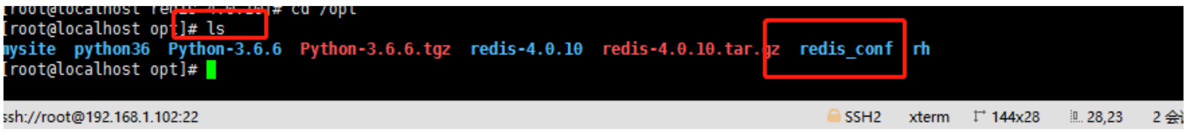
在root目录创建data目录

```
mkdir -p /data/6379
```



```
[root@VM-0-14-centos /]# ls
bin   data  etc   lib    lost+found  mnt        opt   root  sbin  sys  usr
boot  dev   home  lib64  media       Myproject  proc  run   srv   tmp  var
[root@VM-0-14-centos /]# cd data
[root@VM-0-14-centos data]# ls
6379
[root@VM-0-14-centos data]#
```

ssh://root@134.175.176.119:22   SSH2   xterm   123x26   26,29   2 会话



```
[root@localhost opt]# ls
ysite  python36  Python-3.6.6  Python-3.6.6.tgz  redis-4.0.10  redis-4.0.10.tar.gz  redis_conf  rh
[root@localhost opt]#
```

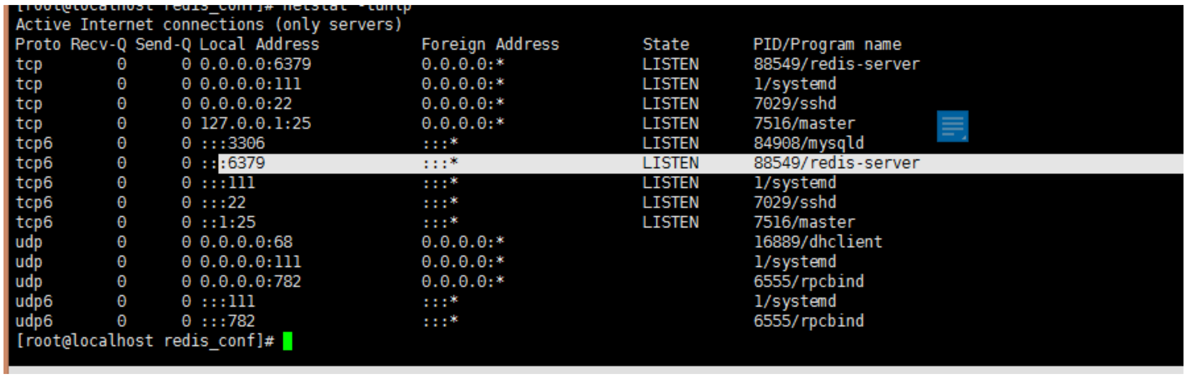
ssh://root@192.168.1.102:22   SSH2   xterm   144x28   28,23   2 会

然后再执行 redis-server /opt/redis_conf/redis-6379.conf 将这个路径指定为启动redis的目录。

```
redis-server /opt/redis_conf/redis-6379.conf
```

后续 在任何目录下，只要执行这个路径，redis就可以被启动了

```
netstat -tunlp 查看端口号码是否启动。
```



```
[root@localhost redis_conf]# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:6379            0.0.0.0:*               LISTEN      88549/redis-server
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      7029/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      7516/master
tcp6       0      0 :::3306                 :::*                    LISTEN      84908/mysqld
tcp6       0      0 :::6379                 :::*                    LISTEN      88549/redis-server
tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
tcp6       0      0 :::22                   :::*                    LISTEN      7029/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      7516/master
udp        0      0 0.0.0.0:68              0.0.0.0:*                           16889/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1/systemd
udp        0      0 0.0.0.0:782             0.0.0.0:*                           6555/rpcbind
udp6       0      0 :::111                  :::*                                1/systemd
udp6       0      0 :::782                  :::*                                6555/rpcbind
[root@localhost redis_conf]#
```

9、redis配置文件详解

```
port 6379                          # 运行在6379的redis数据库实例
daemonize yes                      # 后台运行redis
pidfile /data/6379/redis.pid       # 存放redis pid的文件
loglevel notice                    # 日志等级
logfile "/data/6379/redis.log"     # 指定redis日志文件的生成目录
dir /data/6379                     # 指定redis数据文件夹的目录
protected-mode yes                 # 安全模式
requirepass   haohaio              # 设置redis的密码
```

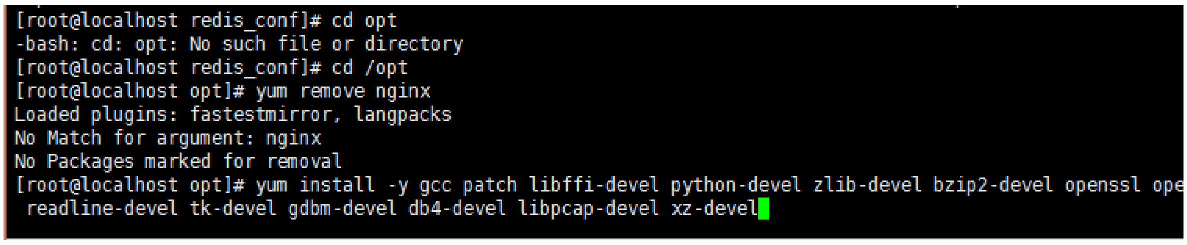
第九步：nginx编译安装

001、卸载nginx

```
yum remove nginx
```

002、 安装所需要的依赖库

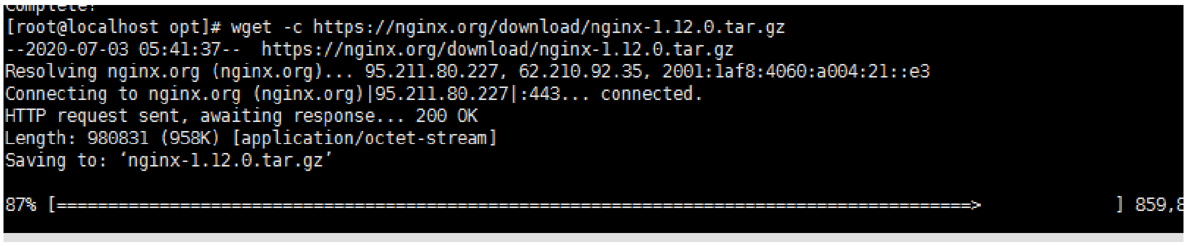
```
yum install -y gcc patch libffi-devel python-devel zlib-devel bzip2-devel
openssl openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gdbm-
devel db4-devel libpcap-devel xz-devel
```



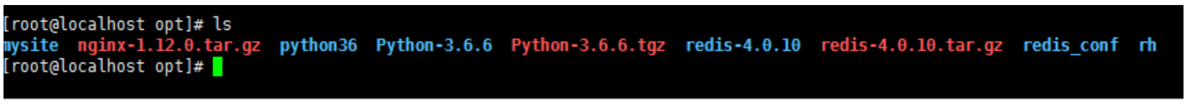
```
[root@localhost redis_conf]# cd opt
-bash: cd: opt: No such file or directory
[root@localhost redis_conf]# cd /opt
[root@localhost opt]# yum remove nginx
Loaded plugins: fastestmirror, langpacks
No Match for argument: nginx
No Packages marked for removal
[root@localhost opt]# yum install -y gcc patch libffi-devel python-devel zlib-devel bzip2-devel openssl ope
 readline-devel tk-devel gdbm-devel db4-devel libpcap-devel xz-devel
```

003 下载nginx安装源码包

wget -c https://nginx.org/download/nginx-1.12.0.tar.gz



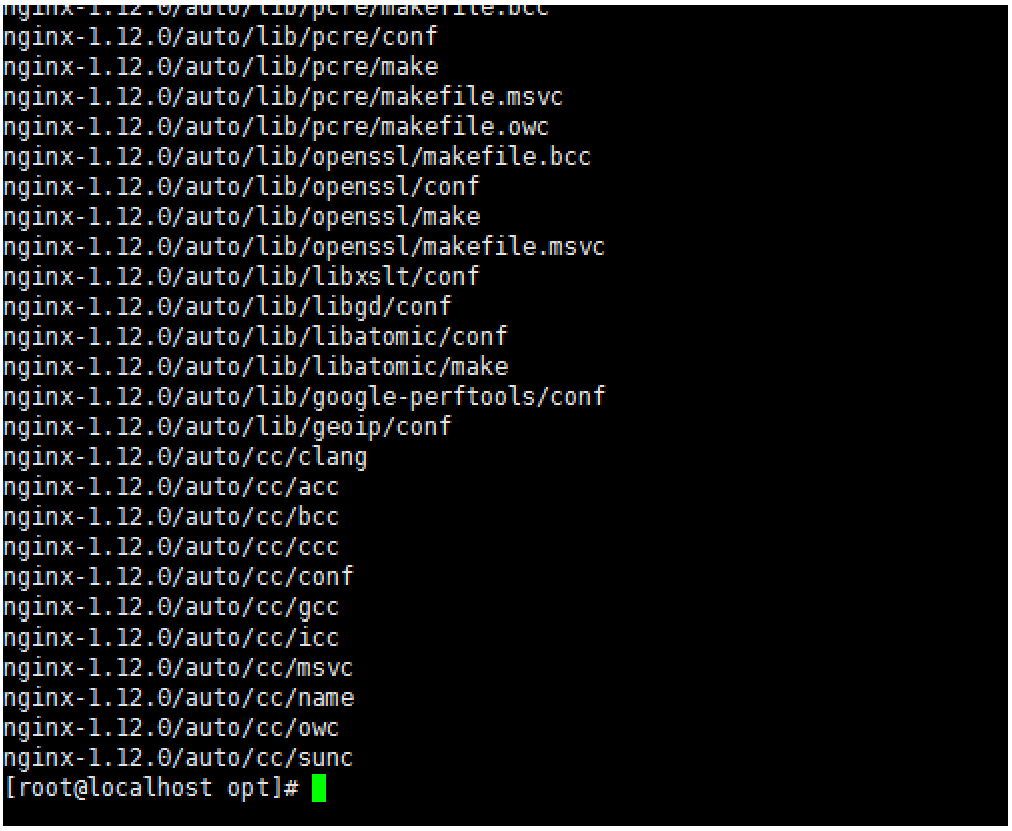
```
[root@localhost opt]# wget -c https://nginx.org/download/nginx-1.12.0.tar.gz
--2020-07-03 05:41:37--  https://nginx.org/download/nginx-1.12.0.tar.gz
Resolving nginx.org (nginx.org)... 95.211.80.227, 62.210.92.35, 2001:1af8:4060:a004:21::e3
Connecting to nginx.org (nginx.org)|95.211.80.227|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 980831 (958K) [application/octet-stream]
Saving to: ‘nginx-1.12.0.tar.gz’

87% [=========================================================================================>             ] 859,8
```



```
[root@localhost opt]# ls
mysite  nginx-1.12.0.tar.gz  python36  Python-3.6.6  Python-3.6.6.tgz  redis-4.0.10  redis-4.0.10.tar.gz  redis_conf  rh
[root@localhost opt]#
```

004.解压缩源码

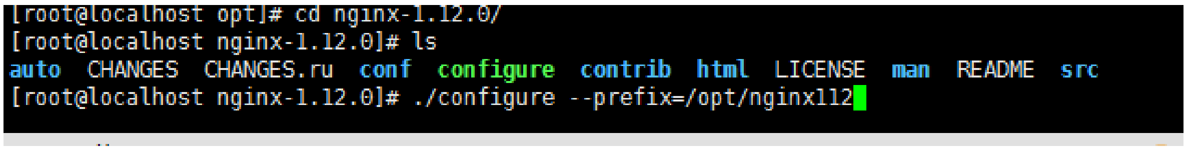
```
tar -zxvf nginx-1.12.0.tar.gz
```



```
nginx-1.12.0/auto/lib/pcre/conf
nginx-1.12.0/auto/lib/pcre/make
nginx-1.12.0/auto/lib/pcre/makefile.msvc
nginx-1.12.0/auto/lib/pcre/makefile.owc
nginx-1.12.0/auto/lib/openssl/makefile.bcc
nginx-1.12.0/auto/lib/openssl/conf
nginx-1.12.0/auto/lib/openssl/make
nginx-1.12.0/auto/lib/openssl/makefile.msvc
nginx-1.12.0/auto/lib/libxslt/conf
nginx-1.12.0/auto/lib/libgd/conf
nginx-1.12.0/auto/lib/libatomic/conf
nginx-1.12.0/auto/lib/libatomic/make
nginx-1.12.0/auto/lib/google-perftools/conf
nginx-1.12.0/auto/lib/geoip/conf
nginx-1.12.0/auto/cc/clang
nginx-1.12.0/auto/cc/acc
nginx-1.12.0/auto/cc/bcc
nginx-1.12.0/auto/cc/ccc
nginx-1.12.0/auto/cc/conf
nginx-1.12.0/auto/cc/gcc
nginx-1.12.0/auto/cc/icc
nginx-1.12.0/auto/cc/msvc
nginx-1.12.0/auto/cc/name
nginx-1.12.0/auto/cc/owc
nginx-1.12.0/auto/cc/sunc
[root@localhost opt]#
```

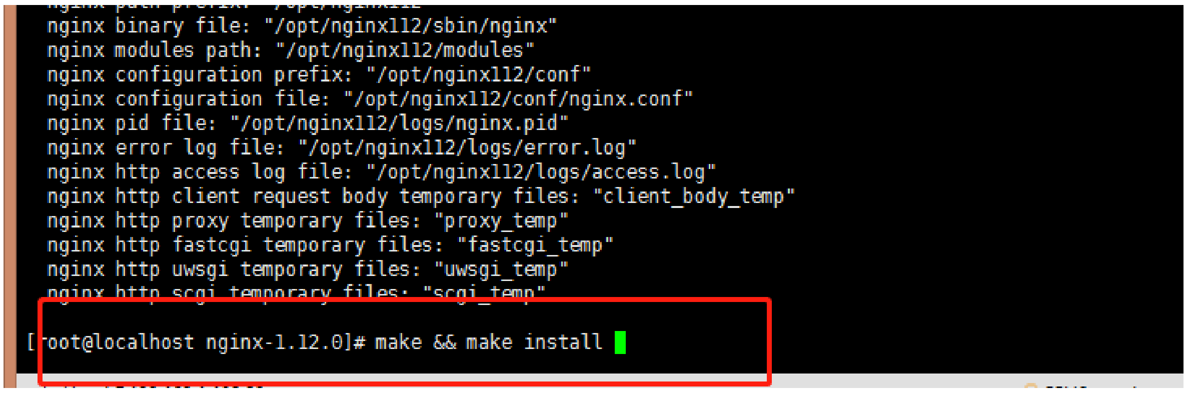
005.配置，编译安装

```
cd nginx-1.12.0/

./configure --prefix=/opt/nginx112
```



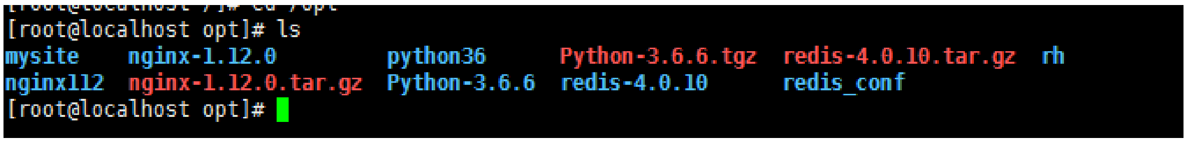
```
[root@localhost opt]# cd nginx-1.12.0/
[root@localhost nginx-1.12.0]# ls
auto  CHANGES  CHANGES.ru  conf  configure  contrib  html  LICENSE  man  README  src
[root@localhost nginx-1.12.0]# ./configure --prefix=/opt/nginx112
```

```
make && make install
```



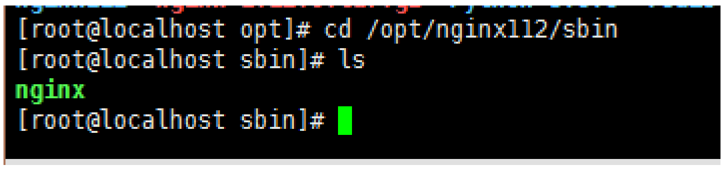
```
  nginx binary file: "/opt/nginx112/sbin/nginx"
  nginx modules path: "/opt/nginx112/modules"
  nginx configuration prefix: "/opt/nginx112/conf"
  nginx configuration file: "/opt/nginx112/conf/nginx.conf"
  nginx pid file: "/opt/nginx112/logs/nginx.pid"
  nginx error log file: "/opt/nginx112/logs/error.log"
  nginx http access log file: "/opt/nginx112/logs/access.log"
  nginx http client request body temporary files: "client_body_temp"
  nginx http proxy temporary files: "proxy_temp"
  nginx http fastcgi temporary files: "fastcgi_temp"
  nginx http uwsgi temporary files: "uwsgi_temp"
  nginx http scgi temporary files: "scgi_temp"

[root@localhost nginx-1.12.0]# make && make install
```

006 .启动nginx，进入sbin目录,找到nginx启动命令



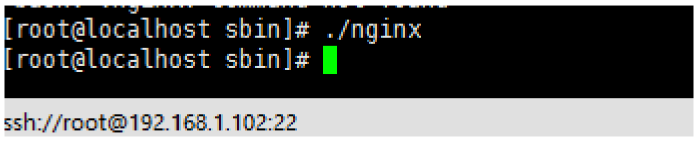
```
[root@localhost opt]# ls
mysite    nginx-1.12.0         python36     Python-3.6.6.tgz  redis-4.0.10.tar.gz  rh
nginx112  nginx-1.12.0.tar.gz  Python-3.6.6  redis-4.0.10      redis_conf
[root@localhost opt]#
```

```
cd /opt/nginx112/sbin
./nginx #启动
./nginx -s stop #关闭
./nginx -s reload # 平滑重启 ，修改了nginx.conf之后，可以不重启服务，加载新的配置
或者  /opt/nginx112/sbin/nginx -s reload  # 绝对路径平滑重启
```



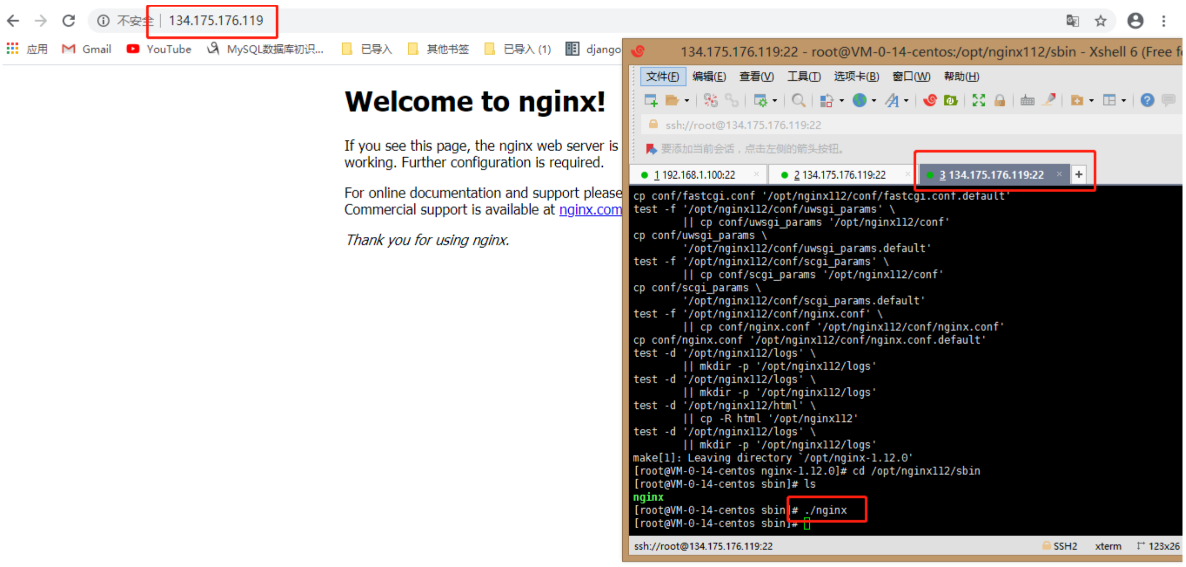
```
[root@localhost opt]# cd /opt/nginx112/sbin
[root@localhost sbin]# ls
nginx
[root@localhost sbin]#
```

启动：



```
[root@localhost sbin]# ./nginx
[root@localhost sbin]#
ssh://root@192.168.1.102:22
```

然后使用本机的服务器ip访问：如果有线上服务器就用线上服务器的外网ip访问，不需要加 端口号



不安全 | 134.175.176.119
应用
Gmail
YouTube
MySQL数据库初识...
已导入
其他书签
已导入 (1)
django
134.175.176.119:22 - root@VM-0-14-centos:/opt/nginx112/sbin - Xshell 6 (Free f
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root@134.175.176.119:22
要添加当前会话，点击左侧的箭头按钮。
1 192.168.1.100:22
2 134.175.176.119:22
3 134.175.176.119:22

Welcome to nginx!

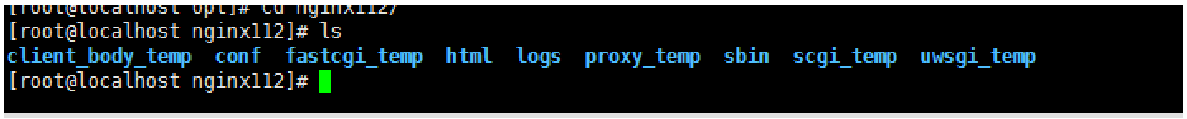
If you see this page, the nginx web server is
working. Further configuration is required.

For online documentation and support please
Commercial support is available at nginx.com

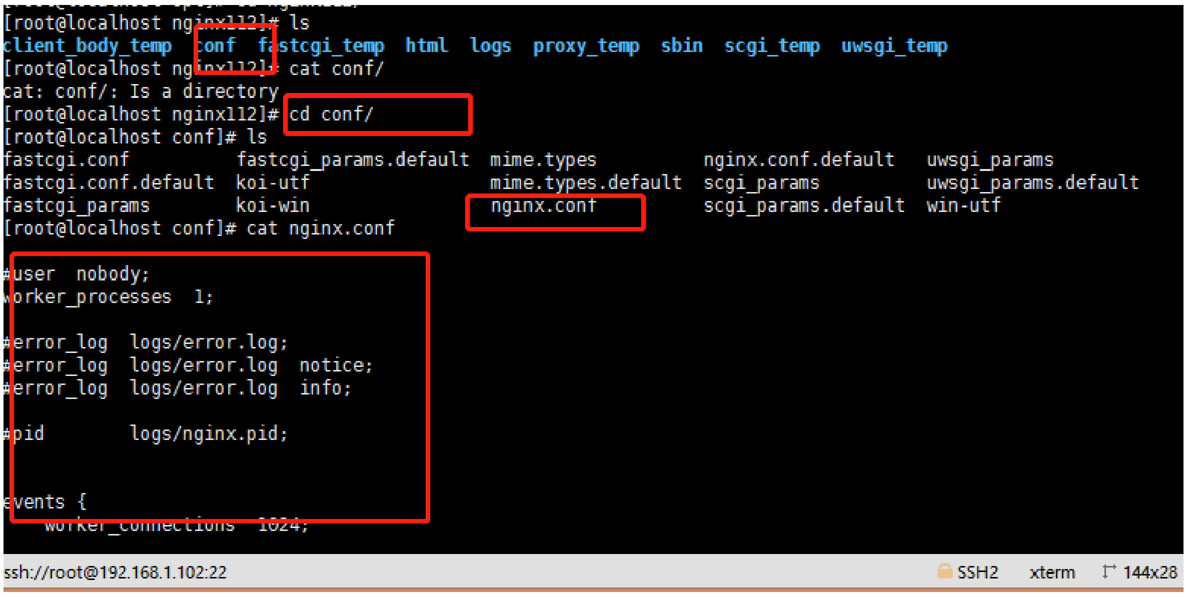
Thank you for using nginx.

```
cp conf/fastcgi.conf '/opt/nginx112/conf/fastcgi.conf.default'
test -f '/opt/nginx112/conf/uwsgi_params' \
        || cp conf/uwsgi_params '/opt/nginx112/conf'
cp conf/uwsgi_params \
        '/opt/nginx112/conf/uwsgi_params.default'
test -f '/opt/nginx112/conf/scgi_params' \
        || cp conf/scgi_params '/opt/nginx112/conf'
cp conf/scgi_params \
        '/opt/nginx112/conf/scgi_params.default'
test -f '/opt/nginx112/conf/nginx.conf' \
        || cp conf/nginx.conf '/opt/nginx112/conf/nginx.conf'
cp conf/nginx.conf '/opt/nginx112/conf/nginx.conf.default'
test -d '/opt/nginx112/logs' \
        || mkdir -p '/opt/nginx112/logs'
test -d '/opt/nginx112/logs' \
        || mkdir -p '/opt/nginx112/logs'
test -d '/opt/nginx112/html' \
        || cp -R html '/opt/nginx112'
test -d '/opt/nginx112/logs' \
        || mkdir -p '/opt/nginx112/logs'
make[1]: Leaving directory `/opt/nginx-1.12.0'
[root@VM-0-14-centos nginx-1.12.0]# cd /opt/nginx112/sbin
[root@VM-0-14-centos sbin]# ls
nginx
[root@VM-0-14-centos sbin]# ./nginx
[root@VM-0-14-centos sbin]#
ssh://root@134.175.176.119:22                    SSH2    xterm    123x26
```

007 nginx的目录结构

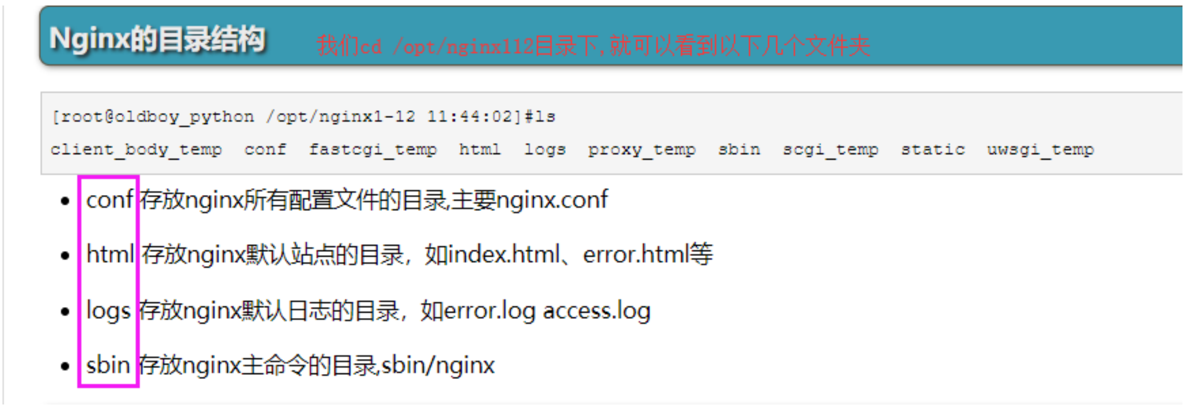


```
[root@localhost nginx112]# ls
client_body_temp  conf  fastcgi_temp  html  logs  proxy_temp  sbin  scgi_temp  uwsgi_temp
[root@localhost nginx112]# 
```



```
[root@localhost nginx112]# ls
client_body_temp  conf  fastcgi_temp  html  logs  proxy_temp  sbin  scgi_temp  uwsgi_temp
[root@localhost nginx112]# cat conf/
cat: conf/: Is a directory
[root@localhost nginx112]# cd conf/
[root@localhost conf]# ls
fastcgi.conf          fastcgi_params.default  mime.types          nginx.conf.default   uwsgi_params
fastcgi.conf.default  koi-utf                 mime.types.default  scgi_params          uwsgi_params.default
fastcgi_params        koi-win                 nginx.conf          scgi_params.default  win-utf
[root@localhost conf]# cat nginx.conf

#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;


events {
    worker_connections  1024;
```

ssh://root@192.168.1.102:22 SSH2 xterm 144x28

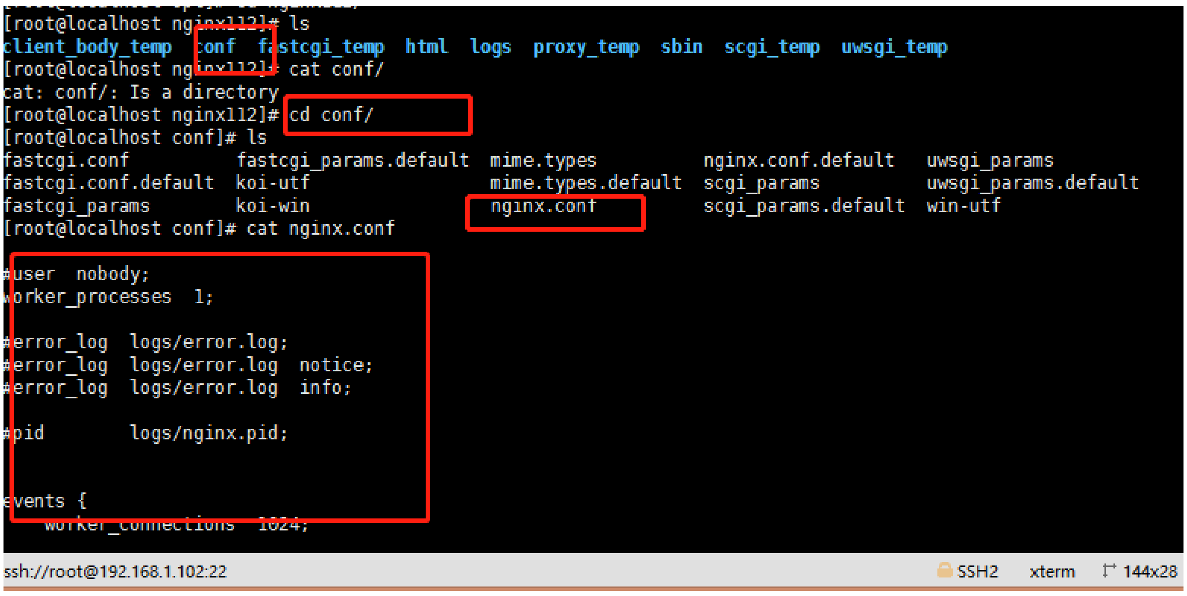


Nginx的目录结构 我们cd /opt/nginx112目录下,就可以看到以下几个文件夹

```
[root@oldboy_python /opt/nginx1-12 11:44:02]#ls
client_body_temp  conf  fastcgi_temp  html  logs  proxy_temp  sbin  scgi_temp  static  uwsgi_temp
```

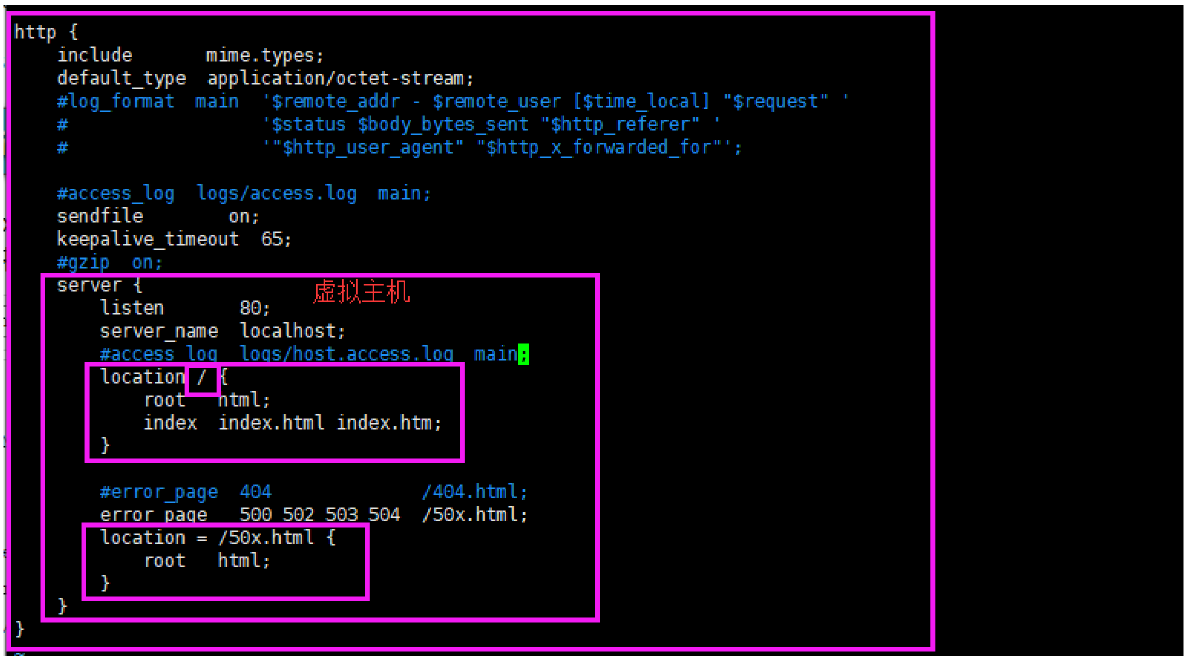
- conf 存放nginx所有配置文件的目录,主要nginx.conf
- html 存放nginx默认站点的目录，如index.html、error.html等
- logs 存放nginx默认日志的目录，如error.log access.log
- sbin 存放nginx主命令的目录,sbin/nginx

008 nginx配置文件详解



```
[root@localhost nginx112]# ls
client_body_temp  conf  fastcgi_temp  html  logs  proxy_temp  sbin  scgi_temp  uwsgi_temp
[root@localhost nginx112]# cat conf/
cat: conf/: Is a directory
[root@localhost nginx112]# cd conf/
[root@localhost conf]# ls
fastcgi.conf          fastcgi_params.default  mime.types          nginx.conf.default   uwsgi_params
fastcgi.conf.default  koi-utf                 mime.types.default  scgi_params          uwsgi_params.default
fastcgi_params        koi-win                 nginx.conf          scgi_params.default  win-utf
[root@localhost conf]# cat nginx.conf

#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;


events {
    worker_connections  1024;
```

ssh://root@192.168.1.102:22 SSH2 xterm 144x28



```
http {
    include       mime.types;
    default_type  application/octet-stream;
    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;
    sendfile        on;
    keepalive_timeout  65;
    #gzip  on;
    server {
        listen       80;
        server_name  localhost;
        #access_log  logs/host.access.log  main;
        location / {
            root   html;
            index  index.html index.htm;
        }

        #error_page  404              /404.html;
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }
}
```

```
#定义nginx工作进程数
worker_processes  5;
#错误日志
#error_log  logs/error.log;
#http定义代码主区域
http {
    include       mime.types;
    default_type  application/octet-stream;
    #定义nginx的访问日志功能
    #nginx会有一个accses.log功能，查看用户访问的记录
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    #开启日志功能
    access_log  logs/access.log  main;
    sendfile        on;
    keepalive_timeout  65;
    #开启gzip压缩传输
    gzip  on;
    #虚拟主机1  定义一个 斗鱼网站
    server {
        #定义nginx的访问入口端口，访问地址是  192.168.11.37:80
        listen       80;
        #定义网站的域名www.woshidouyu.tv
        #如果没有域名，就填写服务器的ip地址  192.168.11.37
        server_name  www.woshidouyu.tv;
        #nginx的url域名匹配
        #只要请求来自于www.woshidouyu.tv/111111111
        #只要请求来自于www.woshidouyu.tv/qweqwewqe
        #最低级的匹配，只要来自于www.woshidouyu.tv这个域名，都会走到这个location
        location / {
            #这个root参数，也是关键字，定义网页的根目录
            #以nginx安装的目录为相对路径  /opt/nginx112/html
            #可以自由修改这个root定义的网页根目录
            root   html;
            #index参数定义网站的首页文件名，默认的文件名
            index  index.html index.htm;
        }
        #错误页面的优化(只要是遇到前面4系列的错误，就会直接跳转到相对目录下的40x.html页面)
        error_page  400 401  402  403  404   /40x.html;
    }
}
```

009、 跑一个斗鱼网站出来

```
    server {
        listen       80;
        server_name  www.qishi2douyu.com;
        #access_log  logs/host.access.log  main;
        location / {
            root   /opt/qishi2douyu/;
            index  index.html index.html;
        }

        #error_page  404              /404.html;
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }
```

二、实际部署myblog项目到centos：创建myblog的专用虚拟环境进行项目的部署

一个非常详细的部署博客：

```
https://www.django.cn/article/show-4.html#banqian  # 这个比较详细点
https://www.jianshu.com/p/956debe2891d
```

在root目录创建一个myblog目录，将 项目文件压缩后，和requirements.txt,myblog.sql一并上传到这个目录

前提：

(1) 收集静态文件 ,修改settings.py配置文件， 加入以下一行配置 STATIC_ROOT='/opt/myblog/static'

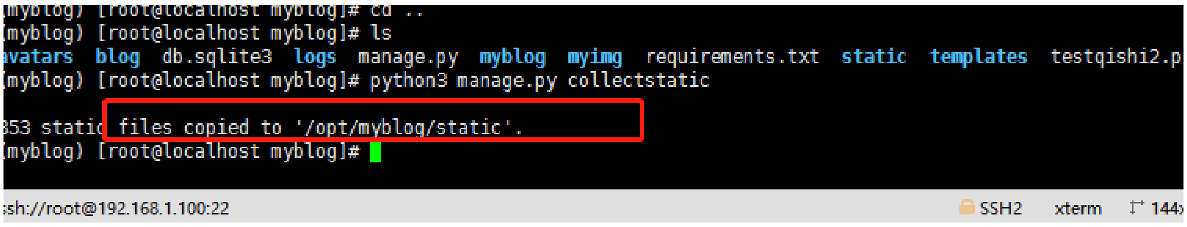
```
STATIC_ROOT='/opt/myblog/static'
```

(2) 创建静态文件存放目录/opt/myblog/static

```
mkdir -p /opt/myblog/static
```

(3) 一条命令收集django项目下面的所有静态文件(切换到存放luffy后端有manage.py的文件下),可以在虚拟环境下收集

```
python3 manage.py collectstatic
```



```
(myblog) [root@localhost myblog]# cd ..
(myblog) [root@localhost myblog]# ls
avatars  blog  db.sqlite3  logs  manage.py  myblog  myimg  requirements.txt  static  templates  testqishi2.p
(myblog) [root@localhost myblog]# python3 manage.py collectstatic

353 static files copied to '/opt/myblog/static'.
(myblog) [root@localhost myblog]#
```

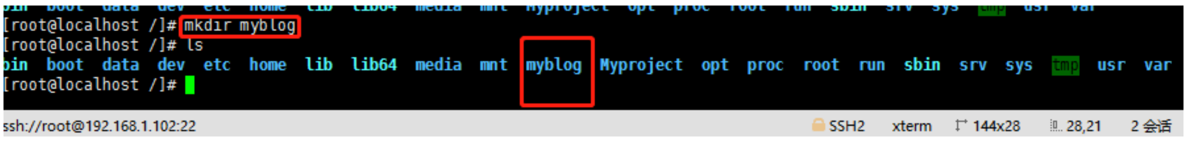
ssh://root@192.168.1.100:22 SSH2 xterm 144x

001、备份本地数据库和导出项目使用的安装包

```
将本地的django项目的数据库备份出来，进入mysql，使用命令行导出
将mylog这个数据库导出到d盘的615文件夹下的myblog.sql

mysqldump -h localhost -u root -p myblog >d:\615\myblog.sql


导出项目使用的安装包
pip3 freeze > requirements.txt
```

002、专门创建一个myblog的文件夹来保存项目和requirements.txt和myblog.sql

```
在 root/ 新建一个myblog文件夹
[root@localhost /]# mkdir myblog
```

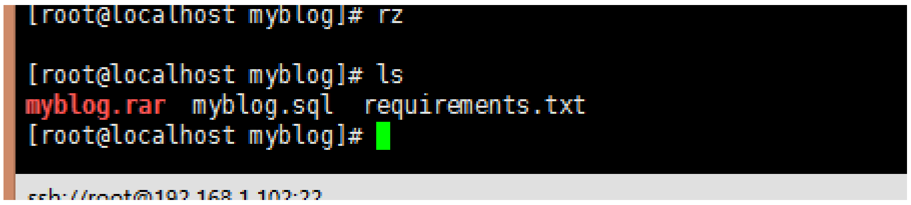


```
bin  boot  data  dev  etc  home  lib  lib64  media  mnt  Myproject  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@localhost /]# mkdir myblog
[root@localhost /]# ls
bin  boot  data  dev  etc  home  lib  lib64  media  mnt  myblog  Myproject  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@localhost /]#
```

ssh://root@192.168.1.102:22 SSH2 xterm 144x28 28,21 2 会话

003、使用rz命令上传文件

```
先安装yum install lrzsz -y

rz
```

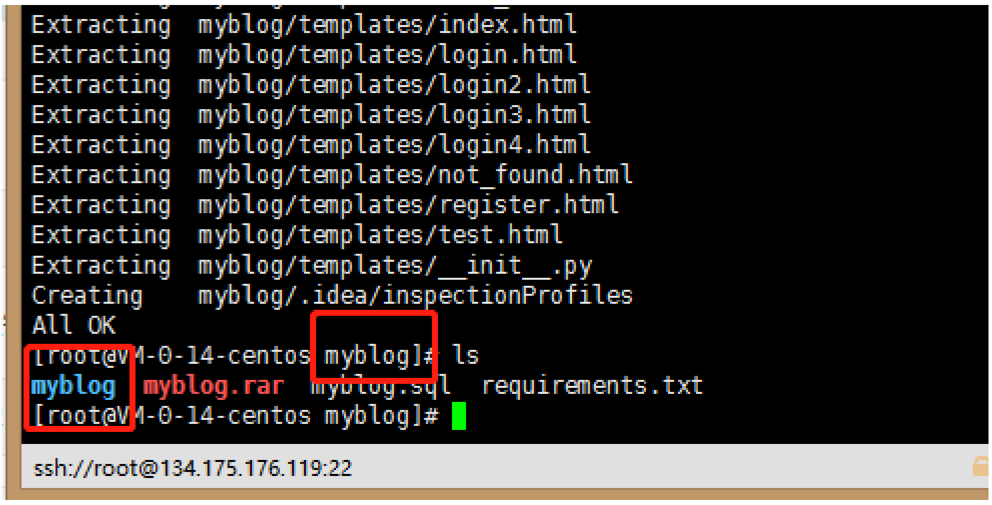


```
[root@localhost myblog]# rz

[root@localhost myblog]# ls
myblog.rar  myblog.sql  requirements.txt
[root@localhost myblog]#
```

ssh://root@192.168.1.102:22

004、由于上传的项目是rar，所以需要解压

```
下载解压软件
wget http://www.rarlab.com/rar/rarlinux-x64-5.3.0.tar.gz
安装
tar zxvf rarlinux-x64-5.3.0.tar.gz

cd rar
make
解压当前目录：rar x test.rar //解压 test.rar 到当前目录
压缩：rar test.rar ./test/ //将 test目录打包为 test.rar
```



```
Extracting  myblog/templates/index.html
Extracting  myblog/templates/login.html
Extracting  myblog/templates/login2.html
Extracting  myblog/templates/login3.html
Extracting  myblog/templates/login4.html
Extracting  myblog/templates/not_found.html
Extracting  myblog/templates/register.html
Extracting  myblog/templates/test.html
Extracting  myblog/templates/__init__.py
Creating    myblog/.idea/inspectionProfiles
All OK
[root@VM-0-14-centos myblog]# ls
myblog  myblog.rar  myblog.sql  requirements.txt
[root@VM-0-14-centos myblog]#
```

ssh://root@134.175.176.119:22

005、创建新的虚拟环境

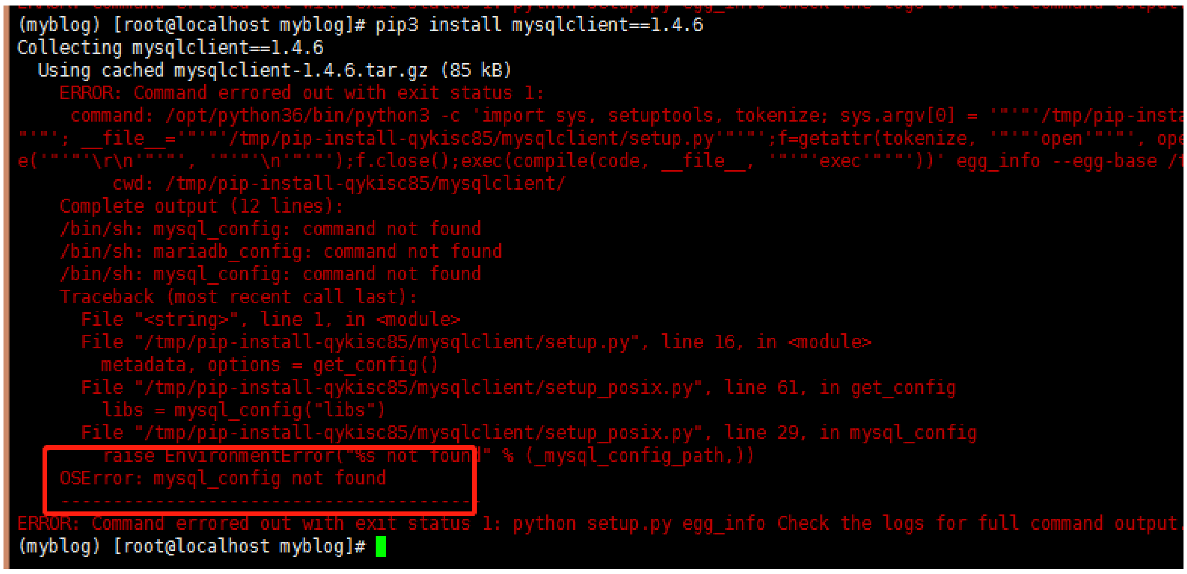
```
mkvirtualenv myblog
```

在虚拟环境下安装mysqlclient报错，主要错误：

```
OSError: mysql_config not found
```

```
   ERROR: Command errored out with exit status 1:
     command: /opt/python36/bin/python3 -c 'import sys, setuptools, tokenize;
sys.argv[0] = '"'"'/tmp/pip-install-qykisc85/mysqlclient/setup.py'"'"';
__file__='"'"'/tmp/pip-install-
qykisc85/mysqlclient/setup.py'"'"';f=getattr(tokenize, '"'"'open'"'"', open)
(__file__);code=f.read().replace('"'"'\r\n'"'"',
'"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' egg_info
--egg-base /tmp/pip-pip-egg-info-q5jii0tc
         cwd: /tmp/pip-install-qykisc85/mysqlclient/
    Complete output (12 lines):
    /bin/sh: mysql_config: command not found
    /bin/sh: mariadb_config: command not found
    /bin/sh: mysql_config: command not found
    Traceback (most recent call last):
      File "<string>", line 1, in <module>
      File "/tmp/pip-install-qykisc85/mysqlclient/setup.py", line 16, in
<module>
        metadata, options = get_config()
      File "/tmp/pip-install-qykisc85/mysqlclient/setup_posix.py", line 61, in
get_config
        libs = mysql_config("libs")
      File "/tmp/pip-install-qykisc85/mysqlclient/setup_posix.py", line 29, in
mysql_config
```

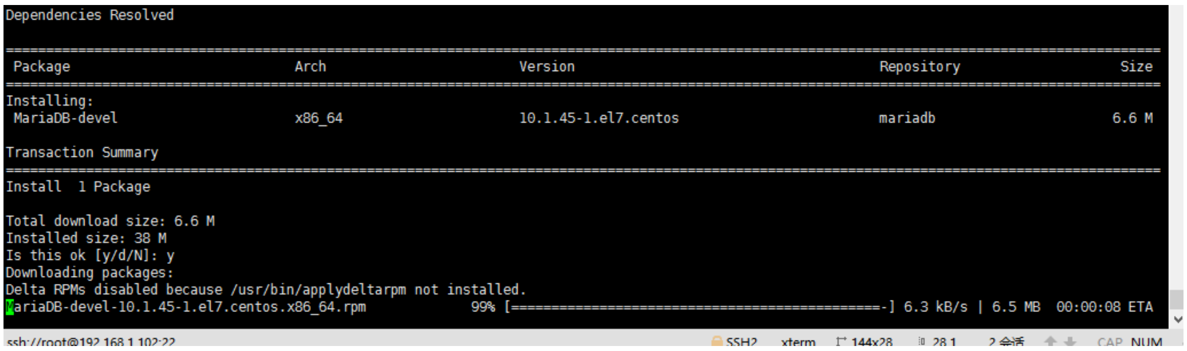
```
        raise EnvironmentError("%s not found" % (_mysql_config_path,))
    OSError: mysql_config not found
    ----------------------------------------
ERROR: Command errored out with exit status 1: python setup.py egg_info Check
the logs for full command output.
```



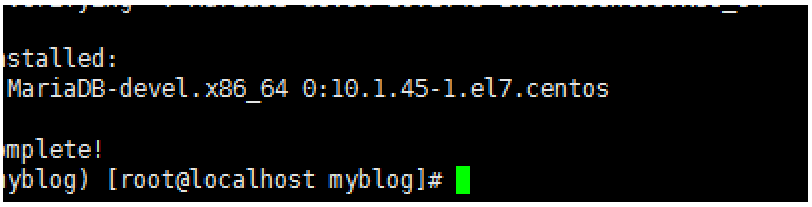
```
(myblog) [root@localhost myblog]# pip3 install mysqlclient==1.4.6
Collecting mysqlclient==1.4.6
  Using cached mysqlclient-1.4.6.tar.gz (85 kB)
    ERROR: Command errored out with exit status 1:
     command: /opt/python36/bin/python3 -c 'import sys, setuptools, tokenize; sys.argv[0] = '"'"'/tmp/pip-insta
'"'"'; __file__='"'"'/tmp/pip-install-qykisc85/mysqlclient/setup.py'"'"';f=getattr(tokenize, '"'"'open'"'"', ope
e('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'exec'"'"'))' egg_info --egg-base /t
         cwd: /tmp/pip-install-qykisc85/mysqlclient/
    Complete output (12 lines):
    /bin/sh: mysql_config: command not found
    /bin/sh: mariadb_config: command not found
    /bin/sh: mysql_config: command not found
    Traceback (most recent call last):
      File "<string>", line 1, in <module>
      File "/tmp/pip-install-qykisc85/mysqlclient/setup.py", line 16, in <module>
        metadata, options = get_config()
      File "/tmp/pip-install-qykisc85/mysqlclient/setup_posix.py", line 61, in get_config
        libs = mysql_config("libs")
      File "/tmp/pip-install-qykisc85/mysqlclient/setup_posix.py", line 29, in mysql_config
        raise EnvironmentError("%s not found" % (_mysql_config_path,))
    OSError: mysql_config not found
    ----------------------------------------
ERROR: Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.
(myblog) [root@localhost myblog]#
```

解决方案：

```
先安装mysql-deve , yum install mysql-devel
会自动找到mriadb-deve 进行安装
再安装mysqlclient
pip3 install mysqlclient==1.4.6
```



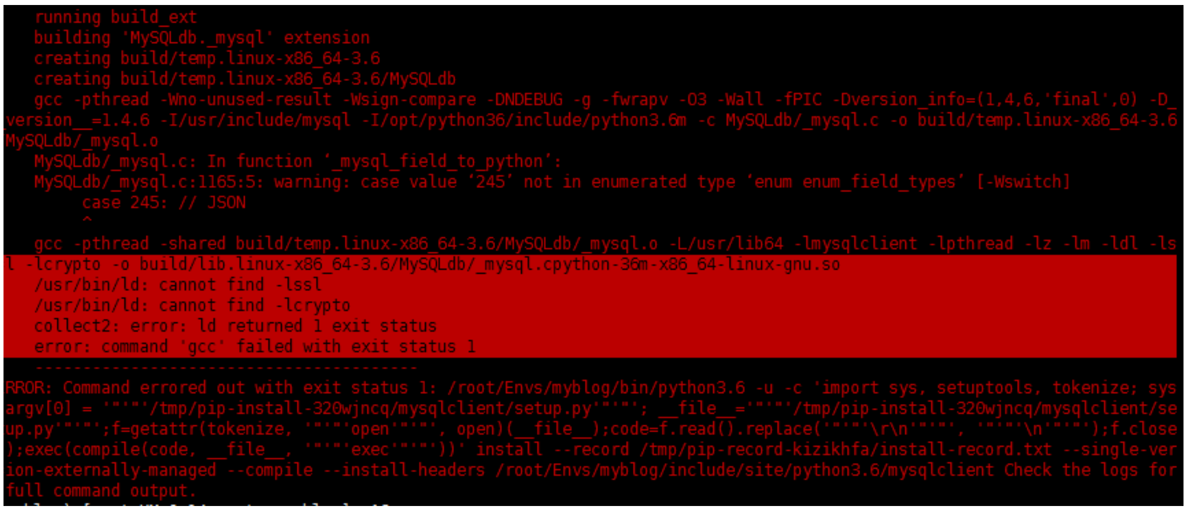
```
Dependencies Resolved

================================================================================================================
 Package                  Arch                Version                            Repository               Size
================================================================================================================
Installing:
 MariaDB-devel            x86_64              10.1.45-1.el7.centos               mariadb                  6.6 M

Transaction Summary
================================================================================================================
Install  1 Package

Total download size: 6.6 M
Installed size: 38 M
Is this ok [y/d/N]: y
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
MariaDB-devel-10.1.45-1.el7.centos.x86_64.rpm          99% [============================================-] 6.3 kB/s | 6.5 MB  00:00:08 ETA
```



```
stalled:
MariaDB-devel.x86_64 0:10.1.45-1.el7.centos

mplete!
yblog) [root@localhost myblog]#
```

安装上面之后可能还会报错：

```
sl -lcrypto -o build/lib.linux-x86_64-3.6/MySQLdb/_mysql.cpython-36m-x86_64-
linux-gnu.so
    /usr/bin/ld: cannot find -lssl
    /usr/bin/ld: cannot find -lcrypto
    collect2: error: ld returned 1 exit status
    error: command 'gcc' failed with exit status 1
```



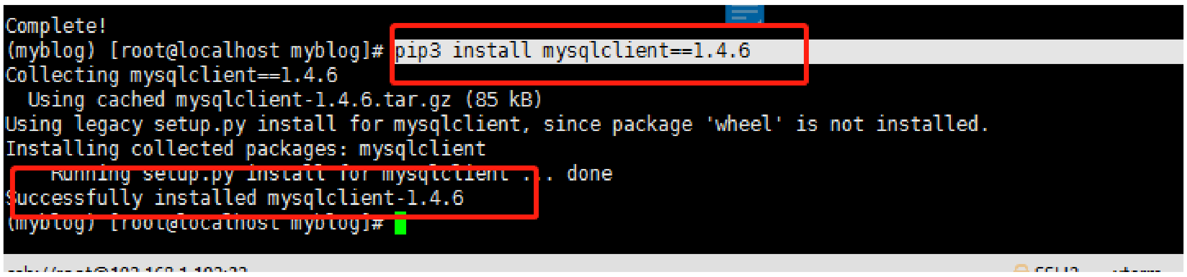
```
   running build_ext
   building 'MySQLdb._mysql' extension
   creating build/temp.linux-x86_64-3.6
   creating build/temp.linux-x86_64-3.6/MySQLdb
   gcc -pthread -Wno-unused-result -Wsign-compare -DNDEBUG -g -fwrapv -O3 -Wall -fPIC -Dversion_info=(1,4,6,'final',0) -D_
version__=1.4.6 -I/usr/include/mysql -I/opt/python36/include/python3.6m -c MySQLdb/_mysql.c -o build/temp.linux-x86_64-3.6
MySQLdb/_mysql.o
   MySQLdb/_mysql.c: In function '_mysql_field_to_python':
   MySQLdb/_mysql.c:1165:5: warning: case value '245' not in enumerated type 'enum enum_field_types' [-Wswitch]
        case 245: // JSON
        ^
   gcc -pthread -shared build/temp.linux-x86_64-3.6/MySQLdb/_mysql.o -L/usr/lib64 -lmysqlclient -lpthread -lz -lm -ldl -ls
l -lcrypto -o build/lib.linux-x86_64-3.6/MySQLdb/_mysql.cpython-36m-x86_64-linux-gnu.so
   /usr/bin/ld: cannot find -lssl
   /usr/bin/ld: cannot find -lcrypto
   collect2: error: ld returned 1 exit status
   error: command 'gcc' failed with exit status 1
   ----------------------------------------
RROR: Command errored out with exit status 1: /root/Envs/myblog/bin/python3.6 -u -c 'import sys, setuptools, tokenize; sys
argv[0] = '"'"'/tmp/pip-install-320wjncq/mysqlclient/setup.py'"'"'; __file__='"'"'/tmp/pip-install-320wjncq/mysqlclient/se
up.py'"'"';f=getattr(tokenize, '"'"'open'"'"', open)(__file__);code=f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close
);exec(compile(code, __file__, '"'"'exec'"'"'))' install --record /tmp/pip-record-kizikhfa/install-record.txt --single-ver
ion-externally-managed --compile --install-headers /root/Envs/myblog/include/site/python3.6/mysqlclient Check the logs for
full command output.
```

安装运行库：

```
yum install gcc libffi-devel python-devel openssl-devel -y
```

再安装：mysqlclient==1.4.6

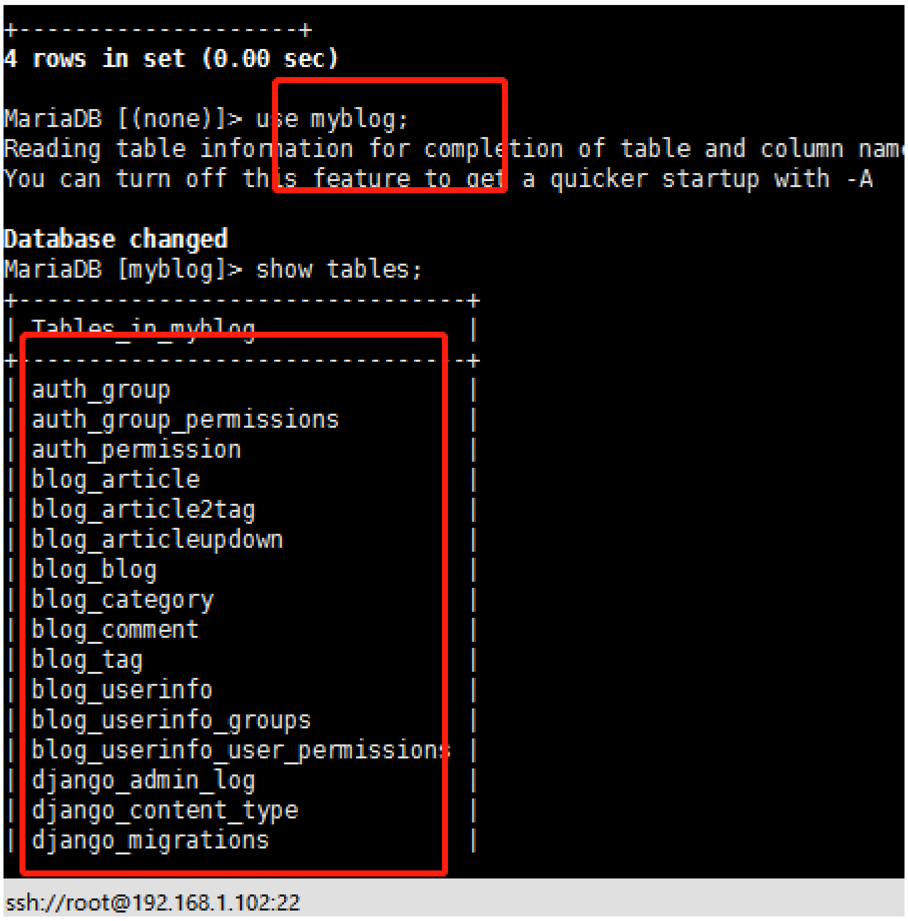
```
pip3 install mysqlclient==1.4.6
```



```
Complete!
(myblog) [root@localhost myblog]# pip3 install mysqlclient==1.4.6
Collecting mysqlclient==1.4.6
  Using cached mysqlclient-1.4.6.tar.gz (85 kB)
Using legacy setup.py install for mysqlclient, since package 'wheel' is not installed.
Installing collected packages: mysqlclient
    Running setup.py install for mysqlclient ... done
Successfully installed mysqlclient-1.4.6
(myblog) [root@localhost myblog]#
```

将myblog.sql 数据导入到centos上的mariadb数据库中

```
cd /myblog   # 有myblog.sql文件的文件夹下
然后进入mysql创建一个数据库
mysql -uroot -p 数据密码登陆
然后：
create database myblog;
接着退出mysql

执行：
[root@localhost myblog]# mysql -uroot -p myblog < myblog.sql
输入密码，然后进去查看
```

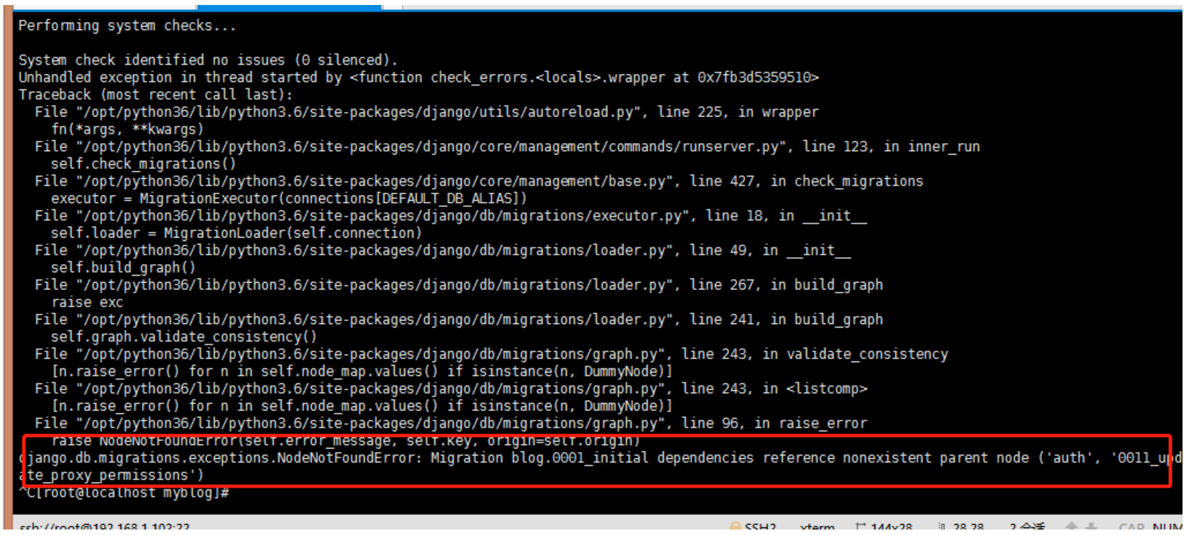


```
+--------------------+
4 rows in set (0.00 sec)

MariaDB [(none)]> use myblog;
Reading table information for completion of table and column nam
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [myblog]> show tables;
+-------------------------------+
| Tables_in_myblog              |
+-------------------------------+
| auth_group                    |
| auth_group_permissions        |
| auth_permission               |
| blog_article                  |
| blog_article2tag              |
| blog_articleupdown            |
| blog_blog                     |
| blog_category                 |
| blog_comment                  |
| blog_tag                      |
| blog_userinfo                 |
| blog_userinfo_groups          |
| blog_userinfo_user_permissions |
| django_admin_log              |
| django_content_type           |
| django_migrations             |
ssh://root@192.168.1.102:22
```

运行项目：会报错

项目部署参考博客：

```
https://www.cnblogs.com/djangocn/p/9538551.html
```



```
Performing system checks...

System check identified no issues (0 silenced).
Unhandled exception in thread started by <function check_errors.<locals>.wrapper at 0x7fb3d5359510>
Traceback (most recent call last):
  File "/opt/python36/lib/python3.6/site-packages/django/utils/autoreload.py", line 225, in wrapper
    fn(*args, **kwargs)
  File "/opt/python36/lib/python3.6/site-packages/django/core/management/commands/runserver.py", line 123, in inner_run
    self.check_migrations()
  File "/opt/python36/lib/python3.6/site-packages/django/core/management/base.py", line 427, in check_migrations
    executor = MigrationExecutor(connections[DEFAULT_DB_ALIAS])
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/executor.py", line 18, in __init__
    self.loader = MigrationLoader(self.connection)
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/loader.py", line 49, in __init__
    self.build_graph()
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/loader.py", line 267, in build_graph
    raise exc
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/loader.py", line 241, in build_graph
    self.graph.validate_consistency()
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/graph.py", line 243, in validate_consistency
    [n.raise_error() for n in self.node_map.values() if isinstance(n, DummyNode)]
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/graph.py", line 243, in <listcomp>
    [n.raise_error() for n in self.node_map.values() if isinstance(n, DummyNode)]
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/graph.py", line 96, in raise_error
    raise NodeNotFoundError(self.error_message, self.key, origin=self.origin)
django.db.migrations.exceptions.NodeNotFoundError: Migration blog.0001_initial dependencies reference nonexistent parent node ('auth', '0011_update_proxy_permissions')
^C[root@localhost myblog]#
```

```
    fn(*args, **kwargs)
  File "/opt/python36/lib/python3.6/site-packages/django/core/management/commands/runserver.py", line 123, in inner_run
    self.check_migrations()
  File "/opt/python36/lib/python3.6/site-packages/django/core/management/base.py", line 427, in check_migrations
```

```
    executor = MigrationExecutor(connections[DEFAULT_DB_ALIAS])
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/executor.py", line 18, in __init__
    self.loader = MigrationLoader(self.connection)
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/loader.py", line 49, in __init__
    self.build_graph()
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/loader.py", line 267, in build_graph
    raise exc
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/loader.py", line 241, in build_graph
    self.graph.validate_consistency()
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/graph.py", line 243, in validate_consistency
    [n.raise_error() for n in self.node_map.values() if isinstance(n, DummyNode)]
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/graph.py", line 243, in <listcomp>
    [n.raise_error() for n in self.node_map.values() if isinstance(n, DummyNode)]
  File "/opt/python36/lib/python3.6/site-packages/django/db/migrations/graph.py", line 96, in raise_error
    raise NodeNotFoundError(self.error_message, self.key, origin=self.origin)
django.db.migrations.exceptions.NodeNotFoundError: Migration blog.0001_initial dependencies reference nonexistent parent node ('auth', '0011_update_proxy_permissions')
```

出现错误的原因，django的版本问题，centos里面是2.0的，我的需要升级到3.0版本

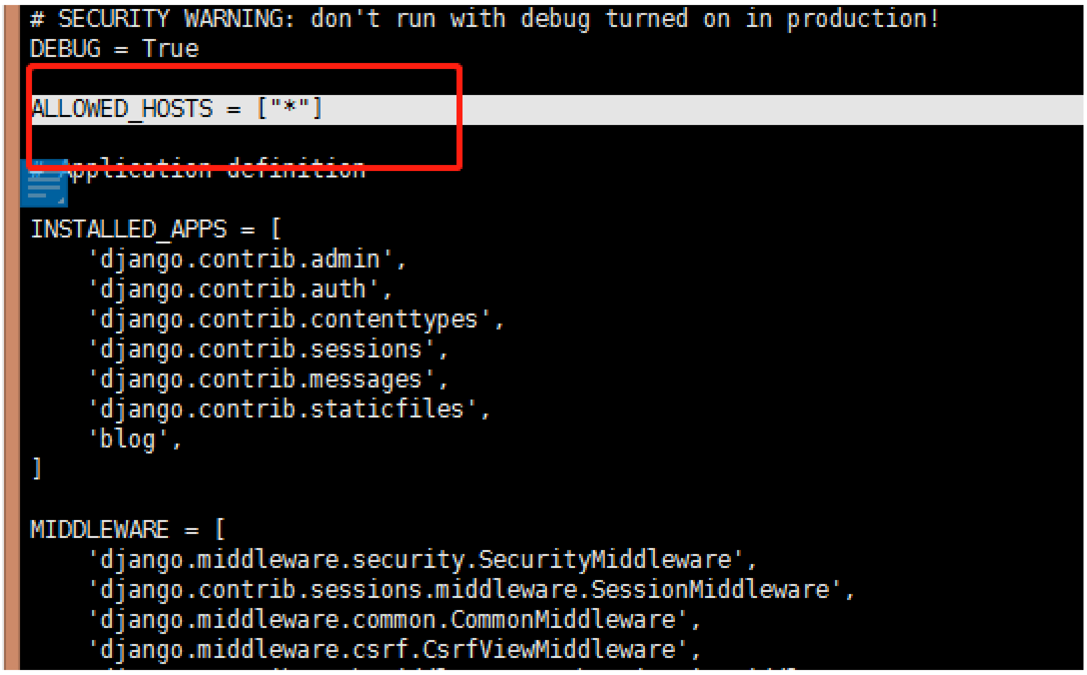
```
pip3 install --upgrade Django==3.0.6
```

重新进入项目文件，启动项目,需要进入虚拟环境运行

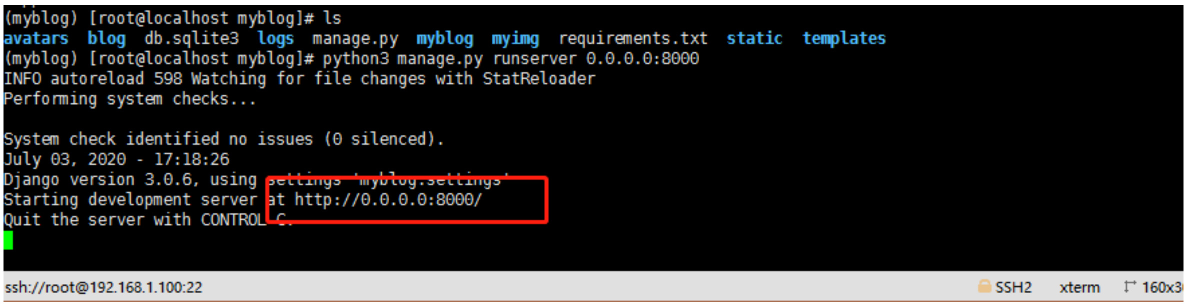
```
python3 manage.py runserver 0.0.0.0:8000
```

然后设置项目的settings.py文件的host

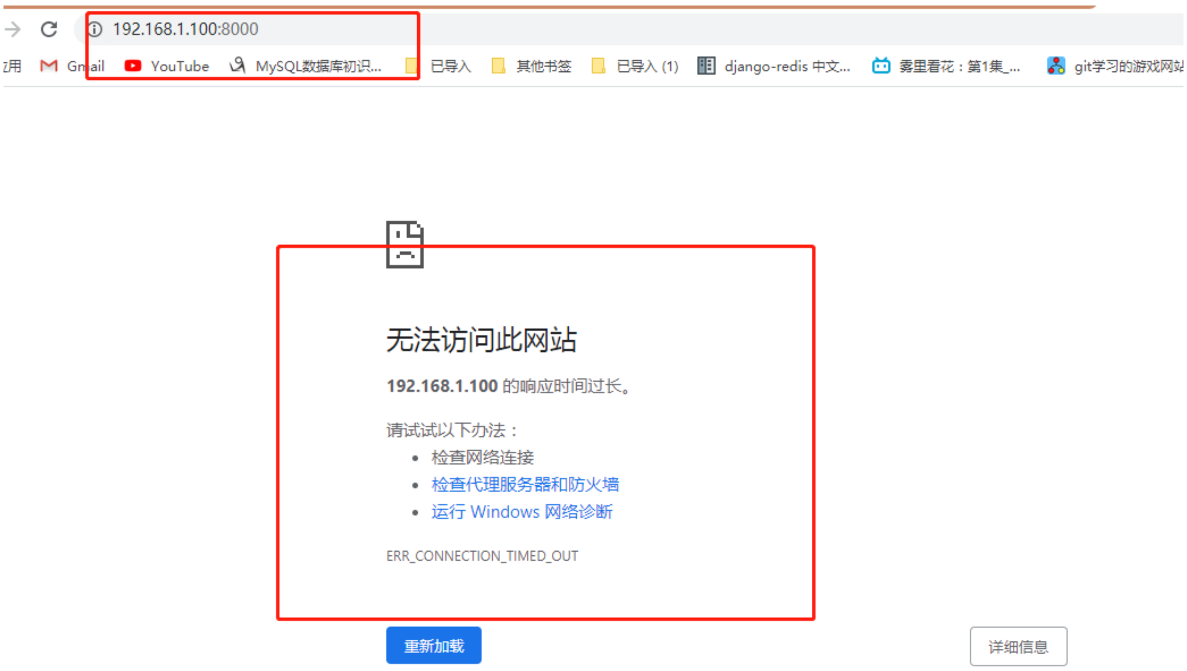
```
ALLOWED_HOSTS = ["*"]
```


```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["*"]

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'blog',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
```

启动没报错：，


```
(myblog) [root@localhost myblog]# ls
avatars  blog  db.sqlite3  logs  manage.py  myblog  myimg  requirements.txt  static  templates
(myblog) [root@localhost myblog]# python3 manage.py runserver 0.0.0.0:8000
INFO autoreload 598 Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
July 03, 2020 - 17:18:26
Django version 3.0.6, using settings 'myblog.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```
ssh://root@192.168.1.100:22   SSH2   xterm   160x3

浏览器使用当前ip加上指定端口访问，发现访问不到,这里使用manage.py 访问需要加端口的访问的


192.168.1.100:8000

无法访问此网站

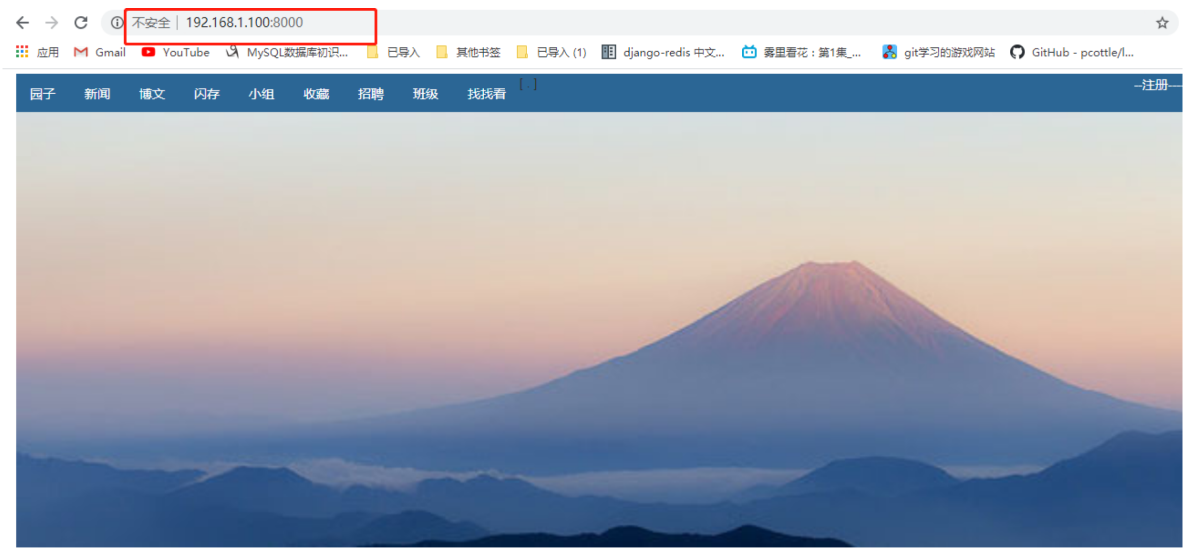
192.168.1.100 的响应时间过长。

请试试以下办法：
- 检查网络连接
- 检查代理服务器和防火墙
- 运行 Windows 网络诊断

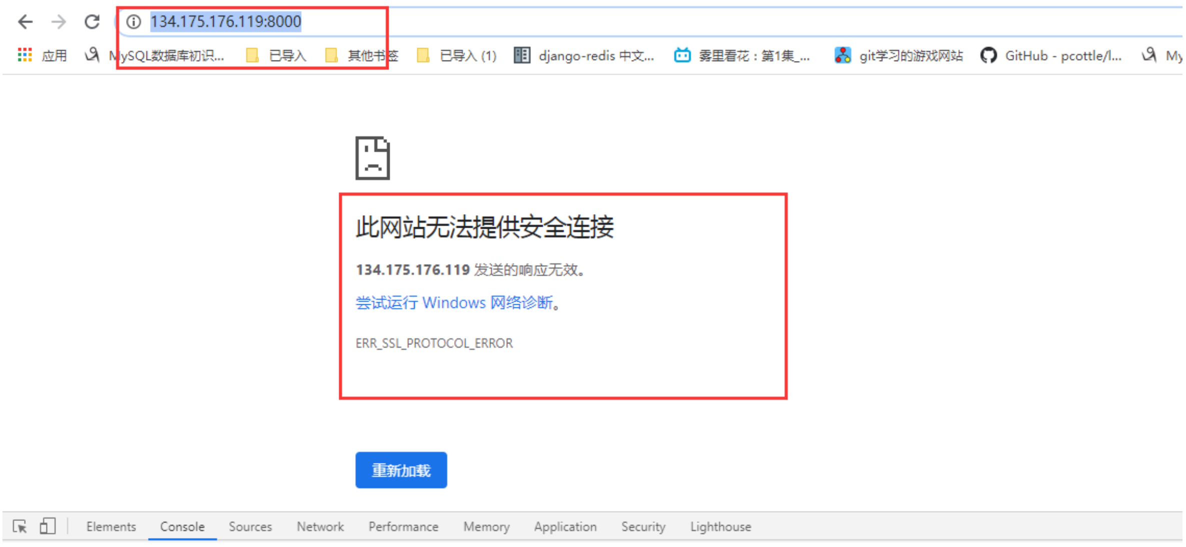
ERR_CONNECTION_TIMED_OUT

关闭防火墙：再重启项目，可以访问到了,当然，项目中settings.py的 修改为ALLOWED_HOSTS = ["*"]

```
然后关闭防火墙：systemctl stop firewalld.service   # 线上服务器好像不用关闭防火墙
```


不安全 | 192.168.1.100:8000
应用 Gmail YouTube MySQL数据库初识... 已导入 其他书签 已导入 (1) django-redis 中文... 雾里看花：第1集_... git学习的游戏网站 GitHub - pcottle/l...
园子 新闻 博文 闪存 小组 收藏 招聘 班级 找找看
--注册--

可能在会出现访问线上服务端ip的时候（https://134.175.176.119:8000/），在浏览器段会出现


134.175.176.119:8000
应用 MySQL数据库初识... 已导入 其他书签 已导入 (1) django-redis 中文... 雾里看花：第1集_... git学习的游戏网站 GitHub - pcottle/l...
重新加载
Elements Console Sources Network Performance Memory Application Security Lighthouse

此网站无法提供安全连接

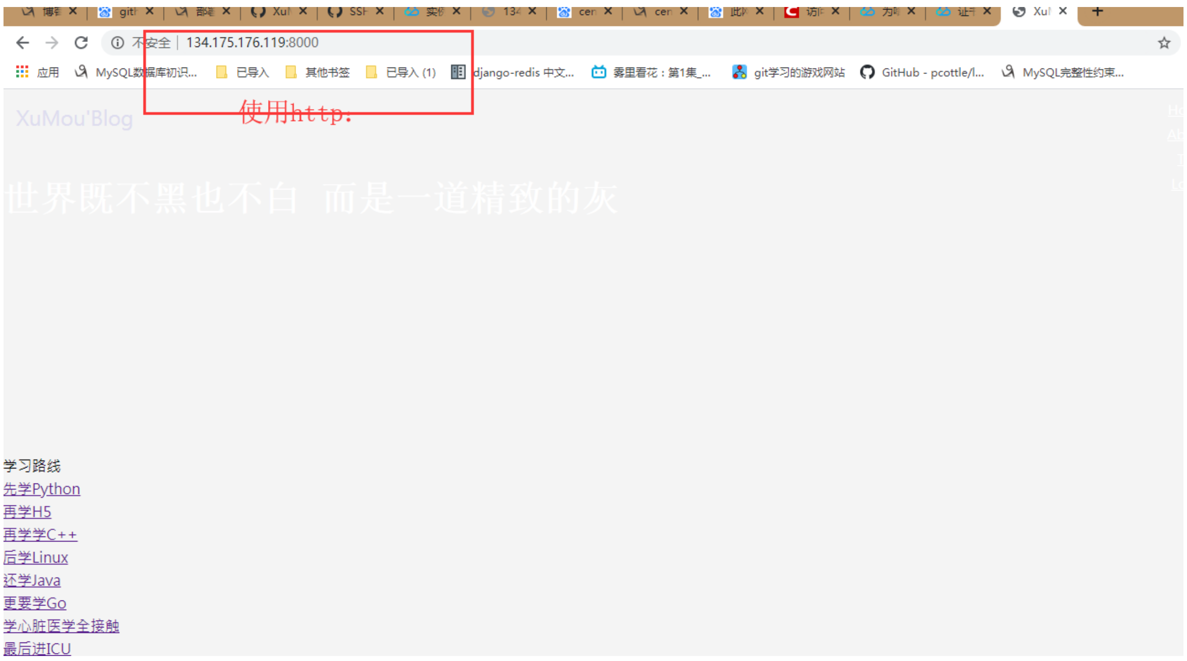
134.175.176.119 发送的响应无效。

尝试运行 Windows 网络诊断。

ERR_SSL_PROTOCOL_ERROR

原因是

，发现是因为https的问题，我在腾讯云上的服务是没有配置安全认证的，但是我在浏览器上输入IP和端口回车之后，浏览器默认补充了https，请求链接变为https://134.175.176.119:8000/，这久导致了上面的提示。去掉s，改为http://134.175.176.119:8000/就可以正常访问了。



学习路线
先学Python
再学H5
再学学C++
后学Linux
还学Java
更要学Go
学心脏医学全接触
最后进ICU

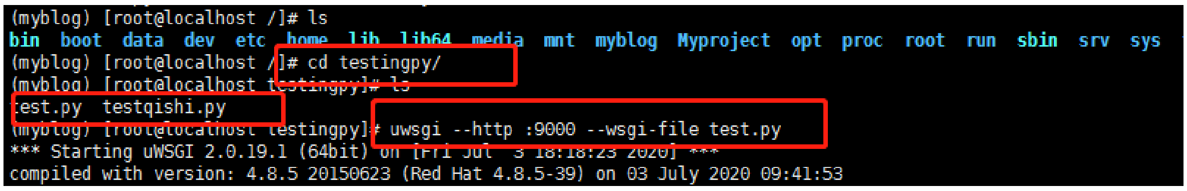
005、配置uwsgi

第一步：进入所在的myblog虚拟环境，安装uwsgi

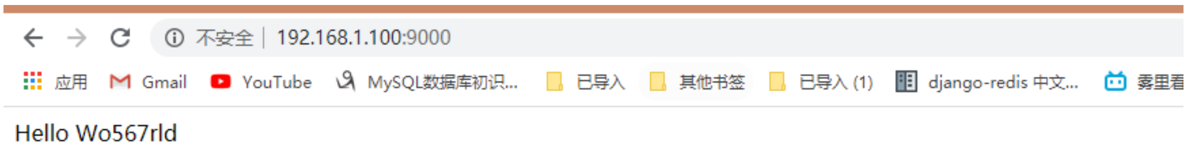
```
workon myblog
pip3 install uwsgi
```

2 uwsgi 启动测试文件

```
(1) 创建一个testqishi2.py
vim testqishi2.py
(2) 写入以下几行代码
def application(env, start_response):
    start_response('200 OK', [('Content-Type','text/html')])
    return [b"Hello World"] # python3
(3) 通过uwsgi命令将测试文件跑起来
uwsgi --http :9000 --wsgi-file testqishi2.py
uwsgi --http :8000 --wsgi-file testqishi.py
(4) 测试通过自己本机的IP+端口访问
http://192.168.12.56:9000
```



```
(myblog) [root@localhost /]# ls
bin  boot  data  dev  etc  home  lib  lib64  media  mnt  myblog  Myproject  opt  proc  root  run  sbin  srv  sys
(myblog) [root@localhost /]# cd testingpy/
(myblog) [root@localhost testingpy]# ls
test.py  testqishi.py
(myblog) [root@localhost testingpy]# uwsgi --http :9000 --wsgi-file test.py
*** Starting uWSGI 2.0.19.1 (64bit) on [Fri Jul  3 18:18:23 2020] ***
compiled with version: 4.8.5 20150623 (Red Hat 4.8.5-39) on 03 July 2020 09:41:53
```

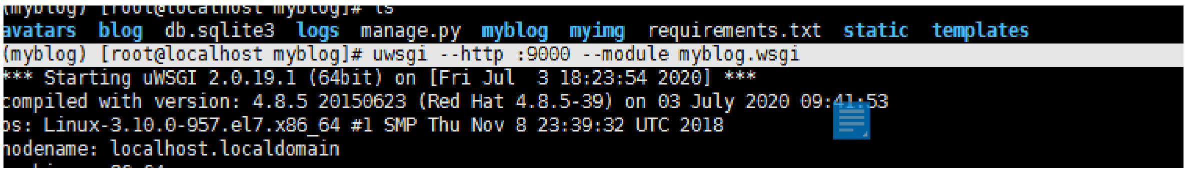


```
192.168.1.100:9000
Hello Wo567rld
```

3 uwsgi启动django项目

```
uwsgi --http :9000 --module myblog.wsgi
```

用uwsgi启动项目，必须要进入到后端django的项目下启动，在项目有manage.py的层面



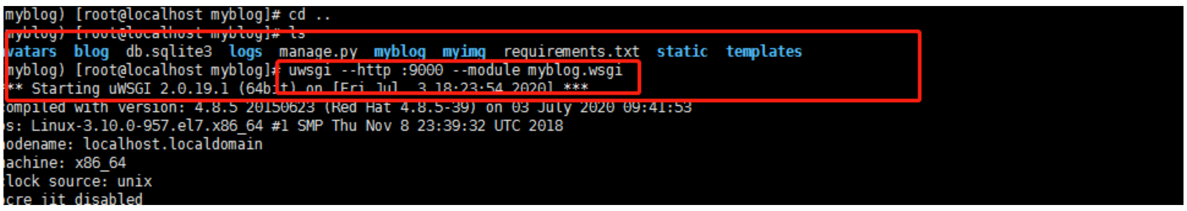
```
(myblog) [root@localhost myblog]# ls
avatars  blog  db.sqlite3  logs  manage.py  myblog  myimg  requirements.txt  static  templates
(myblog) [root@localhost myblog]# uwsgi --http :9000 --module myblog.wsgi
*** Starting uWSGI 2.0.19.1 (64bit) on [Fri Jul  3 18:23:54 2020] ***
compiled with version: 4.8.5 20150623 (Red Hat 4.8.5-39) on 03 July 2020 09:41:53
os: Linux-3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018
nodename: localhost.localdomain
```

我自己的启动命令是：

```
# 说明是启动myblog这个文件夹下的这个wsgi文件，启动项目用module，启动文件用file
uwsgi --http :9000 --module myblog.wsgi
```

进入到项目的manage.py层面，运行,指定端口为9000

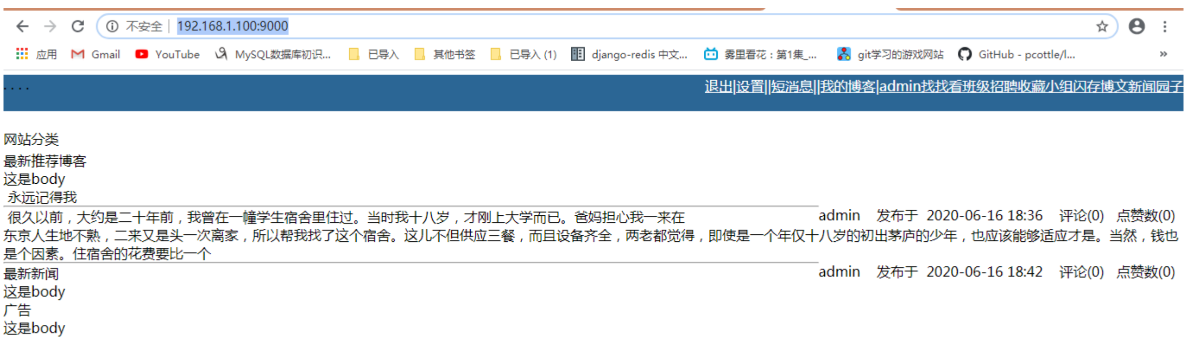
```
(myblog) [root@localhost myblog]# uwsgi --http :9000 --module myblog.wsgi
```



```
myblog) [root@localhost myblog]# cd ..
myblog) [root@localhost myblog]# ls
vatars  blog  db.sqlite3  logs  manage.py  myblog  myimg  requirements.txt  static  templates
myblog) [root@localhost myblog]# uwsgi --http :9000 --module myblog.wsgi
** Starting uWSGI 2.0.19.1 (64bit) on [Fri Jul  3 18:23:54 2020] ***
ompiled with version: 4.8.5 20150623 (Red Hat 4.8.5-39) on 03 July 2020 09:41:53
s: Linux-3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018
odename: localhost.localdomain
achine: x86_64
lock source: unix
cre jit disabled
```

浏览器当前ip+自己指定端口9000，发现没有css等基本样式,下一次运行有了css样式

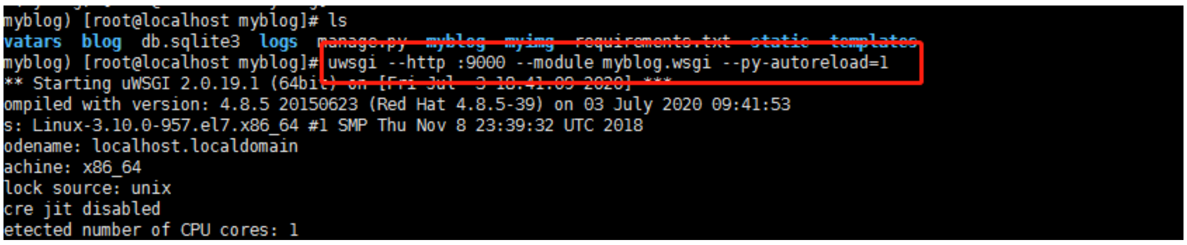
```
http://192.168.1.100:9000/
```



网站分类
最新推荐博客
这是body
永远记得我
很久以前，大约是二十年前，我曾在一幢学生宿舍里住过。当时我十八岁，才刚上大学而已。爸妈担心我一来在东京人生地不熟，二来又是头一次离家，所以帮我找了这个宿舍。这儿不但供应三餐，而且设备齐全，两老都觉得，即使是一个年仅十八岁的初出茅庐的少年，也应该能够适应才是。当然，钱也是个因素。住宿舍的花费要比一个
admin 发布于 2020-06-16 18:36 评论(0) 点赞数(0)
最新新闻
admin 发布于 2020-06-16 18:42 评论(0) 点赞数(0)
这是body
广告
这是body

使用uwsgi热加载来启动项目，启动成功

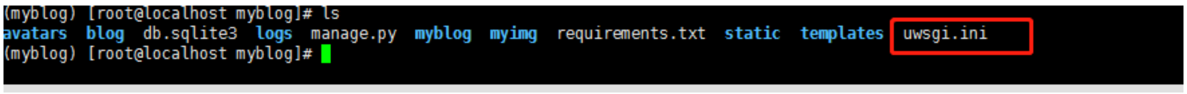
```
uwsgi --http :9000 --module myblog.wsgi --py-autoreload=1
```



```
myblog) [root@localhost myblog]# ls
vatars  blog  db.sqlite3  logs  manage.py  myblog  myimg  requirements.txt  static  templates
myblog) [root@localhost myblog]# uwsgi --http :9000 --module myblog.wsgi --py-autoreload=1
** Starting uWSGI 2.0.19.1 (64bit) on [Fri Jul  3 10:41:09 2020] ***
ompiled with version: 4.8.5 20150623 (Red Hat 4.8.5-39) on 03 July 2020 09:41:53
s: Linux-3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018
odename: localhost.localdomain
achine: x86_64
lock source: unix
cre jit disabled
etected number of CPU cores: 1
```

使用uwsgi配置文件启动django项目

创建一个配置文件uwsgi.ini(注意，这个配置文件不管放在哪里都可以，但是启动的时候，需要找到这个配置文件)
我放在 myblog/myblog 这个文件夹下



```
(myblog) [root@localhost myblog]# ls
avatars  blog  db.sqlite3  logs  manage.py  myblog  myimg  requirements.txt  static  templates  uwsgi.ini
(myblog) [root@localhost myblog]#
```

```
vim uwsgi.ini

#socket和http任选其一  WORKON_HOME=~/Envs
#我安装虚拟环境的时候设置了安装路径    ~/Envs/myblog,但是这里的home 路径却是 /root,
命令行  cd /root  是有Envs目录的
[uwsgi]
chdir           = /myblog/myblog/
module          = myblog.wsgi
home            = /root/Envs/myblog
master          = true
processes       = 1
socket          = 0.0.0.0:9000
#http            = 0.0.0.0:9000
vacuum          = true
```

过指定uwsgi.ini配置文件把django跑起来(myblog的ini文件在myblog/myblog/)

```
uwsgi uwsgi.ini
```

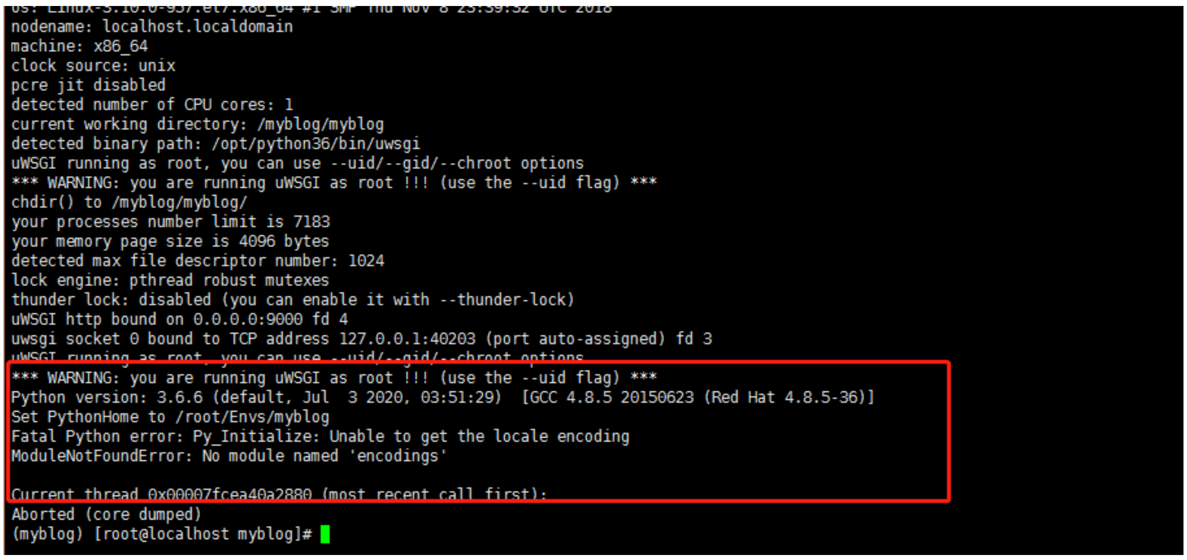
前端会显示"该网页无法正常运作"

只使用uwsgi.ini 启动django项目的时候，前端使用线上服务器加端口访问的时候，前端会显示"该网页无法正常运作" 这是因为，只使用uwsgi配置的时候，应该选择 "运行一个http服务端，就用这个http参数"，当然现实的时没有css样式

```
有时配置好uwsgi.ini后，前端
#如果你使用了nginx，做反向代理，必须填写socket链接，而不是http参数
socket          = 0.0.0.0:8000
#如果你不用nginx，直接使用uwsgi，运行一个http服务端，就用这个http参数
http = 0.0.0.0:8000
```

此时启动报错：

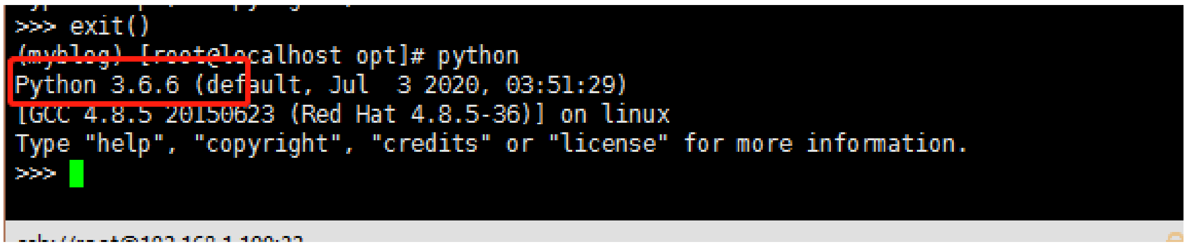
```
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
Python version: 3.6.6 (default, Jul  3 2020, 03:51:29)  [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]
Set PythonHome to /root/Envs/myblog
Fatal Python error: Py_Initialize: Unable to get the locale encoding
ModuleNotFoundError: No module named 'encodings'

Current thread 0x00007fcea40a2880 (most recent call first):
Aborted (core dumped)
```



```
os: Linux-3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018
nodename: localhost.localdomain
machine: x86_64
clock source: unix
pcre jit disabled
detected number of CPU cores: 1
current working directory: /myblog/myblog
detected binary path: /opt/python36/bin/uwsgi
uWSGI running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
chdir() to /myblog/myblog/
your processes number limit is 7183
your memory page size is 4096 bytes
detected max file descriptor number: 1024
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
uWSGI http bound on 0.0.0.0:9000 fd 4
uwsgi socket 0 bound to TCP address 127.0.0.1:40203 (port auto-assigned) fd 3
uWSGI running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
Python version: 3.6.6 (default, Jul  3 2020, 03:51:29)  [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]
Set PythonHome to /root/Envs/myblog
Fatal Python error: Py_Initialize: Unable to get the locale encoding
ModuleNotFoundError: No module named 'encodings'

Current thread 0x00007fcea40a2880 (most recent call first):
Aborted (core dumped)
(myblog) [root@localhost myblog]#
```

出现的原因：

```
因为这里使用的时centos自带的python2，不是我指定的python3，需要在创建虚拟环境的时候指定使用python3
所以需要修改默认的python解释器
参考链接：https://www.jianshu.com/p/1193b1dbaf95
但是在换成解释器后，再创建虚拟环境的时候，会报错，所以我只能重新安装虚拟环境和虚拟环境virtualevewrap
然后创建虚拟环境，重新下载项目需要的安装包。
这时可以测试在虚拟环境下  查看    pyhont版本

上诉的解决方案：
```

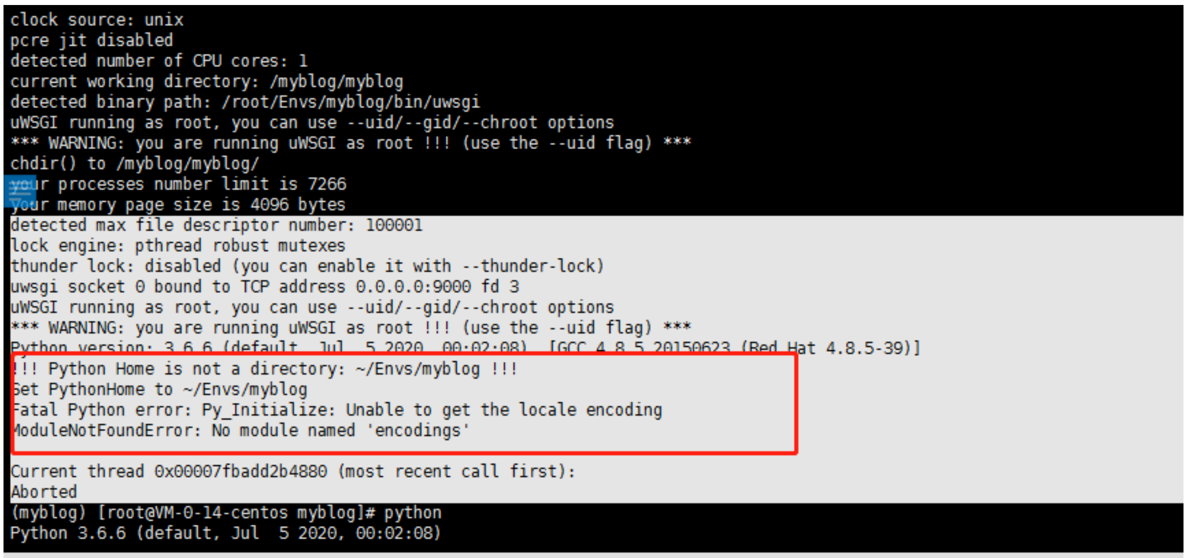


```
>>> exit()
(myblog) [root@localhost opt]# python
Python 3.6.6 (default, Jul  3 2020, 03:51:29)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-36)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

```
Fatal Python error: Py_Initialize: Unable to get the locale encoding
ModuleNotFoundError: No module named 'encodings'
```

也出现了这个错误：

```
detected max file descriptor number: 100001
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
uwsgi socket 0 bound to TCP address 0.0.0.0:9000 fd 3
uWSGI running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
Python version: 3.6.6 (default, Jul  5 2020, 00:02:08)  [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
!!! Python Home is not a directory: ~/Envs/myblog !!!
Set PythonHome to ~/Envs/myblog
Fatal Python error: Py_Initialize: Unable to get the locale encoding
ModuleNotFoundError: No module named 'encodings'

Current thread 0x00007fbadd2b4880 (most recent call first):
Aborted
```



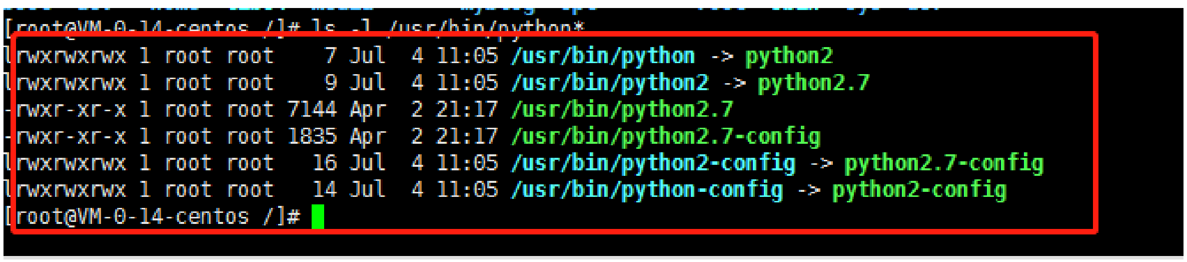
```
clock source: unix
pcre jit disabled
detected number of CPU cores: 1
current working directory: /myblog/myblog
detected binary path: /root/Envs/myblog/bin/uwsgi
uWSGI running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
chdir() to /myblog/myblog/
your processes number limit is 7266
your memory page size is 4096 bytes
detected max file descriptor number: 100001
lock engine: pthread robust mutexes
thunder lock: disabled (you can enable it with --thunder-lock)
uwsgi socket 0 bound to TCP address 0.0.0.0:9000 fd 3
uWSGI running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
Python version: 3.6.6 (default, Jul  5 2020, 00:02:08)  [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
!!! Python Home is not a directory: ~/Envs/myblog !!!
Set PythonHome to ~/Envs/myblog
Fatal Python error: Py_Initialize: Unable to get the locale encoding
ModuleNotFoundError: No module named 'encodings'

Current thread 0x00007fbadd2b4880 (most recent call first):
Aborted
(myblog) [root@VM-0-14-centos myblog]# python
Python 3.6.6 (default, Jul  5 2020, 00:02:08)
```

```
排查后发现可能是 虚拟环境的路径问题，修改为了
home            = /root/Envs/myblog
```

No module named 'encodings' 上诉的解决方案：

假如你的机器上有两个版本的python,如同时装有python2和python3,现在想要替换默认版本为python3,可以进行如下操作:

```
ls -l /usr/bin/python*
```



```
[root@VM-0-14-centos /]# ls -l /usr/bin/python*
lrwxrwxrwx 1 root root    7 Jul  4 11:05 /usr/bin/python -> python2
lrwxrwxrwx 1 root root    9 Jul  4 11:05 /usr/bin/python2 -> python2.7
-rwxr-xr-x 1 root root 7144 Apr  2 21:17 /usr/bin/python2.7
-rwxr-xr-x 1 root root 1835 Apr  2 21:17 /usr/bin/python2.7-config
lrwxrwxrwx 1 root root   16 Jul  4 11:05 /usr/bin/python2-config -> python2.7-config
lrwxrwxrwx 1 root root   14 Jul  4 11:05 /usr/bin/python-config -> python2-config
[root@VM-0-14-centos /]#
```

```
由图中文件的链接情况可知,在控制台中输入python和python2进入python2.7版本,
输入python3进入python3版本.
现在需要将第一个文件删除,然后把python3软链接到该目录下:
```

```
#查看python的链接
ls -l /usr/bin/python*

rm -rf /usr/bin/python   # 我的腾讯云不是删除 而是备份  cp  /usr/bin/python
python.bak ， 再删除这个python
ln -s /usr/local/bin/python3 /usr/bin/python
我的设置是：ln -s /opt/python36/bin/python3.6 /usr/bin/python
```

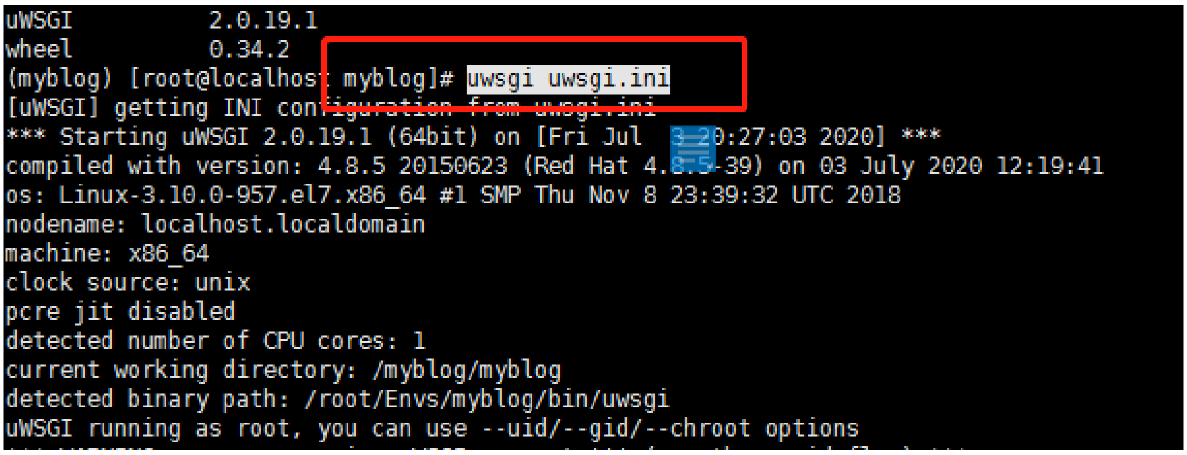
再次使用 uwsgi uwsgi.ini启动可能还会报encoding的错误，可能是python软链接的问题

```

```

接下来按照之前安装的教程就可以了。

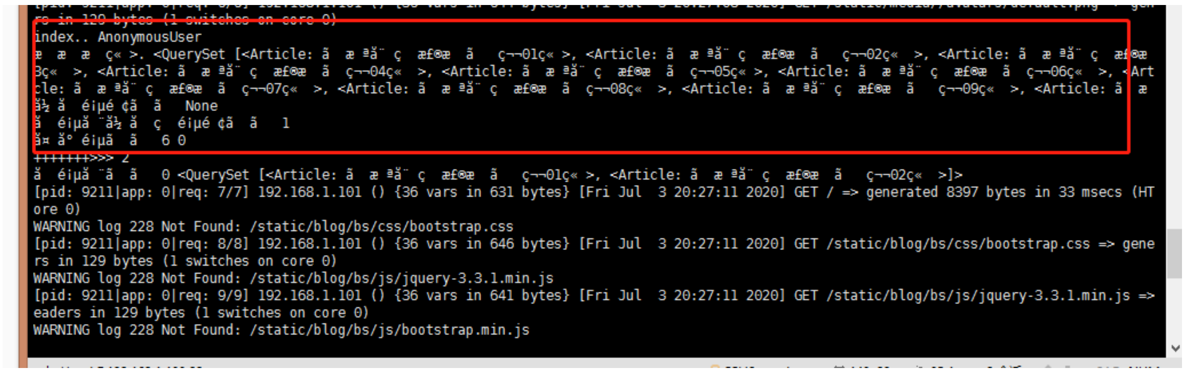
重新执行uwsgi 配置文件来启动django项目，可以执行成功，配置文件就按照上面的配置信息来

```
执行：uwsgi uwsgi.ini
```



```
uWSGI           2.0.19.1
wheel           0.34.2
(myblog) [root@localhost myblog]# uwsgi uwsgi.ini
[uWSGI] getting INI configuration from uwsgi.ini
*** Starting uWSGI 2.0.19.1 (64bit) on [Fri Jul  3 20:27:03 2020] ***
compiled with version: 4.8.5 20150623 (Red Hat 4.8.5-39) on 03 July 2020 12:19:41
os: Linux-3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018
nodename: localhost.localdomain
machine: x86_64
clock source: unix
pcre jit disabled
detected number of CPU cores: 1
current working directory: /myblog/myblog
detected binary path: /root/Envs/myblog/bin/uwsgi
uWSGI running as root, you can use --uid/--gid/--chroot options
```

但是发现在输出的日志信息碰到中文的时候会乱码，以后再处理这个问题吧

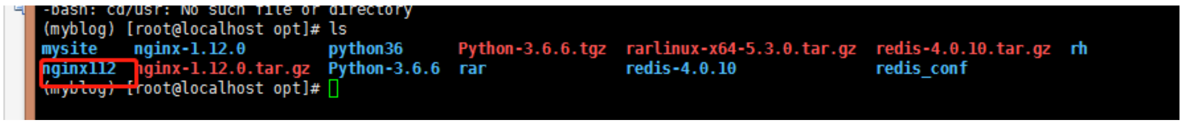


```
rs in 129 bytes (1 switches on core 0)
index.. AnonymousUser
æ  æ  æ  ç« >. <QuerySet [<Article: ã  æ ªă¨ ç  æ£®æ  ã   ç¬¬01ç« >, <Article: ã  æ ªă¨ ç  æ£®æ  ã   ç¬¬02ç«  >, <Article: ã  æ ªă¨ ç  æ£®æ
3ç«  >, <Article: ã  æ ªă¨ ç  æ£®æ  ã  ç¬¬04ç«  >, <Article: ã  æ ªă¨ ç  æ£®æ  ã  ç¬¬05ç« >, <Article: ã  æ ªă¨ ç  æ£®æ  ã  ç¬¬06ç«  >, <Art
cle: ã  æ ªă¨ ç  æ£®æ  ã  ç¬¬07ç«  >, <Article: ã  æ ªă¨ ç  æ£®æ  ã  ç¬¬08ç«  >, <Article: ã  æ ªă¨ ç  æ£®æ  ã   ç¬¬09ç«  >, <Article: ã  æ
ă½ ă  éiµé ¢ã  ã   None
ă  éiµă ¨ă½ ă  ç  éiµé ¢ã  ã   1
ă¤ ă° éiµã  ã   6 0
+++++++>>> 2
ă  éiµă ¨ã  ã   0 <QuerySet [<Article: ã  æ ªă¨ ç  æ£®æ  ã   ç¬¬01ç« >, <Article: ã  æ ªă¨ ç  æ£®æ  ã   ç¬¬02ç«  >]>
[pid: 9211|app: 0|req: 7/7] 192.168.1.101 () {36 vars in 631 bytes} [Fri Jul  3 20:27:11 2020] GET / => generated 8397 bytes in 33 msecs (HT
ore 0)
WARNING log 228 Not Found: /static/blog/bs/css/bootstrap.css
[pid: 9211|app: 0|req: 8/8] 192.168.1.101 () {36 vars in 646 bytes} [Fri Jul  3 20:27:11 2020] GET /static/blog/bs/css/bootstrap.css => gene
rs in 129 bytes (1 switches on core 0)
WARNING log 228 Not Found: /static/blog/bs/js/jquery-3.3.1.min.js
[pid: 9211|app: 0|req: 9/9] 192.168.1.101 () {36 vars in 641 bytes} [Fri Jul  3 20:27:11 2020] GET /static/blog/bs/js/jquery-3.3.1.min.js =>
eaders in 129 bytes (1 switches on core 0)
WARNING log 228 Not Found: /static/blog/bs/js/bootstrap.min.js
```

3、安装配置nginx实现nginx的代理,前提一定要在项目的settings.py 文件中加入

```
收集静态文件 ,修改settings.py配置文件， 加入以下一行配置
STATIC_ROOT='/opt/myblog/static'

并且要在项目根文件夹下  收集静态文件
```

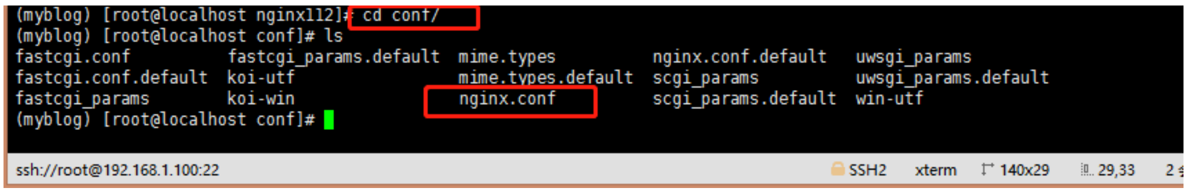
我已经安装了nginx，在/opt/nginx112



```
-bash: cd/usr: No such file or directory
(myblog) [root@localhost opt]# ls
mysite    nginx-1.12.0         python36      Python-3.6.6.tgz  rarlinux-x64-5.3.0.tar.gz  redis-4.0.10.tar.gz  rh
nginx112  nginx-1.12.0.tar.gz  Python-3.6.6  rar               redis-4.0.10               redis_conf
(myblog) [root@localhost opt]# 
```

nginx.conf

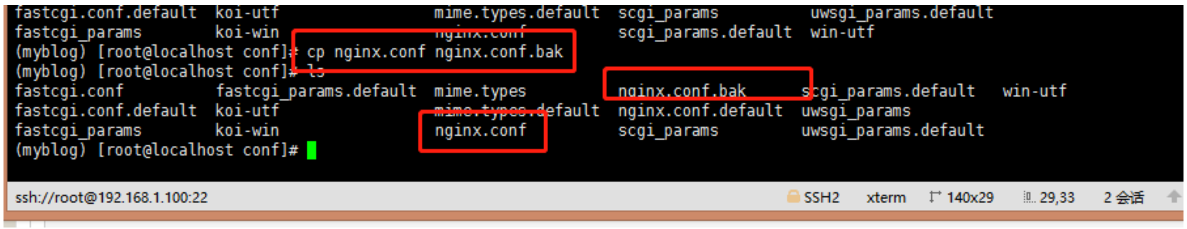
```
cd /opt/nginx112/conf
```



```
(myblog) [root@localhost nginx112]# cd conf/
(myblog) [root@localhost conf]# ls
fastcgi.conf          fastcgi_params.default  mime.types          nginx.conf.default   uwsgi_params
fastcgi.conf.default  koi-utf                 mime.types.default  scgi_params          uwsgi_params.default
fastcgi_params        koi-win                 nginx.conf          scgi_params.default  win-utf
(myblog) [root@localhost conf]# 
ssh://root@192.168.1.100:22          SSH2   xterm   140x29   29,33   2
```

先备份nginx.conf文件防止后续出错

```
cp nginx.conf nginx.conf.bak
```



```
fastcgi.conf.default  koi-utf                 mime.types.default  scgi_params          uwsgi_params.default
fastcgi_params        koi-win                 nginx.conf          scgi_params.default  win-utf
(myblog) [root@localhost conf]# cp nginx.conf nginx.conf.bak
(myblog) [root@localhost conf]# ls
fastcgi.conf          fastcgi_params.default  mime.types          nginx.conf.bak      scgi_params.default   win-utf
fastcgi.conf.default  koi-utf                 mime.types.default  nginx.conf.default  uwsgi_params
fastcgi_params        koi-win                 nginx.conf          scgi_params         uwsgi_params.default
(myblog) [root@localhost conf]# 
ssh://root@192.168.1.100:22          SSH2   xterm   140x29   29,33   2 会话
```

所以我的myblog的nginx配置文件是：uwsgi_pass 127.0.0.1:9000; #端口要和uwsgi里配置的一样

nginx.conf

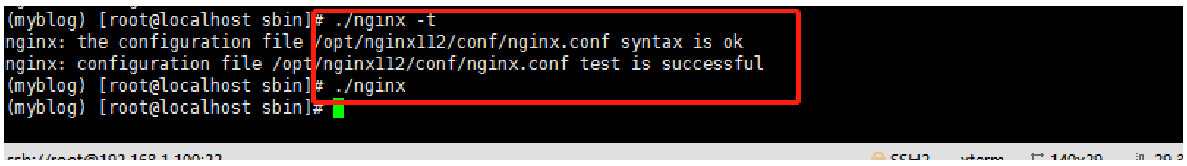
```
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    server {
        listen 80;
        server_name  192.168.1.100; #改为自己的域名，没域名修改为127.0.0.1:80
        charset utf-8;
        location / {
           include uwsgi_params;
           uwsgi_pass 127.0.0.1:9000;  #端口要和uwsgi里配置的一样
           uwsgi_param UWSGI_SCRIPT myblog.wsgi;  #wsgi.py所在的目录名+.wsgi
           uwsgi_param UWSGI_CHDIR /myblog/myblog/myblog/; #项目路径

        }
        location /static/ {
        alias /opt/myblog/static/; #静态资源路径
        }
    }
}
```

在项目的settings文件中需要写上项目的静态文件目录，当然也需要先收集静态文件到自己指定的static目录下，不然会出现502错误，配置后一定要重启nginx和uwsgi

配置好了后：

```
进入/usr/local/nginx/sbin/目录 (这个是自己的nginx安装目录)
执行 ./nginx -t 命令先检查配置文件是否有错，没有错就执行以下命令：
./nginx
```

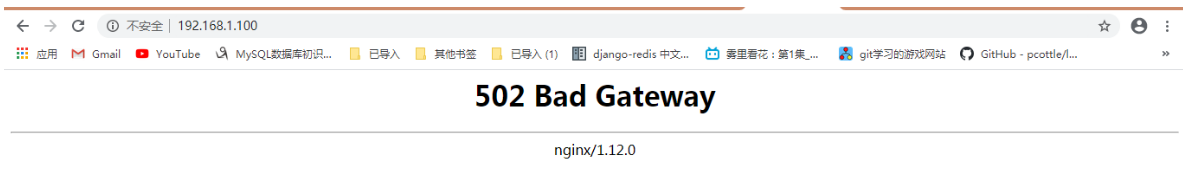
```
cd /opt/nginx112/sbin
./nginx #启动
./nginx -s stop #关闭
./nginx -s reload # 平滑重启 ，修改了nginx.conf之后，可以不重启服务，加载新的配置
或者  /opt/nginx112/sbin/nginx -s reload  # 绝对路径平滑重启
```



```
(myblog) [root@localhost sbin]# ./nginx -t
nginx: the configuration file /opt/nginx112/conf/nginx.conf syntax is ok
nginx: configuration file /opt/nginx112/conf/nginx.conf test is successful
(myblog) [root@localhost sbin]# ./nginx
(myblog) [root@localhost sbin]# 
```

然后在浏览器使用当前服务器的ip登陆，显示的geteway,可能是配置文件nginx.conf配置错误

```
192.168.1.100   # 不需要加8000端口
```



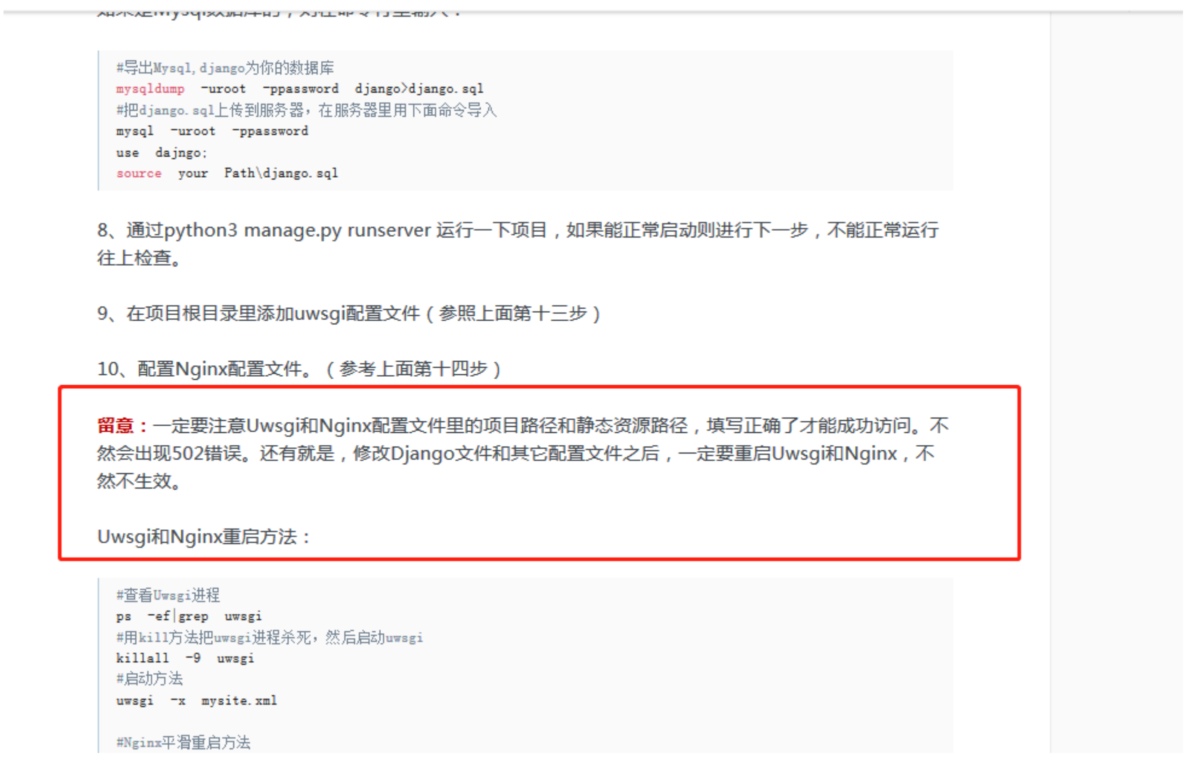
502 Bad Gateway

nginx/1.12.0

ps特别提醒:一直出现502页面，是因为，静态文件的路径的问题和，没有重启uwsgi.ini 文件和重启nginx，

在运行项目的时候，一定要确认uwsgi和nignx都是启动了的，否则会一直出现502的页面，这个问题我搞了好久,

还要将项目的settings.py的debug修改为false, Debug=False,host也需要修改为 *



```
#导出Mysql,django为你的数据库
mysqldump  -uroot  -ppassword  django>django.sql
#把django.sql上传到服务器，在服务器里用下面命令导入
mysql  -uroot  -ppassword
use  dajngo;
source  your  Path\django.sql
```

8、通过python3 manage.py runserver 运行一下项目，如果能正常启动则进行下一步，不能正常运行往上检查。

9、在项目根目录里添加uwsgi配置文件（参照上面第十三步）

10、配置Nginx配置文件。（参考上面第十四步）

留意：一定要注意Uwsgi和Nginx配置文件里的项目路径和静态资源路径，填写正确了才能成功访问。不然会出现502错误。还有就是，修改Django文件和其它配置文件之后，一定要重启Uwsgi和Nginx，不然不生效。

Uwsgi和Nginx重启方法：

```
#查看Uwsgi进程
ps  -ef|grep  uwsgi
#用kill方法把uwsgi进程杀死，然后启动uwsgi
killall  -9  uwsgi
#启动方法
uwsgi  -x  mysite.xml

#Nginx平滑重启方法
```

终端没有任何提示就证明nginx启动成功。可以使用你的服务器地址查看，成功之后就会看到一个nginx欢迎页面。

之后，在settings.py里设置：

1、关闭DEBUG模式。

DEBUG = False

2、ALLOWED_HOSTS设置为* 表示任何IP都可以访问网站。

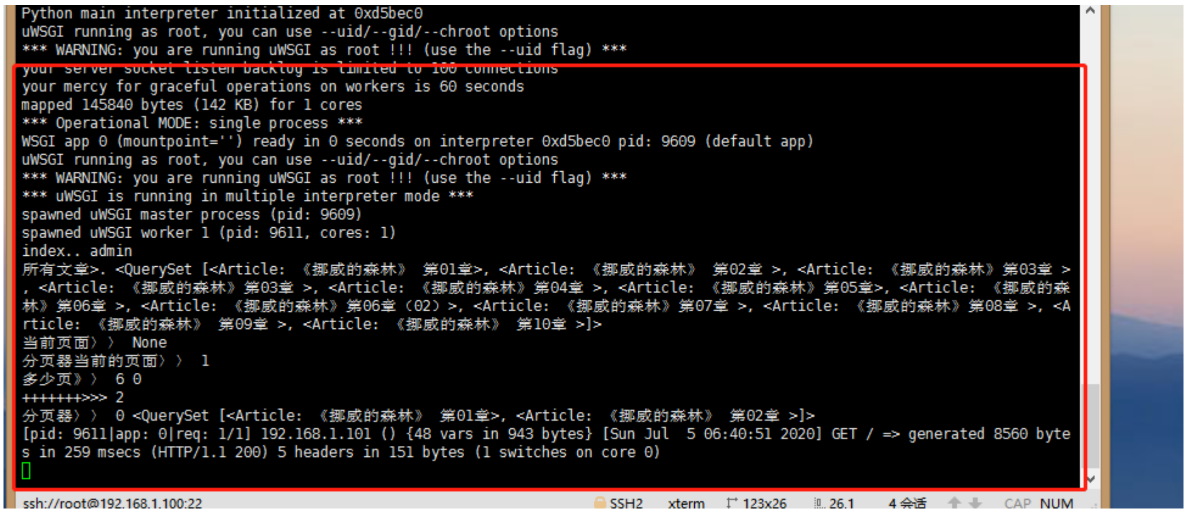
ALLOWED_HOSTS = ['*']

```
重启uwsgi和nginx
进入 自己存放uwsgi的目录，使用
uwsgi  uwsgi.ini  命令启动uwsgi，
然后进入nginx的安装目录下的sbin 目录，使用  ./nginx -t 先检查nginx.conf配置文件是否有
误，如果没有报错提示，则：
./nginx 启动
或则：
./nginx -s reload 平滑重启
```

nginx.conf的配置文件的端口一定要和uwsgi.ini的配置文件 的端口一致

```
nginx.conf的配置文件的端口一定要和uwsgi.ini的配置文件 的端口一致
```

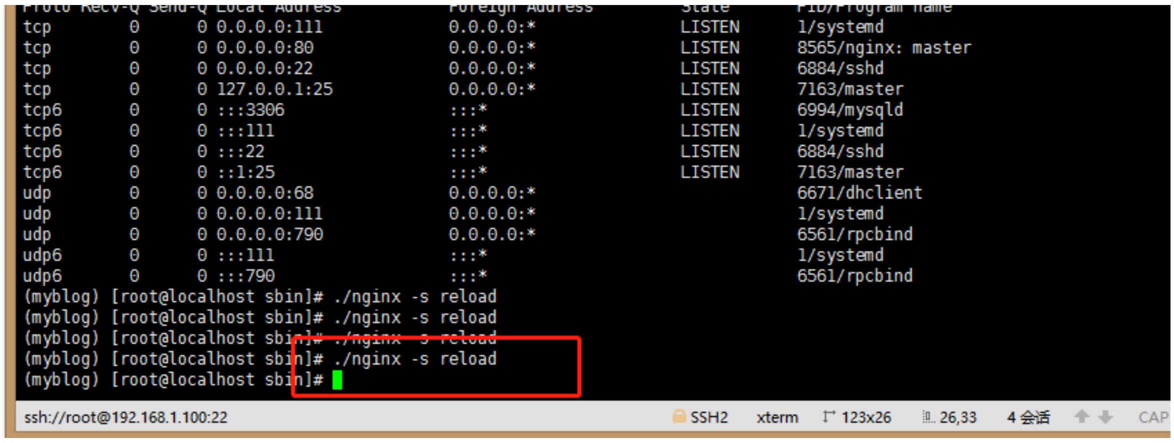
uwsgi启动页面：



```
Python main interpreter initialized at 0xd5bec0
uWSGI running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
your server socket listen backlog is limited to 100 connections
your mercy for graceful operations on workers is 60 seconds
mapped 145840 bytes (142 KB) for 1 cores
*** Operational MODE: single process ***
WSGI app 0 (mountpoint='') ready in 0 seconds on interpreter 0xd5bec0 pid: 9609 (default app)
uWSGI running as root, you can use --uid/--gid/--chroot options
*** WARNING: you are running uWSGI as root !!! (use the --uid flag) ***
*** uWSGI is running in multiple interpreter mode ***
spawned uWSGI master process (pid: 9609)
spawned uWSGI worker 1 (pid: 9611, cores: 1)
index.. admin
所有文章>. <QuerySet [<Article: 《挪威的森林》 第01章>, <Article: 《挪威的森林》 第02章 >, <Article: 《挪威的森林》第03章 >
, <Article: 《挪威的森林》第03章 >, <Article: 《挪威的森林》第04章 >, <Article: 《挪威的森林》第05章>, <Article: 《挪威的森
林》第06章 >, <Article: 《挪威的森林》第06章 (02) >, <Article: 《挪威的森林》第07章 >, <Article: 《挪威的森林》第08章 >, <A
rticle: 《挪威的森林》 第09章 >, <Article: 《挪威的森林》 第10章 >]>
当前页面〉〉 None
分页器当前的页面〉〉 1
多少页》〉 6 0
+++++++>>> 2
分页器〉〉 0 <QuerySet [<Article: 《挪威的森林》 第01章>, <Article: 《挪威的森林》 第02章 >]>
[pid: 9611|app: 0|req: 1/1] 192.168.1.101 () {48 vars in 943 bytes} [Sun Jul  5 06:40:51 2020] GET / => generated 8560 byte
s in 259 msecs (HTTP/1.1 200) 5 headers in 151 bytes (1 switches on core 0)
```

ssh://root@192.168.1.100:22
SSH2 xterm 123x26 26,1 4 会话 CAP NUM

nginx 启动页面：



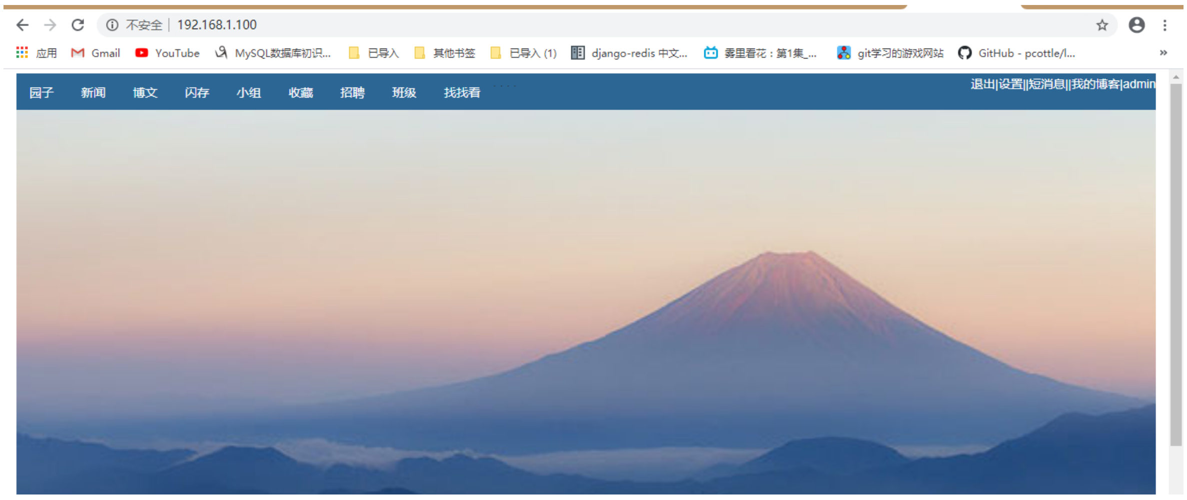
```
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      8565/nginx: master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      6884/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      7163/master
tcp6       0      0 :::3306                 :::*                    LISTEN      6994/mysqld
tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
tcp6       0      0 :::22                   :::*                    LISTEN      6884/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      7163/master
udp        0      0 0.0.0.0:68              0.0.0.0:*                           6671/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1/systemd
udp        0      0 0.0.0.0:790             0.0.0.0:*                           6561/rpcbind
udp6       0      0 :::111                  :::*                                1/systemd
udp6       0      0 :::790                  :::*                                6561/rpcbind
(myblog) [root@localhost sbin]# ./nginx -s reload
(myblog) [root@localhost sbin]# ./nginx -s reload
(myblog) [root@localhost sbin]# ./nginx -s reload
(myblog) [root@localhost sbin]# ./nginx -s reload
(myblog) [root@localhost sbin]#
```

ssh://root@192.168.1.100:22
SSH2 xterm 123x26 26,33 4 会话 CAP

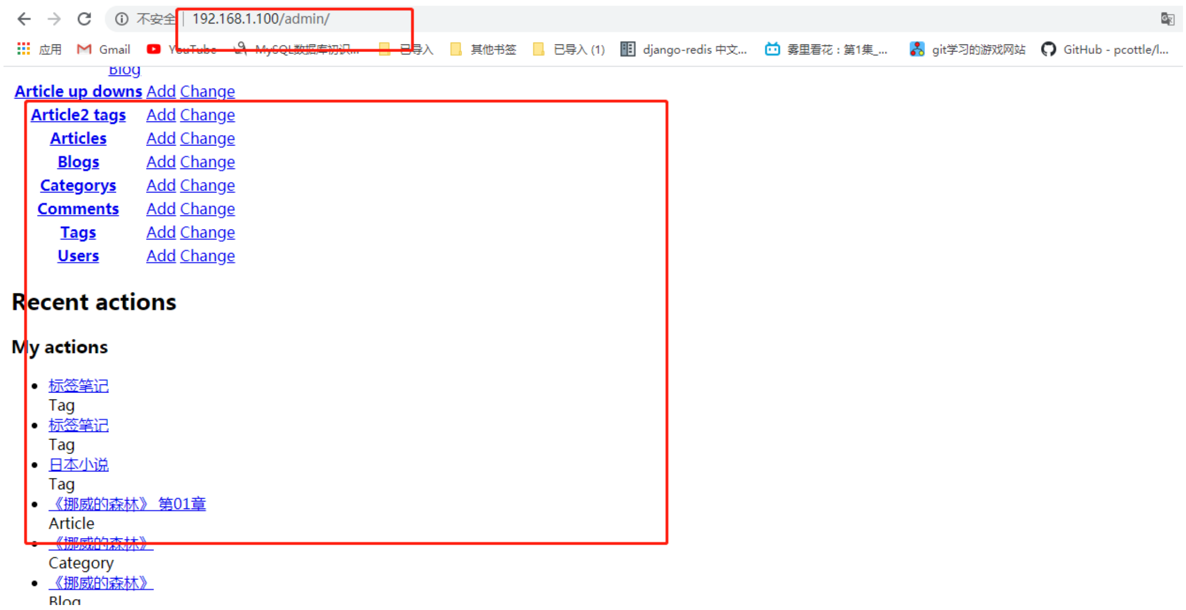
使用电脑的ip 访问

```
192.168.1.100
```

前端访问页面：


不安全 | 192.168.1.100
应用 Gmail YouTube MySQL数据库初识... 已导入 其他书签 已导入 (1) django-redis 中文... 雾里看花：第1集_... git学习的游戏网站 GitHub - pcottle/l...
园子 新闻 博文 闪存 小组 收藏 招聘 班级 找找看
退出|设置||短消息||我的博客|admin

可以正常启动后，访问前端会出现访问admin页面的时候，admin页面没有样式，就像这样


不安全 | 192.168.1.100/admin/
应用 Gmail YouTube MySQL数据库初识... 已导入 其他书签 已导入 (1) django-redis 中文... 雾里看花：第1集_... git学习的游戏网站 GitHub - pcottle/l...

Blog
Article up downs Add Change
Article2 tags Add Change
Articles Add Change
Blogs Add Change
Categorys Add Change
Comments Add Change
Tags Add Change
Users Add Change

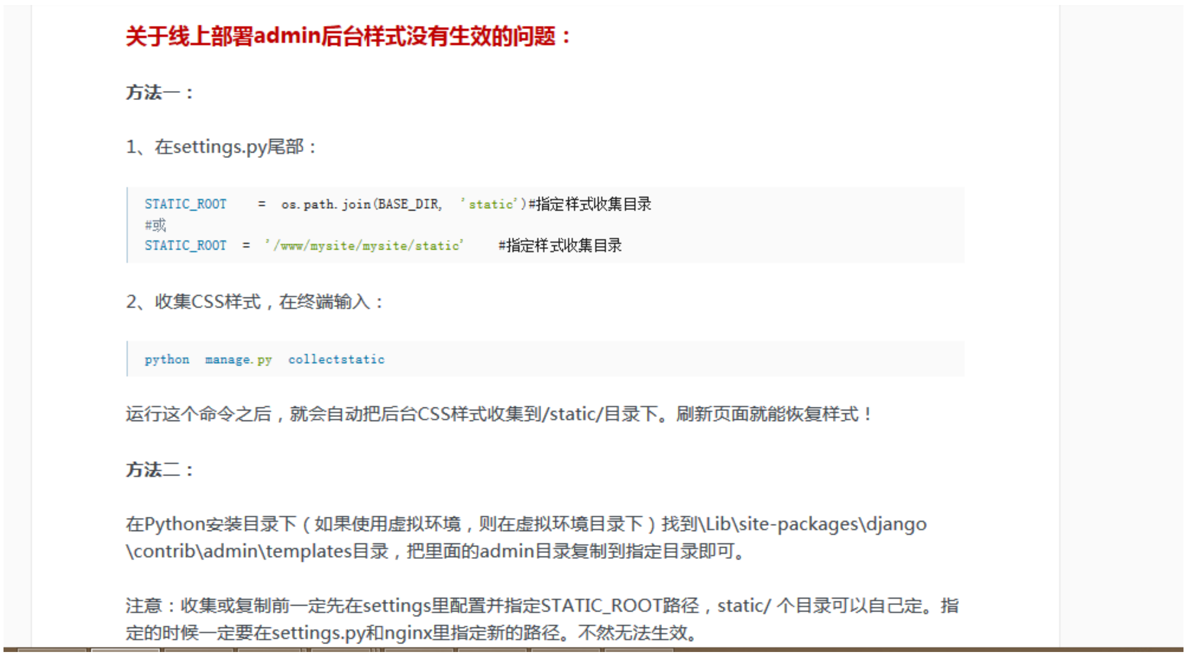
Recent actions

My actions

- 标签笔记
  Tag
- 标签笔记
  Tag
- 日本小说
  Tag
- 《挪威的森林》 第01章
  Article
- 《挪威的森林》
  Category
- 《挪威的森林》
  Blog

解决方案：

网上的解决方案：



关于线上部署admin后台样式没有生效的问题：

方法一：

1、在settings.py尾部：

```
STATIC_ROOT  =  os.path.join(BASE_DIR,  'static')#指定样式收集目录
#或
STATIC_ROOT  =  '/www/mysite/mysite/static'     #指定样式收集目录
```

2、收集CSS样式，在终端输入：

```
python  manage.py  collectstatic
```

运行这个命令之后，就会自动把后台CSS样式收集到/static/目录下。刷新页面就能恢复样式！

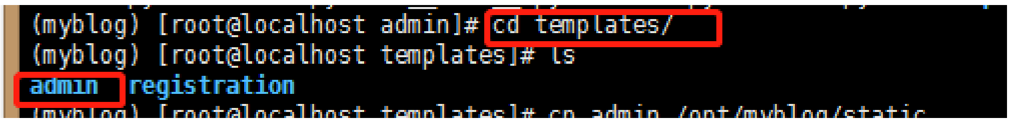
方法二：

在Python安装目录下（如果使用虚拟环境，则在虚拟环境目录下）找到\Lib\site-packages\django\contrib\admin\templates目录，把里面的admin目录复制到指定目录即可。

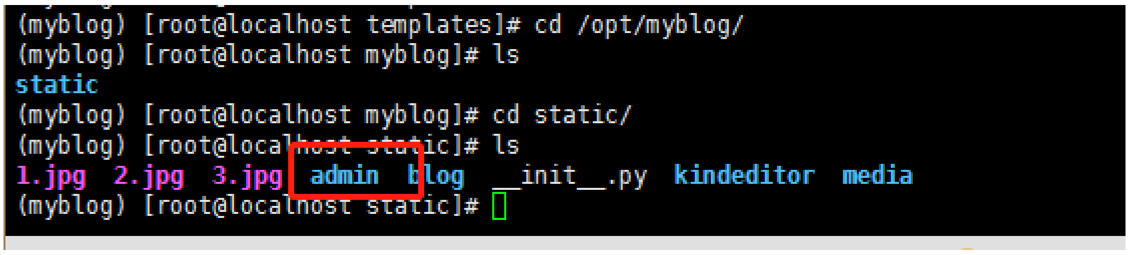
注意：收集或复制前一定先在settings里配置并指定STATIC_ROOT路径，static/ 个目录可以自己定。指定的时候一定要在settings.py和nginx里指定新的路径。不然无法生效。

将/opt/python36/lib/python3.6/site-packages/django/contrib/admin/templates/ 下面的这个admin文件夹复制到之前专门收集静态文件的目录。我的时/opt/myblog/static

```
cd /opt/python36/lib/python3.6/site-packages/django/contrib/admin/templates/

然后   cp -r admin /opt/myblog/static
```



```
(myblog) [root@localhost admin]# cd templates/
(myblog) [root@localhost templates]# ls
admin  registration
```

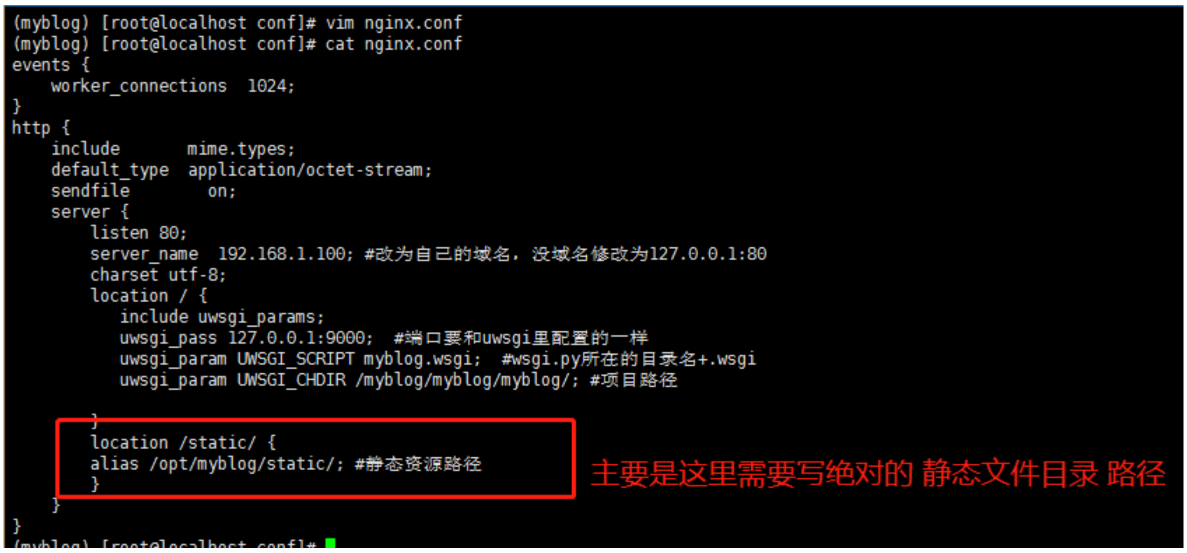


```
(myblog) [root@localhost templates]# cd /opt/myblog/
(myblog) [root@localhost myblog]# ls
static
(myblog) [root@localhost myblog]# cd static/
(myblog) [root@localhost static]# ls
1.jpg  2.jpg  3.jpg  admin  blog  __init__.py  kindeditor  media
(myblog) [root@localhost static]# 
```

之后可能会需要修改nginx.conf

完整版

```
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    server {
        listen 80;
        server_name  192.168.1.100; #改为自己的域名，没域名修改为127.0.0.1:80
        charset utf-8;
        location / {
           include uwsgi_params;
           uwsgi_pass 127.0.0.1:9000;  #端口要和uwsgi里配置的一样
           uwsgi_param UWSGI_SCRIPT myblog.wsgi;  #wsgi.py所在的目录名+.wsgi
           uwsgi_param UWSGI_CHDIR /myblog/myblog/myblog/; #项目路径

        }
        location /static/ {
        alias /opt/myblog/static/; #静态资源路径
        }
    }
}
```



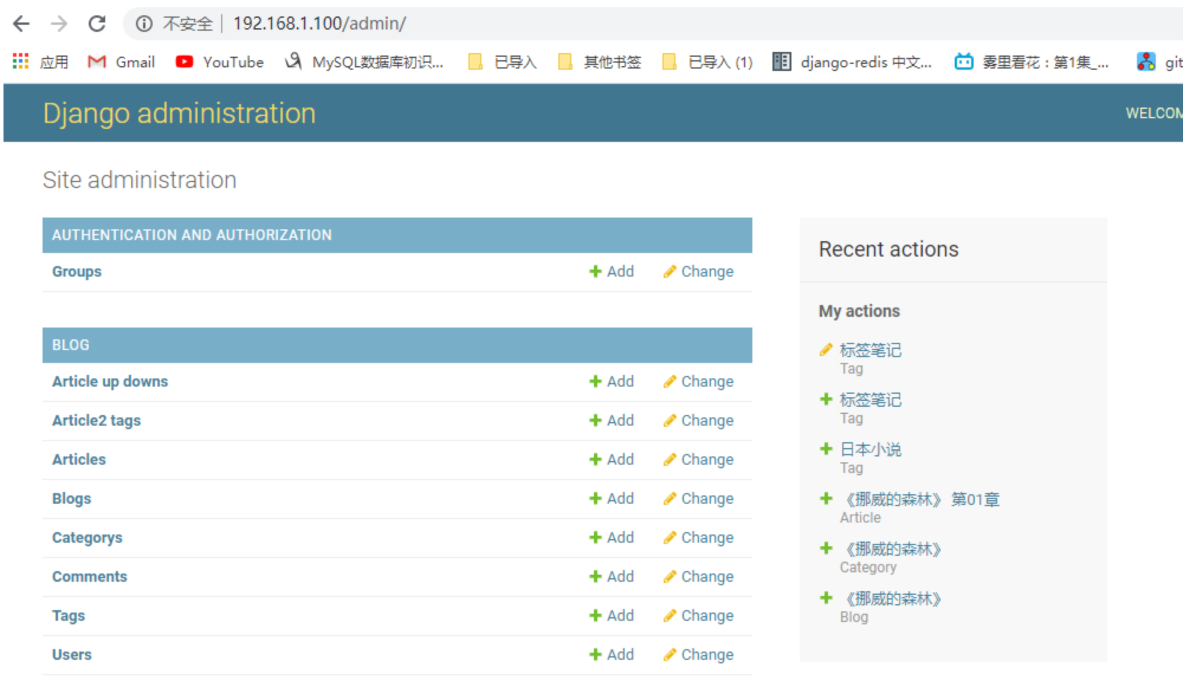
```
(myblog) [root@localhost conf]# vim nginx.conf
(myblog) [root@localhost conf]# cat nginx.conf
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    server {
        listen 80;
        server_name  192.168.1.100; #改为自己的域名，没域名修改为127.0.0.1:80
        charset utf-8;
        location / {
           include uwsgi_params;
           uwsgi_pass 127.0.0.1:9000;  #端口要和uwsgi里配置的一样
           uwsgi_param UWSGI_SCRIPT myblog.wsgi;  #wsgi.py所在的目录名+.wsgi
           uwsgi_param UWSGI_CHDIR /myblog/myblog/myblog/; #项目路径

        }
        location /static/ {
        alias /opt/myblog/static/; #静态资源路径
        }
    }
}
```

然后重启nginx和uwsgi，使用uwsgi和nginx部署后，访问网站不需要加端口，直接ip访问就可以了

```
uwsgi -ini uwsgi.ini
./nginx -s reload
 重启之后就可以了
```

如果直接使用ip无法访问，显示是502 Bad Gateway或则访问的时候没有显示样式，则需要参考下面信息

```

有时配置好uwsgi.ini后，前端
#如果你使用了nginx，做反向代理，必须填写socket链接，而不是http参数
socket          = 0.0.0.0:8000
#如果你不用nginx，直接使用uwsgi，运行一个http服务端，就用这个http参数
http = 0.0.0.0:8000
```


不安全 | 192.168.1.100/admin/

应用 Gmail YouTube MySQL数据库初识... 已导入 其他书签 已导入 (1) django-redis 中文... 雾里看花：第1集_... git

Django administration

WELCOM

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	Add	Change

BLOG		
Article up downs	Add	Change
Article2 tags	Add	Change
Articles	Add	Change
Blogs	Add	Change
Categorys	Add	Change
Comments	Add	Change
Tags	Add	Change
Users	Add	Change

Recent actions

My actions

标签笔记
Tag

标签笔记
Tag

日本小说
Tag

《挪威的森林》 第01章
Article

《挪威的森林》
Category

《挪威的森林》
Blog

博客项目的nginx.conf 和uwsgi的配置文件内容

uwsgi.ini

```
[uwsgi]
chdir           = /myblog/myblog/
module          = myblog.wsgi
home            = /root/Envs/myblog
master          = true
processes       = 1
 socket          = 0.0.0.0:9000
# http            = 0.0.0.0:9000
vacuum          = true
~
```

nginx.conf

```
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    server {
        listen 80;
        server_name  134.175.176.119; #改为自己的域名，没域名修改为127.0.0.1:80
        charset utf-8;
        location / {
           include uwsgi_params;
           uwsgi_pass 134.175.176.119:9000;  #端口要和uwsgi里配置的一样
           uwsgi_param UWSGI_SCRIPT myblog.wsgi;  #wsgi.py所在的目录名+.wsgi
           uwsgi_param UWSGI_CHDIR /myblog/myblog/myblog/; #项目路径

        }
        location /static/ {
```

```
        alias /opt/myblog/static/; #静态资源路径
        }
    }
}
```

三、使项目在云服务器后台常驻，nohup

```
ps最最重要的步骤,当我将项目部署完成后 ，使用xshell进行nginx和uwsgi启动后，在xshell终端界面运行的时候访问是没有问题的，但是关闭终端后，再去访问就访问不到了，我不能一直开着电脑啊；所以现在使用nohup命令进行后台运行，即使关闭xshell，也能访问到项目
```

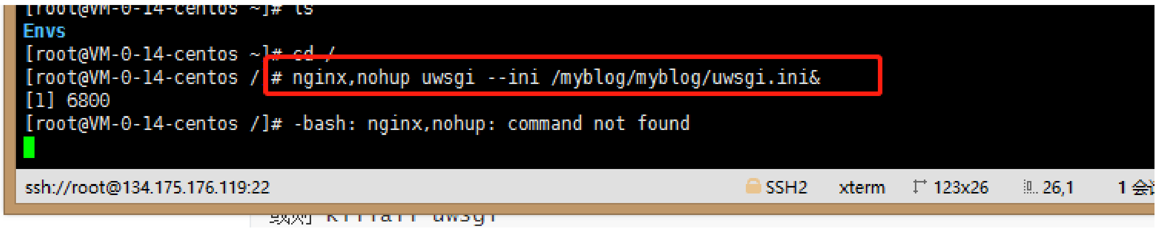
01、使用方法：

```
cd nginx的的启动文件夹
先启动nginx，为了防止说抵制使用，先kill所有uwsgi和nginx的pid
```

02、启动nginx

```
启动先查看配置文件是否填写正常
./nginx -t  # 如果没报错，则启动正常
./nginx  # 启动nginx，./nginx -s reload  平滑重启nginx，就算修改配置文件也不会断掉
```

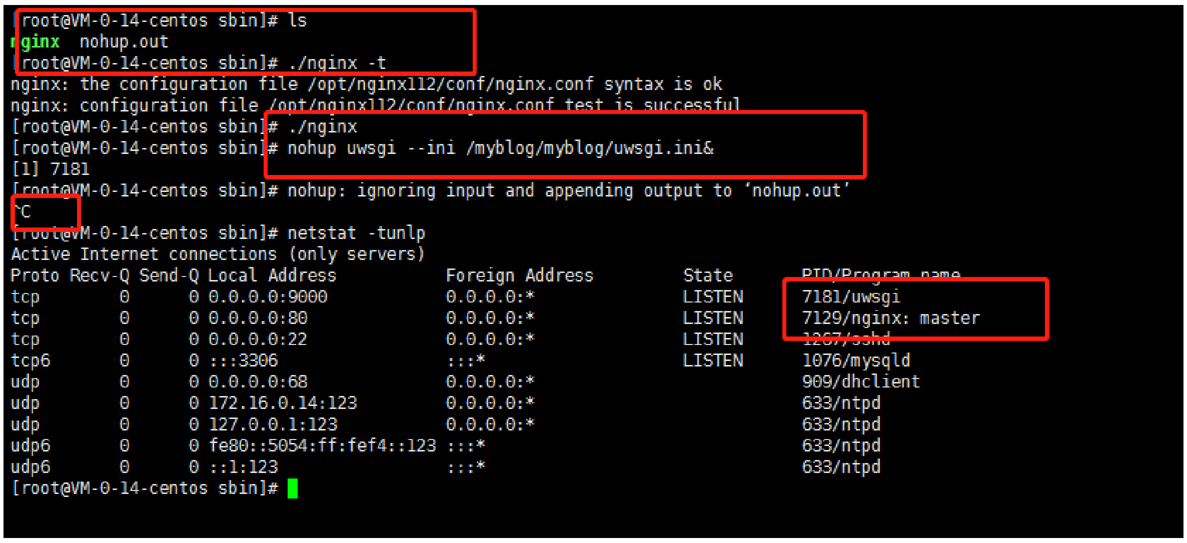
直接在root终端使用 nginx,nohup uwsgi --ini /myblog/myblog/uwsgi.ini&   无法直接启动项目



```
[root@VM-0-14-centos ~]# ls
Envs
[root@VM-0-14-centos ~]# cd /
[root@VM-0-14-centos /]# nginx,nohup uwsgi --ini /myblog/myblog/uwsgi.ini&
[1] 6800
[root@VM-0-14-centos /]# -bash: nginx,nohup: command not found
```

ssh://root@134.175.176.119:22   SSH2   xterm   123x26   26,1   1 会

所以我选择先进入nginx目录，先启动nginx，再启动uwsgi，因为我知道自己的uwsi.ini 的文件路径

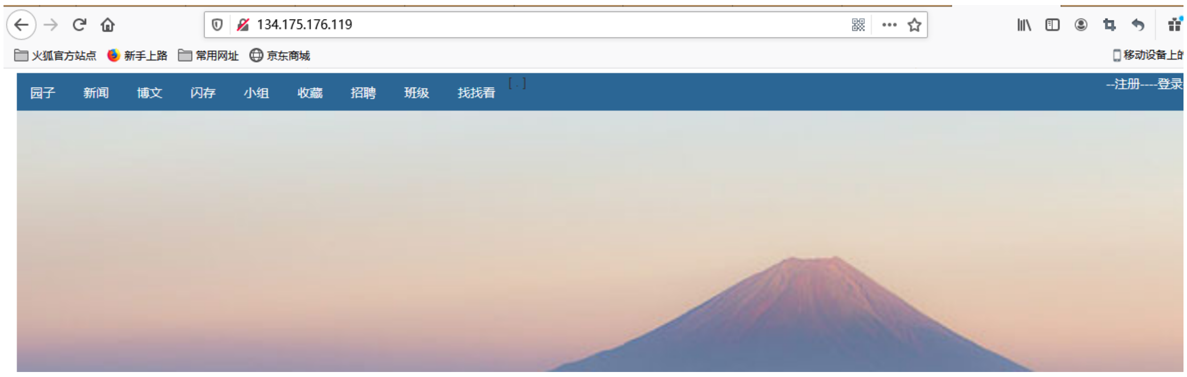


```
[root@VM-0-14-centos sbin]# ls
nginx  nohup.out
[root@VM-0-14-centos sbin]# ./nginx -t
nginx: the configuration file /opt/nginx112/conf/nginx.conf syntax is ok
nginx: configuration file /opt/nginx112/conf/nginx.conf test is successful
[root@VM-0-14-centos sbin]# ./nginx
[root@VM-0-14-centos sbin]# nohup uwsgi --ini /myblog/myblog/uwsgi.ini&
[1] 7181
[root@VM-0-14-centos sbin]# nohup: ignoring input and appending output to 'nohup.out'
^C
[root@VM-0-14-centos sbin]# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:9000            0.0.0.0:*               LISTEN      7181/uwsgi
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      7129/nginx: master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1267/sshd
tcp6       0      0 :::3306                 :::*                    LISTEN      1076/mysqld
udp        0      0 0.0.0.0:68              0.0.0.0:*                           909/dhclient
udp        0      0 172.16.0.14:123         0.0.0.0:*                           633/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           633/ntpd
udp6       0      0 fe80::5054:ff:fef4::123 :::*                                633/ntpd
udp6       0      0 ::1:123                 :::*                                633/ntpd
[root@VM-0-14-centos sbin]#
```

3、使用nohup进行后台常驻

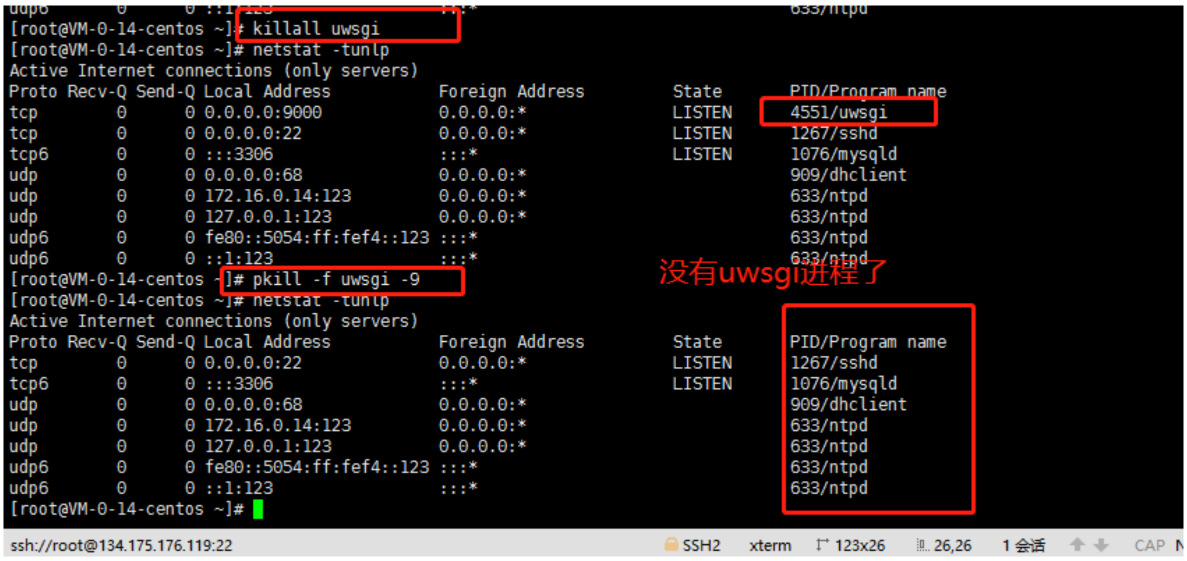
```
先进入nginx启动目录启动nginx，在按照下面的命令启动uwsig.ini，
nohup uwsgi --ini /myblog/myblog/uwsgi.ini&        # 后面是自己配置的uwsgi配置文件
的路径
```

关闭xshell然后自己的网址ip,访问ok



04、需要停止的话

```
netstat -tunlp    #查看所有的nginx和uwsgi的端口
不要启动的话, 就  kill  pid(进程id)  杀死进程

或则 killall uwsgi
     killall nginx

如果出现 kill 掉uwsgi进程, 则使用 pkill -f uwsgi -9  杀死
```



```
udp6       0      0 ::1:123                 :::*                    633/ntpd
[root@VM-0-14-centos ~]# killall uwsgi
[root@VM-0-14-centos ~]# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:9000            0.0.0.0:*               LISTEN      4551/uwsgi
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1267/sshd
tcp6       0      0 :::3306                 :::*                    LISTEN      1076/mysqld
udp        0      0 0.0.0.0:68              0.0.0.0:*                           909/dhclient
udp        0      0 172.16.0.14:123         0.0.0.0:*                           633/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           633/ntpd
udp6       0      0 fe80::5054:ff:fef4::123 :::*                                633/ntpd
udp6       0      0 ::1:123                 :::*                                633/ntpd
[root@VM-0-14-centos ~]# pkill -f uwsgi -9
[root@VM-0-14-centos ~]# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1267/sshd
tcp6       0      0 :::3306                 :::*                    LISTEN      1076/mysqld
udp        0      0 0.0.0.0:68              0.0.0.0:*                           909/dhclient
udp        0      0 172.16.0.14:123         0.0.0.0:*                           633/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           633/ntpd
udp6       0      0 fe80::5054:ff:fef4::123 :::*                                633/ntpd
udp6       0      0 ::1:123                 :::*                                633/ntpd
[root@VM-0-14-centos ~]#
```

没有uwsgi进程了

四、网站后期是需要不断更新代码和功能的，可以选择使用github或则gitee，将代码放到上面，然后服务器直接使用git] clone，git pull进行不断的更新

每次修改代码后，更新到服务器上也很简单。在**虚拟环境**中并**进入项目目录**，依次（collectstatic 和 migrate 是可选的）执行以下命令：

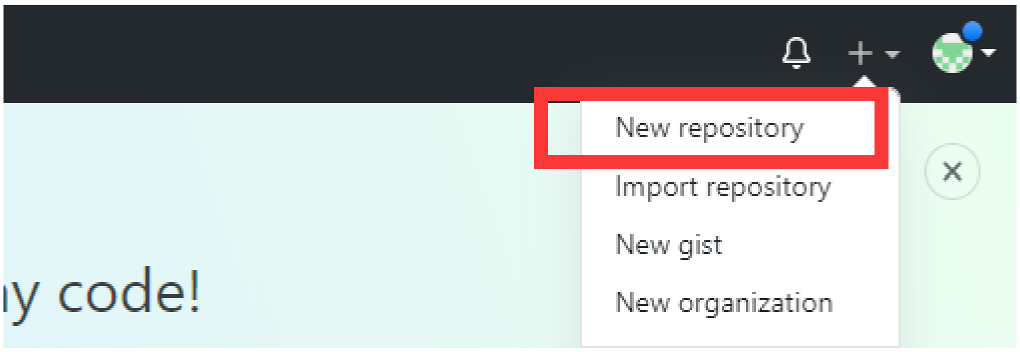
```
git pull origin master
python3 manage.py collectstatic
python3 manage.py migrate
# 重启 uwsgi
killall uwsgi
然后 uwsgi 启动命令
```

如果你更改了 Nginx 的配置文件，还需要重启 Nginx 服务：

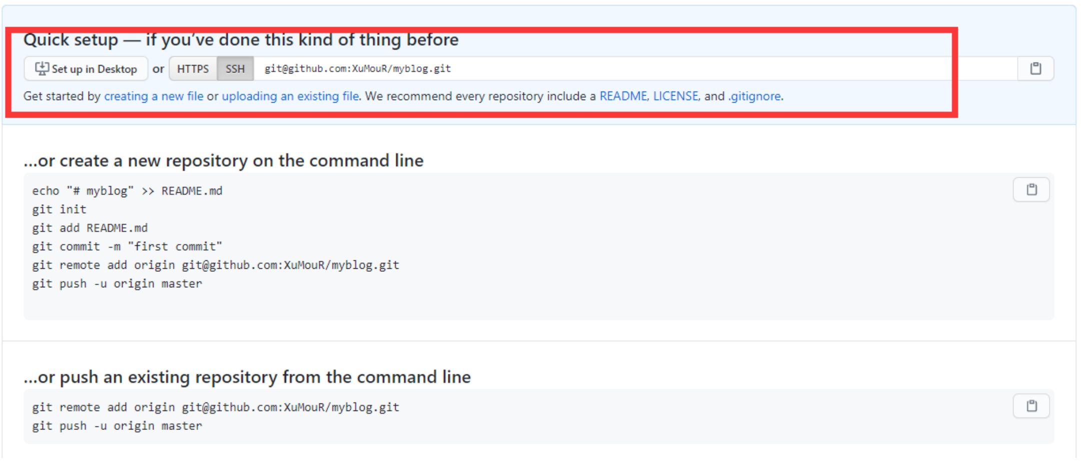
```
./nginx -s reload
```

五、使用github进行项目托管，并将项目上传到github私有仓库

第一步：在github创建私有仓库



然后创建选择在、私人仓库就可以了，然后会显示页面



Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH git@github.com:XuMouR/myblog.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# myblog" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:XuMouR/myblog.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:XuMouR/myblog.git
git push -u origin master
```

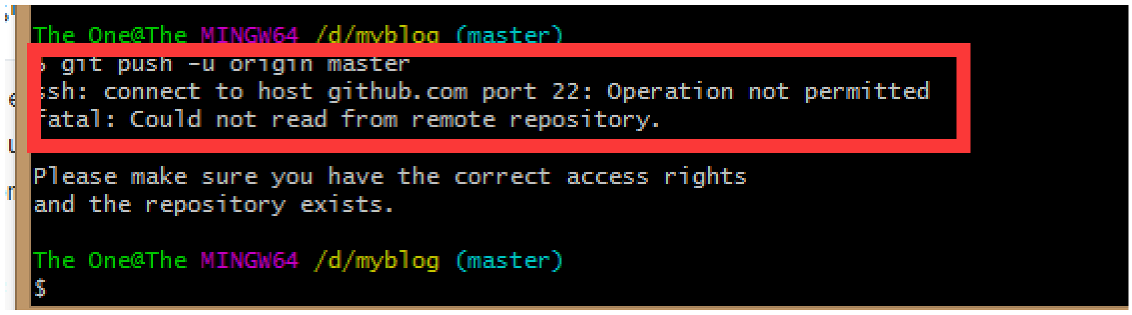
第二步、将本地的myblog项目进行上传

在项目文件中依次执行下面的命令

```
echo "# myblog" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:XuMouR/myblog.git
git push -u origin master
```

我在执行了 git remote add origin git@github.com:XuMouR/myblog.git 后报错

```
$ git push -u origin master
ssh: connect to host github.com port 22: Operation not permitted
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

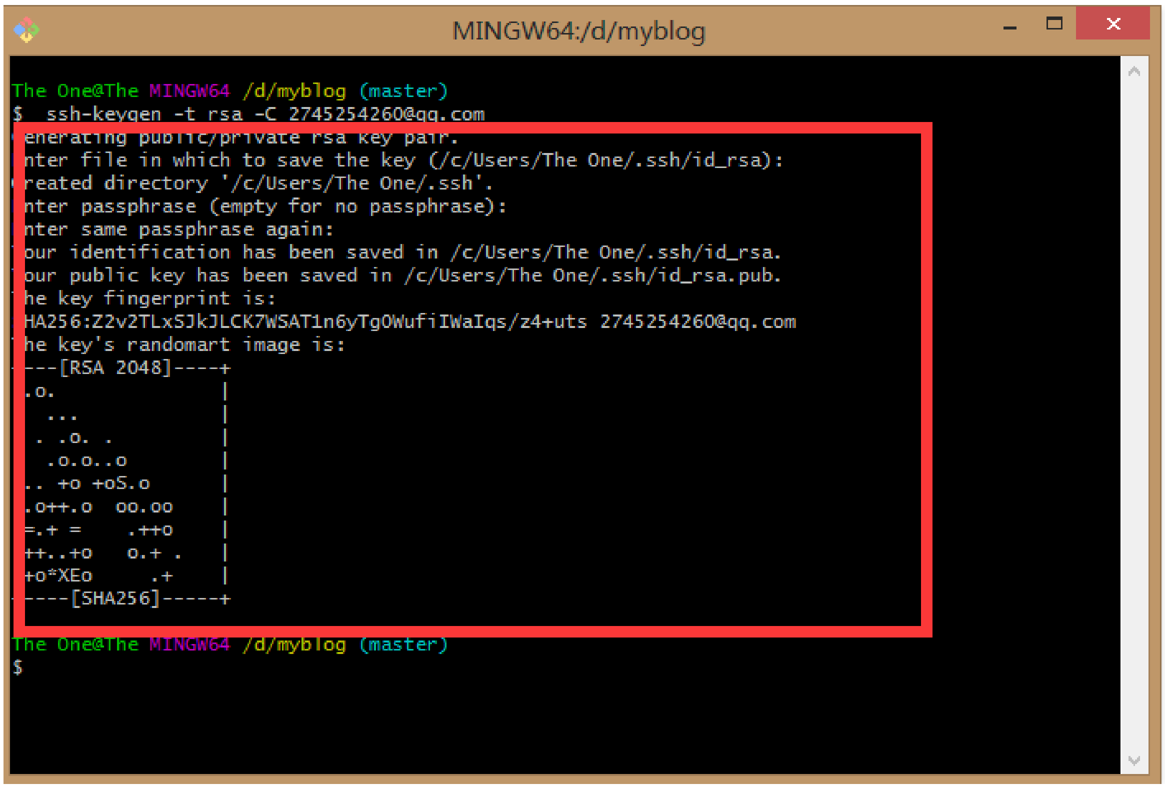


```
The One@The MINGW64 /d/myblog (master)
; git push -u origin master
;sh: connect to host github.com port 22: Operation not permitted
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

The One@The MINGW64 /d/myblog (master)
$
```

错误原因是因为我使用的时ssh，所以需要配置密钥

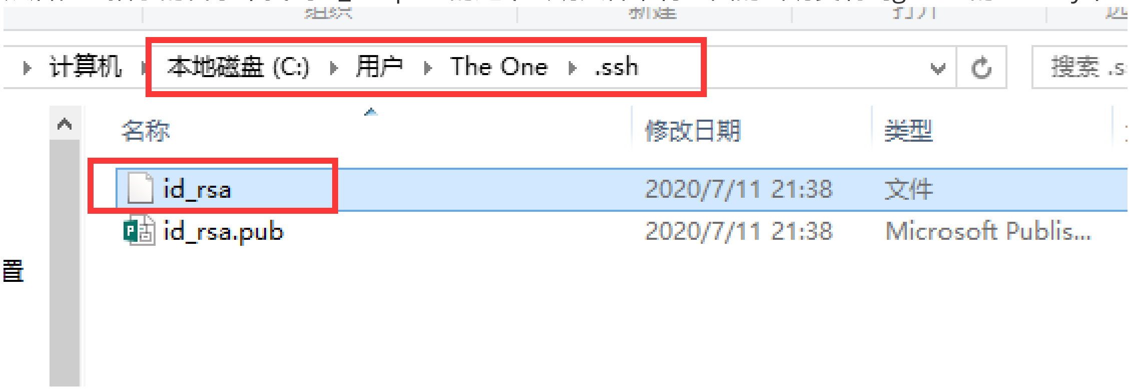
先生成密钥：直接用默认的就可以了

```
ssh-keygen -t rsa -C 2745254260@qq.com
```



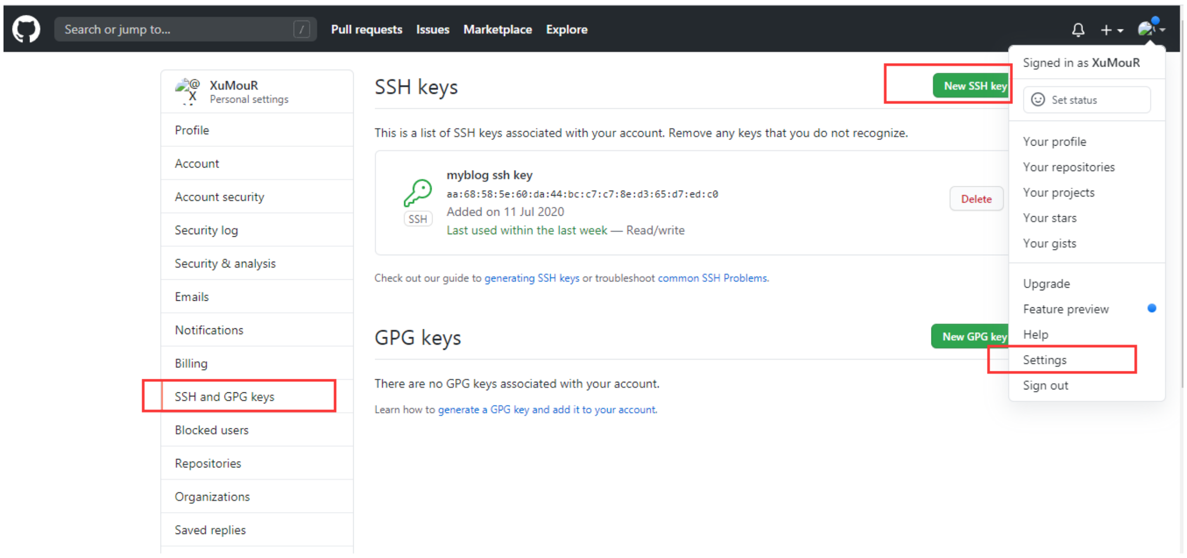
```
The One@The MINGW64 /d/myblog (master)
$  ssh-keygen -t rsa -C 2745254260@qq.com
enerating public/private rsa key pair.
nter file in which to save the key (/c/Users/The One/.ssh/id_rsa):
reated directory '/c/Users/The One/.ssh'.
nter passphrase (empty for no passphrase):
nter same passphrase again:
our identification has been saved in /c/Users/The One/.ssh/id_rsa.
our public key has been saved in /c/Users/The One/.ssh/id_rsa.pub.
he key fingerprint is:
HA256:Z2v2TLxSJkJLCK7WSAT1n6yTg0WufiIWaIqs/z4+uts 2745254260@qq.com
he key's randomart image is:
---[RSA 2048]----+
.o.             |
  ...           |
 . .o. .        |
  .o.o..o       |
.. +o +oS.o     |
.o++.o  oo.oo   |
=.+ =    .++o   |
++..+o   o.+ .  |
+o*XEo     .+   |
----[SHA256]-----+
The One@The MINGW64 /d/myblog (master)
$
```

然后在它指示的目录中找到id_rsa.pub 的这个公钥文件，将里面的密钥复制到github的ssh key中



名称	修改日期	类型
id_rsa	2020/7/11 21:38	文件
id_rsa.pub	2020/7/11 21:38	Microsoft Publis...

置



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

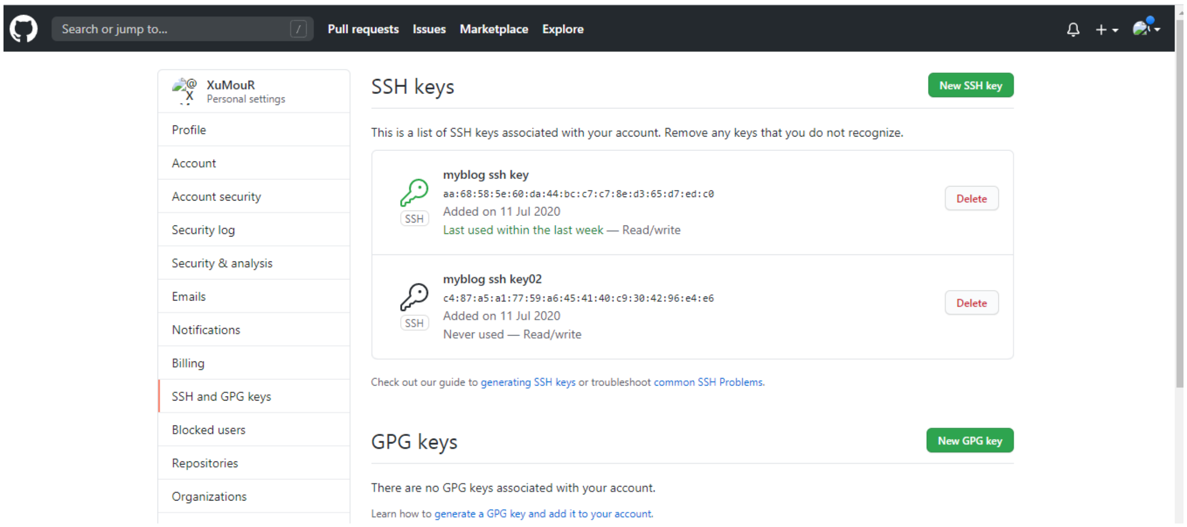
myblog ssh key
aa:68:58:5e:60:da:44:bc:c7:c7:8e:d3:65:d7:ed:c0
Added on 11 Jul 2020
Last used within the last week — Read/write

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

myblog ssh key
aa:68:58:5e:60:da:44:bc:c7:c7:8e:d3:65:d7:ed:c0
Added on 11 Jul 2020
Last used within the last week — Read/write

myblog ssh key02
c4:87:a5:a1:77:59:a6:45:41:40:c9:30:42:96:e4:e6
Added on 11 Jul 2020
Never used — Read/write

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

然后我再使用git push 发现还是上面的报错，就算换成https 也是一样的报错

```
$ git push -u origin master
ssh: connect to host github.com port 22: Operation not permitted
fatal: Could not read from remote repository.
```

最后实在找不到解决方案：

001、换成了https 来进行远程仓库链接

```
先查看先前的协议链接
$ git remote -v
origin  git@github.com:unlimitbladeworks/Data-Struts-Learning.git (fetch)
origin  git@github.com:unlimitbladeworks/Data-Struts-Learning.git (push)

# 移除远程仓库配置
git remote rm origin
```

002、重新添加远程仓库配置，使用https形式

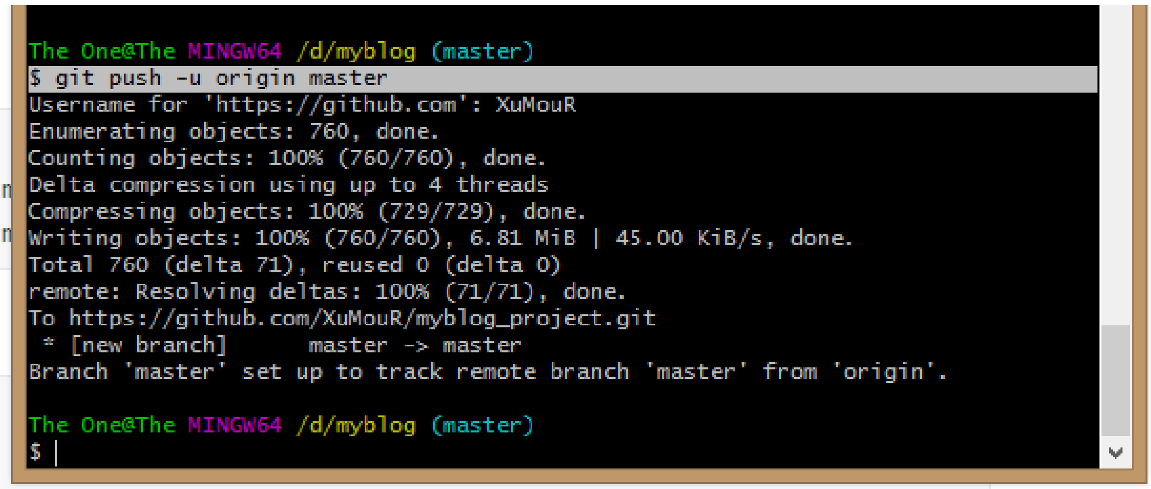
```
git remote add origin https://github.com/unlimitbladeworks/Data-Struts-
Learning.git
```

再次查看:

```
git remote -v
origin  https://github.com/unlimitbladeworks/Data-Struts-Learning.git (fetch)
origin  https://github.com/unlimitbladeworks/Data-Struts-Learning.git (push)
```

然后执行pull

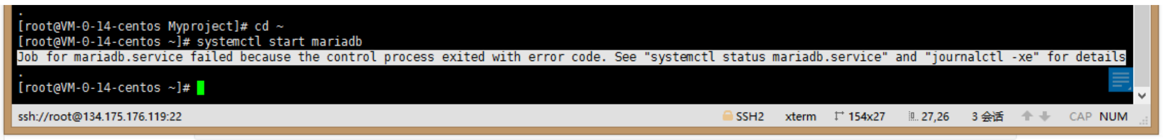
```
git pull
$ git push -u origin master

```



```
The One@The MINGW64 /d/myblog (master)
$ git push -u origin master
Username for 'https://github.com': XuMouR
Enumerating objects: 760, done.
Counting objects: 100% (760/760), done.
Delta compression using up to 4 threads
Compressing objects: 100% (729/729), done.
Writing objects: 100% (760/760), 6.81 MiB | 45.00 KiB/s, done.
Total 760 (delta 71), reused 0 (delta 0)
remote: Resolving deltas: 100% (71/71), done.
To https://github.com/XuMouR/myblog_project.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

The One@The MINGW64 /d/myblog (master)
$ |
```

六、重启服务器后再启动 mysql 时，出现报错信息

```
Job for mariadb.service failed because the control process exited with error
code. See "systemctl status mariadb.service" and "journalctl -xe" for details
```



```
.
[root@VM-0-14-centos Myproject]# cd ~
[root@VM-0-14-centos ~]# systemctl start mariadb
Job for mariadb.service failed because the control process exited with error code. See "systemctl status mariadb.service" and "journalctl -xe" for details
.
[root@VM-0-14-centos ~]#
```

网上的参考：我的/etc/my/cnf 事先备份为 my_bak.cnf

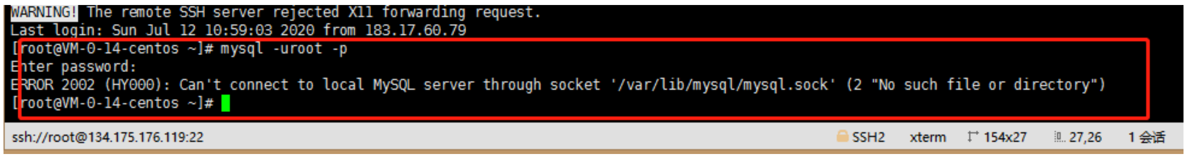
```
cp /usr/share/mysql/my-huge.cnf /etc/my.cnf

cp: overwrite ‘/etc/my.cnf’? y

之后执行

systemctl start mariadb.service

mariadb服务启动成功
```

无效，还是报一样的错

登陆mysql时：mysql -uroot -p

提示错误：

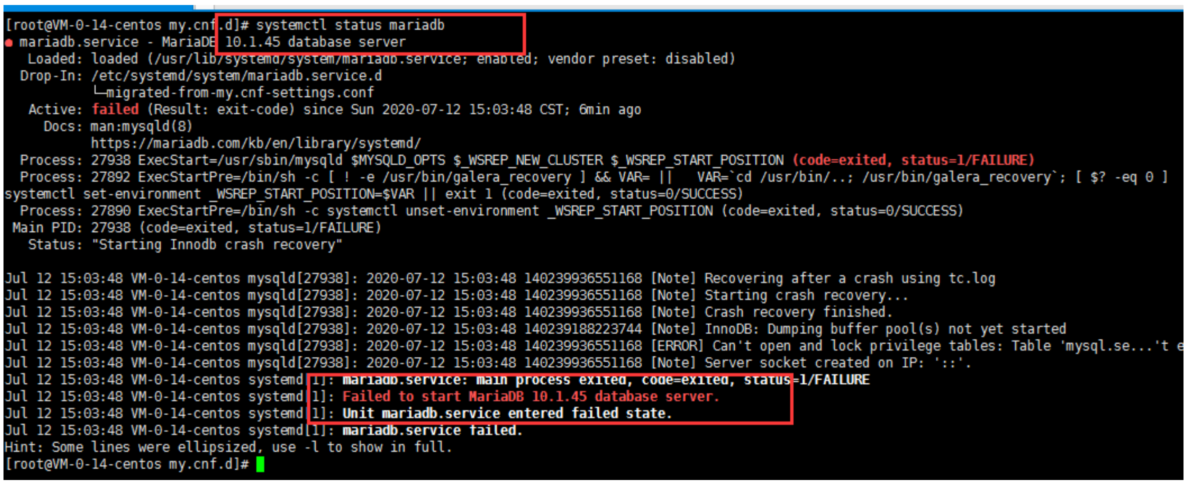
```
ERROR 2002 (HY000): Can't connect to local MySQL server through socket
'/var/lib/mysql/mysql.sock' (2 "No such file or directory")
```



```
WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Sun Jul 12 10:59:03 2020 from 183.17.60.79
[root@VM-0-14-centos ~]# mysql -uroot -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (2 "No such file or directory")
[root@VM-0-14-centos ~]#
```

查看mysql是否启动

```
service mysql status

systemctl status mariadb
服务端没启动top
```



```
[root@VM-0-14-centos my.cnf.d]# systemctl status mariadb
● mariadb.service - MariaDB 10.1.45 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/mariadb.service.d
           └─migrated-from-my.cnf-settings.conf
   Active: failed (Result: exit-code) since Sun 2020-07-12 15:03:48 CST; 6min ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 27938 ExecStart=/usr/sbin/mysqld $MYSQLD_OPTS $_WSREP_NEW_CLUSTER $_WSREP_START_POSITION (code=exited, status=1/FAILURE)
  Process: 27892 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`cd /usr/bin/..; /usr/bin/galera_recovery`; [ $? -eq 0 ]
systemctl set-environment _WSREP_START_POSITION=$VAR || exit 1 (code=exited, status=0/SUCCESS)
  Process: 27890 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
 Main PID: 27938 (code=exited, status=1/FAILURE)
   Status: "Starting Innodb crash recovery"

Jul 12 15:03:48 VM-0-14-centos mysqld[27938]: 2020-07-12 15:03:48 140239936551168 [Note] Recovering after a crash using tc.log
Jul 12 15:03:48 VM-0-14-centos mysqld[27938]: 2020-07-12 15:03:48 140239936551168 [Note] Starting crash recovery...
Jul 12 15:03:48 VM-0-14-centos mysqld[27938]: 2020-07-12 15:03:48 140239936551168 [Note] Crash recovery finished.
Jul 12 15:03:48 VM-0-14-centos mysqld[27938]: 2020-07-12 15:03:48 140239188223744 [Note] InnoDB: Dumping buffer pool(s) not yet started
Jul 12 15:03:48 VM-0-14-centos mysqld[27938]: 2020-07-12 15:03:48 140239936551168 [ERROR] Can't open and lock privilege tables: Table 'mysql.se...'t e
Jul 12 15:03:48 VM-0-14-centos mysqld[27938]: 2020-07-12 15:03:48 140239936551168 [Note] Server socket created on IP: '::'.
Jul 12 15:03:48 VM-0-14-centos systemd[1]: mariadb.service: main process exited, code=exited, status=1/FAILURE
Jul 12 15:03:48 VM-0-14-centos systemd[1]: Failed to start MariaDB 10.1.45 database server.
Jul 12 15:03:48 VM-0-14-centos systemd[1]: Unit mariadb.service entered failed state.
Jul 12 15:03:48 VM-0-14-centos systemd[1]: mariadb.service failed.
Hint: Some lines were ellipsized, use -l to show in full.
[root@VM-0-14-centos my.cnf.d]#
```