[一、辅助功能安装 3](#_Toc19566055)

[（一）安装网络工具net-tools和vim工具 3](#_Toc19566056)

[（二）设置root用户登录 3](#_Toc19566057)

[（三）更改Ubuntu的源为清华的（阿里云不需要） 3](#_Toc19566058)

[（四）系统更新和安装必要库 4](#_Toc19566059)

[（五）挂载新硬盘 4](#_Toc19566060)

[（六）创建soft、cloud文件夹 5](#_Toc19566061)

[（七）安装Conda环境 5](#_Toc19566062)

[（八）调整必要的环境变量 5](#_Toc19566063)

[（九）修改pip和conda默认的下载源为清华的（阿里云pip不需要） 5](#_Toc19566064)

[（十）更新pip和python的软链接 6](#_Toc19566065)

[（十一）安装virtualenv和virtualenvwrapper 6](#_Toc19566066)

[（十二）更新conda的版本 6](#_Toc19566067)

[二、设置Shadowsocks全局代理（有可能需要多试几次） 6](#_Toc19566068)

[三、安装DeepChem 8](#_Toc19566069)

[（一）源码方式安装 8](#_Toc19566070)

[（二）conda命令方式安装 9](#_Toc19566071)

[（三）直接方式安装 9](#_Toc19566072)

[（四）DeepChem测试 9](#_Toc19566073)

[四、安装Saleor 10](#_Toc19566074)

[（一）安装PostgreSQL 10](#_Toc19566075)

[（二）安装redis 11](#_Toc19566076)

[（三）、安装nodejs及更改其下载源 12](#_Toc19566077)

[（四）、创建虚拟环境并使用 12](#_Toc19566078)

[（五）、下载saleor源代码 12](#_Toc19566079)

[（六）、安装依赖 13](#_Toc19566080)

[（七）、同步数据库 13](#_Toc19566081)

[（八）、编译语言 13](#_Toc19566082)

[（九）、安装前台的依赖 13](#_Toc19566083)

[（十）、准备前台资源模块和E-Mail模块 14](#_Toc19566084)

[（十一）、运行服务器 14](#_Toc19566085)

[（十二）、教程 14](#_Toc19566086)

[（十三）、增加SSL支持 14](#_Toc19566087)

[五、安装KinomeRender 14](#_Toc19566088)

[（一）、安装 14](#_Toc19566089)

[（二）、增加可执行权限 14](#_Toc19566090)

[（三）、创建软链接 15](#_Toc19566091)

[六、安装依赖库 15](#_Toc19566092)

[七、安装主代码 15](#_Toc19566093)

[八、安装服务程序 15](#_Toc19566094)

[九、安装uWSGI 15](#_Toc19566095)

[（一）、安装 15](#_Toc19566096)

[（二）、配置 16](#_Toc19566097)

[十、安装Nginx 16](#_Toc19566098)

[（一）、安装 16](#_Toc19566099)

[（二）、配置 16](#_Toc19566100)

[十一、连接Saleor、uWGSI和Nginx 17](#_Toc19566101)

[（一）、部署静态文件 17](#_Toc19566102)

[（二）、连接三个软件 18](#_Toc19566103)

[十二、细节问题解决 18](#_Toc19566104)

[（一）、遇到import matplotlib.pyplot as plt问题（在deepchem安装部分解决） 18](#_Toc19566105)

[（二）、DEBUG=False时的说明 18](#_Toc19566106)

[（三）、遇到图片显示过大的问题 19](#_Toc19566107)

[（四）、修改settings.py使服务器自动创建对应分辨率的图片 19](#_Toc19566108)

[（五）、和图片相关的html文件修改后刷新没有更新的处理 19](#_Toc19566109)

[（六）、不退出后台运行程序 19](#_Toc19566110)

[（七）、Saleor和uWSGI结合后造成路径不对，找不到资源和模板 19](#_Toc19566111)

[（八）、遇到“uWSGI process got Segmentation Fault”类似问题 19](#_Toc19566112)

[（九）、遇到redis-server无法启动的问题 20](#_Toc19566113)

[（十）、遇到“/usr/bin/python: No module named virtualenvwrapper”问题 20](#_Toc19566114)

###### 一、辅助功能安装

（一）安装网络工具net-tools和vim工具

sudo apt-get remove -y vim-common

sudo apt-get install -y net-tools openssh-server vim

如果使用apt-get报错：

执行

sudo rm /var/lib/dpkg/lock

删除了之后，可以使用命令来更新一下，然后就可以安装程序了

sudo apt update

（二）设置root用户登录

sudo passwd root

sudo vi /etc/ssh/sshd\_config

将文件里的PermitRootLogin prohibit-password改成PermitRootLogin yes。之后重启服务：

sudo systemctl restart sshd

sudo reboot

后面所有操作均使用root用户登录操作，为了和阿里云一致。

（三）更改Ubuntu的源为清华的（阿里云不需要）

mv /etc/apt/sources.list /etc/apt/sources.list.bak

vi /etc/apt/sources.list

将如下内容拷贝到文件中：

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main restricted universe multiverse

（四）系统更新和安装必要库

apt-get -y update && apt-get -y upgrade

apt-get install -y git build-essential python3-dev python3-pip python3-cffi libcairo2 libpango-1.0-0 libpangocairo-1.0-0 libgdk-pixbuf2.0-0 libffi-dev shared-mime-info gettext ghostscript pdf2svg zip

**注意：** virtualenv后面使用pip安装。

（五）挂载新硬盘

说明：系统盘的容量太小，后面要计算任务的时候采用单独分区的形式，分区使用独立的硬盘挂载，便于管理。

fdisk -l

确定好需要挂载和分区的硬盘，然后创建分区、格式化、挂载等：

fdisk /dev/sdb

依次输入n、p、1、回车、y、w等（依次代表创建分区、创建主分区、分区号1、确定开始和结束扇区、移除已有的分区签名、写入修改等，如果事先有分区使用d命令删除），接下来格式化建立好的新分区：

mkfs -t ext4 /dev/sdb1

创建挂载点，并将新分区挂载到/cloud挂载点上：

mkdir /cloud

mount /dev/sdb1 /cloud

配置开机自动挂载：

blkid

上述命令会输出硬盘的uuid号，需要记下，下面用。之后编辑开机挂载文件：

vi /etc/fstab

添加类似下面的行：

UUID=413376c9-a9d0-4743-8496-de5599f79c13 /cloud ext4 defaults 0 1

使用命令测试挂载是否成功：

mount -a

（六）创建soft、cloud文件夹

说明：为了后续方便对系统进行管理，建立soft文件夹安装软件，建立cloud文件夹来放置一些资源文件。如果上面已经创建过就不用再次操作。

mkdir /soft

mkdir /cloud

（七）安装Conda环境 //Conda是管理多个python版本的工具

说明：这一步是为了安装虚拟环境conda。此时应该已经把Anaconda3-2019.07-Linux-x86\_64.sh文件传输到了/root目录下。

Cd //回到家目录

chmod +x Anaconda3-2019.07-Linux-x86\_64.sh //修改权限，这个文件需要下载

./Anaconda3-2019.07-Linux-x86\_64.sh //执行sh文件

安装的时候注意选择Anaconda的安装路径，这里假定安装在了/soft/conda目录下。

（八）调整必要的环境变量

直接将/soft/conda/bin下的conda命令建立/usr/bin下的软链接：

ln -s /soft/conda/bin/conda /usr/bin/conda

（九）修改pip和conda默认的下载源为清华的（阿里云pip不需要）

//用户操作

使用如下命令修改pip的源：

pip3 install pip -U

pip3 config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple

使用下述命令修改conda的源：

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/

conda config --set show\_channel\_urls yes

（十）更新pip和python的软链接

ln -s /usr/bin/pip3 /usr/bin/pip

ln -s /usr/bin/python3 /usr/bin/python

（十一）安装virtualenv和virtualenvwrapper

pip install virtualenv

pip install virtualenvwrapper

配置virtualenvwrapper：

vi .bashrc

添加下面的内容：

export WORKON\_HOME=$HOME/.virtualenvs

export VIRTUALENVWRAPPER\_PYTHON=/usr/bin/python

source /usr/local/bin/virtualenvwrapper.sh

或者是：

export WORKON\_HOME=$HOME/.virtualenvs  
export VIRTUALENVWRAPPER\_VIRTUALENV=/usr/local/bin/virtualenv  
export VIRTUALENVWRAPPER\_PYTHON=/usr/bin/python3  
source /usr/local/bin/virtualenvwrapper.sh

（十二）更新conda的版本

conda update -y -n base conda

###### 二、设置Shadowsocks全局代理（有可能需要多试几次）

pip3 install shadowsocks

遇到不能加载“main”的问题，解决方式在后面。新建一个配置文件shadowsocks.json，然后配置相应的参数：

{

"server": "149.28.214.22",

"server\_port": 1990,

"local\_port": 1080,

"password": "LIZHAOJUN19901112",

"timeout": 600,

"method": "aes-256-cfb"

}

直接启动会有问题，是由于在openssl 1.1.0中废弃了 EVP\_CIPHER\_CTX\_cleanup() 函数而引入了 EVE\_CIPHER\_CTX\_reset() 函数所导致的，所以首先修改相应的文件。

vi /usr/local/lib/python3.6/dist-packages/shadowsocks/crypto/openssl.py

搜索所有的cleanup （一共2处）并将其替换为 reset，之后启动服务。

sslocal -c shadowsocks.json -d start

安装polipo

apt-get install -y polipo

修改polipo的配置文件

vi /etc/polipo/config

添加下面的内容：

proxyAddress = "0.0.0.0"

socksParentProxy = "127.0.0.1:1080"

socksProxyType = socks5

chunkHighMark = 50331648

objectHighMark = 16384

serverMaxSlots = 64

serverSlots = 16

serverSlots1 = 32

重启polipo服务

service polipo restart

配置http和https代理

export http\_proxy="http://127.0.0.1:8123/"

export https\_proxy="http://127.0.0.1:8123/"

测试能否翻墙

curl www.google.com

ping www.baidu.com

如果都有响应即为成功。如果想使用全局代理，每次服务器重启后，都需要重新执行下面三个命令

sslocal -c shadowsocks.json -d start

export http\_proxy="http://127.0.0.1:8123/"

export https\_proxy="http://127.0.0.1:8123/"

###### 三、安装DeepChem

（一）源码方式安装

DeepChem的安装以源码方式为宜，需要注意的是，因为有些依赖包需要使用外网的conda channel源去下载，所以速度很慢（基于https方式），所以，采取的策略为先找到这些包，手动下载然后拷贝到/root/deepchem\_srcs目录下，使用手动安装的方式。

cd

conda create -y -n deepchem python=3.6

直接使用source或者conda命令激活虚拟环境会有问题，因为没有把安装路径放置到PATH中，所以需要额外使用一个命令把相应的路径添加到.bashrc中。

echo ". /soft/conda/etc/profile.d/conda.sh" >> ~/.bashrc

exit

conda activate deepchem

conda install ~/fftw3f-3.3.4-2.tar.bz2

conda install ~/openmm-7.1.1-py36\_0.tar.bz2

conda install ~/pdbfixer-1.4-py36\_0.tar.bz2

(这三个软件是外网channel里才有的，其他的依赖软件从清华的一般都可以下载)

将DeepChem的源码压缩包拷贝到/root目录下，并安装：

tar -zxvf deepchem-2.0.0.tar.gz

cd deepchem

将修改过的安装脚本install\_deepchem\_conda.2.0.0.sh替换成install\_deepchem\_conda.sh，然后安装（注意修改tensorflow的版本为cpu或者gpu，还有deepchem文件夹下的setup.cfg文件中的版本号）：

mv ~/install\_deepchem\_conda.2.0.0.sh scripts/install\_deepchem\_conda.sh

bash scripts/install\_deepchem\_conda.sh

python setup.py install

（二）conda命令方式安装

Conda命令方式使用比较简单，但是在没有好的网络情况下速度奇慢无比，可以在配置好全局代理，网络较好的时候使用。

conda install -c deepchem -c omnia deepchem=2.0.0

（三）直接方式安装

在修改过的安装脚本install\_kinome\_conda.sh路径下安装，不需要安装openmm等软件，注意修改tensorflow的版本为cpu或者gpu：

conda create -y -n kinome python=3.6

echo ". /soft/conda/etc/profile.d/conda.sh" >> ~/.bashrc

exit

conda activate kinome

/bin/bash install\_kinome\_conda.sh

然后将deepchem的安装文件夹直接放置在conda对应环境的路径下即可。

（四）DeepChem测试

测试DC，测试的时候需要先把deepchem安装包里的datasets文件夹放到安装conda环境的site-packages下（如conda环境名字为deepchem，则放在/soft/conda/envs/deepchem/lib/python3.6/site-packages下）一般情况下可以测试成功，不过也有特殊情况，只要能正常使用即可。

nosetests -a '!slow' -v deepchem –nologcapture

**注意：**

1、 需要手动把deepchem的datasets文件夹，放到/soft/conda/envs/deepchem/lib/python3.6/site-packages路径下；

2、 Tensorflow的版本目前只能选1.6的，因为新版本里去掉了一个函数；

3、测试的时候，不要在路径下有“deepchem”的文件夹位置处运行测试，否则会默认使用这个模块；

4、如果出现“AssertError”这种测试错误，多试几次就可以成功；

5、安装完deepchem后，需要手动更新numpy的版本（至少是1.14.3以上）: conda upgrade numpy，否则会出现“Intel MKL FATAL ERROR: Cannot load libmkl\_avx2.so or libmkl\_def.so.”错误；

6、如果出现KERAS\_BACKEND无法切换的情况，需要执行“rm /soft/conda/envs/deepchem/etc/conda/activate.d/ -rf”和“rm /soft/conda/envs/ deepchem /etc/conda/deactivate.d/ -rf”操作，然后重新激活虚拟环境即可；

7、有一些测试是“flaky”的，即有时候成功，有时候不成功，只要能成功一次即可。更完整的测试命令可参考：nosetests --with-flaky -a '!slow' --with-timer --with-coverage --cover-package=deepchem -v deepchem --nologcapture。Flaky is a plugin for nose or py.test that automatically reruns flaky tests。它是一个nose和py.test的插件，能够在那些不稳定的case失败之后，重复执行这些case，或者带参数的nosetests --with-flaky --max-runs=5 --min-passes=1 -a '!slow' --with-timer --with-coverage --cover-package=deepchem -v deepchem –nologcapture；

8、如果出现找不到deepchem包的情况，有可能是python的路径错误导致的，必须要修改成链接到python 3环境下。

###### 四、安装Saleor

（一）安装PostgreSQL

**注意：** 首先要退出conda环境。

基于Django的Web一般都使用PostgreSQL作为数据库引擎，和MySQL的使用方式大同小异，有一些语法上的区别。安装过程为：

apt-get install -y postgresql postgresql-client

创建数据库用户并配置为超级用户，类似于在MySQL里新建用户，并grant all privileges给他所有权限。比如这里配置数据库用户simm，并指定其为超级用户，使用下面的命令：

sudo -u postgres createuser --superuser simm

（出现“could not change directory to "/root": Permission denied”错误可以忽略。）更改新创建用户的密码。登录数据库控制台，为刚刚创建的用户设置密码，命令如下：

//系统用户（root或者加sudo可以的）

sudo -u postgres psql

\password simm

（这时候会提示你输入新密码，输入两遍即可，界面上是不明文显示的，这里假定输入的是simm，这些参数后面用）

\q（这是退出PostgreSQL数据库的命令）

创建新的数据库，并指定owner。例如创建一个叫web的数据库，并指定刚创建的simm用户为其所有人，命令为：

sudo -u postgres createdb --owner=simm web

（出现上面同样的错误忽略。）下面是一些PostgreSQL数据库的常用命令，例如登录数据库为：

psql -U simm -d web -h 127.0.0.1 -p 5432， -U 后是用户名，-d后是数据库的名字，-h是主机地址，-p是端口号，一般为5432（类似MySQL为3306的意思）。

PostgreSQL查看数据库命令为（必须要登录才能用）：

\l

PostgreSQL使用某个数据库命令为（databaseName为数据库名）：

\c databaseName

删除数据库web的命令为：

sudo -u postgres dropdb web

（二）安装redis

redis是一个基于内存的数据库，安装命令为：

apt-get -y install redis-server

一般使用上述命令安装完后会自动启动redis，如果要手动启动redis，可使用下面的命令（阿里云上自动启动失败，需要手动启动，另外redis-server如果安装不成功，需要先删除redis-tools软件）（运行时不启动也可以）：

redis-server

想要查看redis是否启动，可使用命令：

redis-cli

在提示符后输入ping，输出PONG说明成功安装，再输入exit可退出环境。

（三）、安装nodejs及更改其下载源

curl -sL https://deb.nodesource.com/setup\_12.x | sudo -E bash -

apt-get install -y nodejs

npm config set registry https://registry.npm.taobao.org

（四）、创建虚拟环境并使用

创建一个名为saleor的虚拟环境并使用：

mkvirtualenv saleor

此时shell界面将显示类似(saleor)root@ia:~#，即已经在虚拟环境中。上述命令如无法执行则需先退出当前shell，再重新进入。

**注：**要激活saleor虚拟环境，使用命令为：

workon saleor

**注：**要取消激活虚拟环境，使用命令为：

deactivate

（五）、下载saleor源代码

这一步可以在配置完全局代理之后，使用git clone直接下载下来，使用的命令为：

git clone -b 2.8.0 https://github.com/mirumee/saleor.git

而如果没有配置全局代理，就用打包好的gz包安装。

-------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

tar -zxvf saleor\_20190916\_2.8.0.tar.gz

（六）、安装依赖

**注意：**安装前首先要删除requirements.txt和requirements\_dev.txt中对源的指定，即-i选项，对其第一行使用“#”注释掉即可。

pip install -r requirements.txt -i <https://pypi.tuna.tsinghua.edu.cn/simple> //注释-i以后这样默认使用的系统的源

pip install -r requirements\_dev.txt // wrapt==1.11.2 这里可能会出现报错

**注意：**使用代理时，上述命令有可能需要重复执行几次才可以完成。

使用命令安装ssl依赖（django-sslserver是为了后面的ssl访问）：

pip install django-sslserver

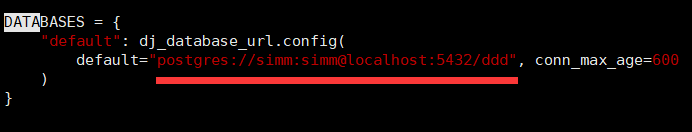
（七）、同步数据库

在文件/saleor/saleor/settings.py中修改：



好像时不是空字符串，运行manage.py会报错的，值写成:

”abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&\*(-\_=+)”





同步数据库是Django软件自带的功能，直接会根据py代码创建好对应的数据表，将修改好的saleor文件夹覆盖掉对应的原始2.8.0版本文件夹（主要是locale、saleor和templates文件夹，删除不必要的locale文件夹），使用如下命令完成数据库的同步（**注意要先确定saleor的settings.py没问题**）：

python manage.py makemigrations

（这一步是自动检查数据表的修改情况）

python manage.py migrate

（这一步会输出很多Applying，即是对数据库的一些操作信息）

（八）、编译语言

Saleor安装路径下的locale语言文件夹中的文件需要后必须要编译才能正常使用，因此必须使用下面的命令操作一下：

python manage.py compilemessages

（九）、安装前台的依赖

这个是利用NodeJS的功能，安装一些依赖包，如下：

npm install

**注意：**这个命令如果遇到问题，可以重复执行几次。“PUPPETEER\_SKIP\_CHROMIUM\_DOWNLOAD”命令可以取消下载Chromium软件，使用的命令为env PUPPETEER\_SKIP\_CHROMIUM\_DOWNLOAD=true npm install。

（十）、准备前台资源模块和E-Mail模块

使用的命令为：

npm run build-assets && npm run build-emails

（十一）、运行服务器

这一步将最终运行saleor软件，可以加一些参数，简单的方式为：

python manage.py runserver 0.0.0.0:80

（十二）、教程

具体使用方法参照教程https://docs.getsaleor.com/docs/，也可以使用中文教程https://jeremyyin.com/saleor/，最好使用英文的，更符合原意。

（十三）、增加SSL支持

先把证书文件放到特定路径下，启动服务器需要使用命令：

python manage.py runsslserver --certificate /path/to/certificate --key /path/to/key 0.0.0.0:443

###### 五、安装KinomeRender

（一）、安装

将KinomeRender的所有文件（modules文件夹、postscript文件夹和kinome-render文件）放置在/soft/kinome\_render/bin路径下。

（二）、增加可执行权限

切换到/soft/kinome\_render/bin下，给kinome-render文件增加可执行权限：

chmod +x kinome-render

（三）、创建软链接

ln -s /soft/kinome\_render/bin/kinome-render /usr/bin/kinome-render

###### 六、安装依赖库

将Lib的所有文件（simm文件夹）放在/conda/envs/kinome环境对应的位置上。

###### 七、安装主代码

主代码是服务运行的启动代码，对于Kinome的例子来说，主代码是exe\_start.py文件，可执行脚本是exe.sh文件。exe.sh文件中激活conda环境的方法是“.”，exe\_start.py使用的参数使用“$1”、“$2”等形式传递过去。exe\_start.py和exe.sh打包为sources.zip，用于在服务运行时作为启动主程序，放置在/cloud/sources下对应的位置上。

###### 八、安装服务程序

服务程序是Web后台的监控程序，负责作业的启停，以及相关文件的复制、处理等等。将所有的服务代码放置在/soft/backend路径下。

###### 九、安装uWSGI

（一）、安装

saleor安装的时候已经有了uwsgi，但是这个软件没有ssl的支持，因此必须先重新安装才可以。

pip uninstall uwsgi

apt-get install libssl-dev

git clone https://github.com/unbit/uwsgi.git

cd uwsgi

python uwsgiconfig.py --build

为了和之前uwsgi安装的时候路径一致，建立和之前一样的软链接：

ln -s /soft/uwsgi/uwsgi /usr/local/bin/uwsgi

（二）、配置

在saleor目录下saleor/wsgi下有样例配置文件uwsgi.ini，需要修改一些值，重要的值为：chdir必须切换到Saleor的目录，不然找不到启动模块。processes和threads分别为进程和线程数量。socket为与后面nginx通讯的socket接口，必须要和后面一样。**这里不指定http或https的值，因为后面使用nginx作为服务器。**

###### 十、安装Nginx

（一）、安装

apt-get install -y nginx

（二）、配置

首先需要修改nginx默认的配置文件，默认监听的80端口会对后面造成影响。

vi /etc/nginx/sites-enabled/default

修改成如下的值：

listen 4321 default\_server;

listen [::]:4321 default\_server;

在manage.py同路径下创建两个文件，一个uwsgi\_params，一个saleor\_nginx.conf文件，uwsgi\_params文件的内容为官方给的典型值，saleor\_nginx.conf的重要的值为：

upstream django {

server 127.0.0.1:1234; #必须要和uwsgi里设置的一致

}

server {

listen 0.0.0.0:443;#ssl时必须这样指定

ssl on;#必须要加如下选项

ssl\_certificate /soft/keys/fullchain.pem;

ssl\_certificate\_key /soft/keys/privkey.pem;

location /media {

alias /cloud/server/media;

}

location /static {

alias /cloud/server/static;

}

location / {

uwsgi\_pass django;

include /soft/saleor/uwsgi\_params;#上述值必须要正确

}

}

创建软链接，链接到nginx的配置文件目录下：

ln -s /soft/saleor/saleor\_nginx.conf /etc/nginx/sites-enabled/

上述命令必须使用绝对路径才可以。

###### 十一、连接Saleor、uWGSI和Nginx

（一）、部署静态文件

在saleor路径下，执行下面的命令：

python manage.py collectstatic

会生成static文件夹，将其复制到/cloud/server下：

cp static /cloud/server/ -R

将media文件夹也复制到/cloud/server下，saleor本身的media文件夹存放上传的文件，不再做结果展示的存储目录。

cp media /cloud/server -R

（二）、连接三个软件

启动nginx：

service nginx restart

启动uwsgi：

nohup uwsgi --ini /soft/saleor/saleor/wsgi/uwsgi.ini > uwsgi.log 2>&1 &

uwsgi的停止命令为：

killall -9 uwsgi

到此即实现了并发访问设置。

###### 十二、细节问题解决

（一）、遇到import matplotlib.pyplot as plt问题（在deepchem安装部分解决）

原因：安装的matplotlib版本过高，解决方式：

pip uninstall matplotlib

pip install matplotlib

（二）、DEBUG=False时的说明

1、settings.py里DEBUG=False；

2、urls.py里添加如下字段：

import django.views.static as s\_static

#在if settings.DEBUG后的else中

urlpatterns += [

#url(r'^static/(?P<path>.\*)$', s\_static.serve, {'document\_root': settings.STATIC\_ROOT}),

url(r'^media/(?P<path>.\*)$', s\_static.serve, {'document\_root': settings.MEDIA\_ROOT})

]

（三）、遇到图片显示过大的问题

图片显示大的时候，需要使用小的图片，做法是修改settings.py中的参数，使服务器自动创建对应分辨率的图片，具体方式见四。

（四）、修改settings.py使服务器自动创建对应分辨率的图片

settings.py中的VERSATILEIMAGEFIELD\_SETTINGS参数，设置为True即可：

VERSATILEIMAGEFIELD\_SETTINGS = {

# Images should be pre-generated on Production environment

'create\_images\_on\_demand': get\_bool\_from\_env(

'CREATE\_IMAGES\_ON\_DEMAND', True), }

（五）、和图片相关的html文件修改后刷新没有更新的处理

1、试着清空cache；

2、重启服务器。

（六）、不退出后台运行程序

仿照如下命令

nohup python manage.py runsslserver --certificate /encrypt\_kinome/fullchain.pem --key /encrypt\_kinome/privkey.pem 0.0.0.0:443 > web.log 2>&1 &

（七）、Saleor和uWSGI结合后造成路径不对，找不到资源和模板

需要修改Saleor的setttings.py文件，其中的PROJECT\_ROOT值修改为：

PROJECT\_ROOT = os.path.normpath(os.path.join(os.path.dirname(os.path.abspath(\_\_file\_\_)), '..'))

（八）、遇到“uWSGI process got Segmentation Fault”类似问题

psycopg2需要从源码来重装，链接为：https://pypi.org/project/psycopg2/

（九）、遇到redis-server无法启动的问题

按以下方式处理：

sudo apt-get purge redis-server

sudo add-apt-repository ppa:chris-lea/redis-server

sudo apt-get update

sudo apt -y install redis-server

（十）、遇到“/usr/bin/python: No module named virtualenvwrapper”问题

原因在于之前安装软件时破坏了python的软链接，修正：

rm /usr/bin/python

ln -s /usr/bin/python3 /usr/bin/python