

情况说明：

现在工作较忙，所以暂时先写了一个手动安装itelftool的文档，脚本自动安装后续有时间会逐渐完善。

1、操作系统安装

系统：centos 7.2(1511) 64位

操作界面：字符界面即可，即最小化安装

配置：4核4G内存100G磁盘（磁盘可以根据自身使用环境进行增加或减小）

注1：安装系统时，语言选择中文（简体中文）

注2：注意分区，尽量手动分区

注3：安装好系统后，注意配置固定ip地址，DNS，SSH，以及禁用防火墙（如果是内网则禁用防火墙，如果是外网则需要开放后续指定的程序端口）

注4：如果是刚装的系统，可以使用yum安装net-tools,telnet,vim等工具来沿袭使用centos7以前系统的部分常用命令

2、安装基本软件

```
# yum -y groupinstall "Development tools"
```

3、安装python

python版本我们选择的是3.5.6

首先安装python依赖包：

```
# yum -y install openssl-devel sqlite-devel bzip2-devel ncurses-devel gdbm-devel
 readline-devel tcl-devel tk-devel xz-devel zlib-devel db4-devel libpcap-devel
```

接着下载Python3.5的源码包并编译：

解压至/home/soft/下：

```
# pwd
/home/soft/Python-3.5.6
# ./configure --prefix=/home/software/python3.5 --enable-shared
# make -j 4
# make install
```

make install 完成之后，会在最后输出：

```
Installing collected packages: setuptools, pip
```

```
Successfully installed pip-9.0.1 setuptools-28.8.0
```

意思是说安装python自带pip以及setuptools

后续步骤：

```
# which python  
/usr/bin/python  
# cd /usr/bin/  
# mv python python2.7.5  
# ln -s /home/software/python3.5/bin/python3 /usr/bin/python  
# echo "/home/software/python3.5/lib" > /etc/ld.so.conf.d/python3.5.conf  
# ldconfig
```

测试python是否能用：

```
# python --version  
Python 3.5.6
```

修改yum脚本，否则无法使用yum：

将/usr/bin/yum以及/usr/libexec/urlgrabber-ext-down的首行修改成：

```
#!/usr/bin/python2.7.5
```

执行

```
# yum clean all  
# yum list  
yum恢复正常使用
```

升级pip、setuptools：

因为python3.5自带pip，setuptools

```
# /home/software/python3.5/bin/pip3 install --upgrade pip
```

```
# ln -s /home/software/python3.5/bin/pip /usr/bin/pip3
```

```
# pip3 -V
```

```
pip 18.0 from /home/software/python3.5/lib/python3.5/site-packages/pip (python  
3.5)
```

```
# wget https://bootstrap.pypa.io/ez_setup.py -O - | python
```

```
# ln -s /home/software/python3.5/bin/easy_install /usr/bin/easy_install
# easy_install --version
setuptools 33.1.1 from /home/software/python3.5/lib/python3.5/site-
packages/setuptools-33.1.1-py3.5.egg (Python 3.5)
```

4、安装mysql

安装版本：5.6.40 社区版

安装依赖关系包：

```
# yum -y install make gcc-c++ cmake bison-devel ncurses-devel libaio libaio-devel
perl-Data-Dumper
```

创建mysql组和用户：

```
# groupadd mysql
# useradd -g mysql -s /sbin/nologin mysql
```

创建相关目录：

```
# mkdir /home/software/mysql/sock
# mkdir /home/software/mysql
# mkdir /home/software/mysql/data
# mkdir /home/software/mysql/pid
# mkdir /home/software/mysql/log
# echo > /home/software/mysql/log/mysql.error.log
# echo > /home/software/mysql/pid/localhost.localdomain.pid
# chown mysql.mysql mysql/ -R
```

将源码包解压，并进入解压包，执行cmake操作：

```
# cmake -DCMAKE_INSTALL_PREFIX=/home/software/mysql -
DMYSQL_DATADIR=/home/software/mysql/data -DSYSCONFDIR=/etc -
DWITH_MYISAM_STORAGE_ENGINE=1 -DWITH_INNOBASE_STORAGE_ENGINE=1 -
DWITH_MEMORY_STORAGE_ENGINE=1 -DWITH_READLINE=1 -
DMYSQL_UNIX_ADDR=/home/software/mysql/sock/mysql.sock -
DMYSQL_TCP_PORT=3306 -DENABLED_LOCAL_INFILE=1 -
DWITH_PARTITION_STORAGE_ENGINE=1 -DEXTRA_CHARSETS=all -
DDEFAULT_CHARSET=utf8 -DDEFAULT_COLLATION=utf8_general_ci
```

```
# make -j 8  
# make install
```

赋权给mysql用户：

```
# cd /home/software/  
# chown mysql.mysql mysql/ -R  
# ll -d mysql/  
drwxr-xr-x. 14 mysql mysql 4096 8月 22 16:57 mysql/
```

初始化数据库：

```
# cd /home/software/mysql/scripts  
# ./mysql_install_db --basedir=/home/software/mysql --  
datadir=/home/software/mysql/data --user=mysql  
# ls /home/software/mysql/data  
ibdata1 ib_logfile0 ib_logfile1 mysql performance_schema test  
有数据即可
```

拷贝服务脚本：

```
# cd /home/software/mysql  
# cp -ar support-files/mysql.server /etc/init.d/mysqld  
# chkconfig mysqld on
```

重新编写配置文件：

```
# cat /etc/my.cnf  
[client]  
port      = 3306  
socket     = /home/software/mysql/sock/mysqld.sock  
default_character_set=utf8  
[mysqld]  
port      = 3306  
socket     = /home/software/mysql/sock/mysqld.sock  
pid-file  = /home/software/mysql/pid/localhost.localdomain.pid  
log-error = /home/software/mysql/log/mysql.error.log  
general-log-file = /home/software/mysql/log/mysql.log
```

```
log_bin_trust_function_creators=1
long_query_time=15
slow_query_log=1
slow-query-log_file=/home/software/mysql/log/mysql-slow-query.log
event_scheduler=ON
symbolic-links=0
skip-name-resolve
default_storage_engine=INNODB
character_set_server=utf8
collation_server=utf8_general_ci
log-bin=mysql-bin
log-slave-updates=1
server-id = 1
[mysqldump]
quick
max_allowed_packet = 16M
[mysql]
no-auto-rehash
[myisamchk]
key_buffer_size = 512M
sort_buffer_size = 256M
read_buffer = 32M
write_buffer = 32M
[mysqlhotcopy]
interactive-timeout
```

数据库删除空密码用户，以及创建密码：

先安装客户端：

```
# yum -y install mysql
```

空密码登录：

```
# mysql -uroot -p
```

```
mysql> delete from mysql.user where user='';
```

```
mysql> delete from mysql.db where user='';
```

```
mysql> update mysql.user set password=password('asdf1234');
```

```
mysql> grant all on *.* to 'root'@'%' identified by 'asdf1234' with grant option;
```

```
mysql> flush privileges;
```

5、安装redis

redis 版本，选择的是4.0.11

```
# tar xzf redis-4.0.11.tar.gz  
# cd redis-4.0.11  
# make -j 4
```

将redis.conf 拷贝到etc目录下：

```
# cp -ar redis.conf /etc/
```

将/etc/redis.conf修改 daemonize no 为 daemonize yes，这样就可以默认启动就后台运行

```
vim /etc/redis.conf  
daemonize yes
```

将redis移动到统一软件路径下：

```
# mv redis-4.0.11 /home/software/redis
```

启动：

```
# /home/software/redis/src/redis-server /etc/redis.conf
```

将上述命令添加到/etc/rc.local 设置为开机自启动

6、安装django

```
# pip3 install Django==1.11.14
```

Installing collected packages: pytz, Django

Successfully installed Django-1.11.14 pytz-2018.5

检查/home/software/python3.5/lib/python3.5/site-packages/下是否有Django模块

python测试是否能导入django模块：

```
# python3  
Python 3.5.6 (default, Aug 22 2018, 14:18:58)  
[GCC 4.8.5 20150623 (Red Hat 4.8.5-4)] on linux  
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import django  
>>> exit()
```

7、创建一个django测试工程

```
# cd /home/app/  
# /home/software/python3.5/lib/python3.5/site-packages/django/bin/django-admin.py startproject testproject
```

```
# cd /home/app/testproject/testproject
```

下面黑体为新增测试内容：

```
# cat views.py  
from django.shortcuts import render,HttpResponse  
def helloworld(request):  
    return HttpResponse('hello world')
```

```
# cat urls.py  
from django.conf.urls import url  
from django.contrib import admin  
from . import views  
urlpatterns = [  
    url(r'^admin/', admin.site.urls),  
    url(r'^hello/', views.helloworld),  
]
```

内容更改：

settings.py文件中：

将

ALLOWED_HOSTS = []

更改为：

ALLOWED_HOSTS = ['192.168.1.241']

启动测试项目：

```
# cd /home/app/testproject  
# python manage.py runserver 0.0.0.0:8008
```

浏览器访问：

<http://192.168.1.241:8008/hello/>

页面返回hello world即为测试成功。

8、安装python插件

注：如果在pip安装过程中，非常慢或者失败，可以尝试使用阿里云安装源，也就是在安装命令后加上（--trusted-host mirrors.aliyun.com）或者使用清华大学的pip安装源，在命令后加上（-i <https://pypi.tuna.tsinghua.edu.cn/simple>）

```
# pip3 install djangorestframework==3.8.2 -i  
https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install django-session-security==2.6.0 -i  
https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install cryptography==2.3 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install paramiko==2.0.2 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install pycparser==2.18 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install PyMySQL==0.9.2 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install sh==1.12.9 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# yum -y install python-devel mysql-devel  
# pip3 install mysqlclient==1.3.12 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install pytest==3.0.7 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install celery==4.2.1 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install django-celery-beat==1.1.1 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install django_celery_results==1.0.1 -i  
https://pypi.tuna.tsinghua.edu.cn/simple (有缺陷)  
# pip3 install gunicorn==19.7.1 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install redis==2.10.6 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install psutil==5.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install schedule==0.4.3 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install pymongo==3.3.0 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install requests==2.11.1 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install django-db==0.0.7 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install Pillow==5.2.0 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

ansible模块相关：

```
# yum install -y epel-release gcc expect python-pip python-devel smartmontools  
dmidecode libselinux-python git rsync dos2unix openssl openssl-devel  
# pip3 install ansible==2.6.3 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

webshell模块相关：

```
# pip3 install channels==2.0.2 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install channels-redis==2.1.0 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install amqp==1.4.9 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install anyjson==0.3.3 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install asgi-redis==1.4.3 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install asgiref==2.3.0 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install async-timeout==3.0.0 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install attrs==17.4.0 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

域名模块相关：

```
# pip3 install aiohttp==3.4.4 -i https://pypi.tuna.tsinghua.edu.cn/simple  
# pip3 install bs4==0.0.1 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

9、安装相关模块需要的条件

```
# cd /home/app/
```

下载源码：

```
# git clone https://github.com/420521738/itelftool.git
```

或者在github上下载zip包，浏览器下载zip包更快，使用git clone比较慢，我这里是先下载好zip包

```
# cd /home/soft  
# unzip itelftool-master.zip  
# mv itelftool-master /home/app/itelftool
```

ansible模块相关：

创建目录：

```
main_dir="/var/opt/itelftool"  
data_dir="$main_dir/data"  
mkdir -p $data_dir/ansible/playbook  
mkdir -p $data_dir/ansible/roles
```

```
mkdir -p /etc/ansible/
# cp -ar /home/app/itelftool/installfile/server/ansible/ansible.cfg /etc/ansible/
```

做软链接：

```
# ln -s /home/software/python3.5/bin/ansible /usr/bin/ansible
# ln -s /home/software/python3.5/bin/ansible-playbook /usr/bin/ansible-playbook
```

创建hosts文件（下面是举例）：

```
/etc/ansible/hosts
# cat /etc/ansible/hosts
LinuxServer1 ansible_host=192.168.1.234 host_name=LinuxServer1
LinuxServer2 ansible_host=192.168.1.235 host_name=LinuxServer2
CentosServer1 ansible_host=192.168.1.240 host_name=CentosServer1
WindowsServer1 ansible_host=192.168.7.199 host_name=WindowsServer1
[UbuntuGroup]
LinuxServer1
LinuxServer2
[CentosGroup]
CentosServer1
[WindowsGroup]
WindowsServer1
[LinuxGroup]
LinuxServer1
LinuxServer2
CentosServer1
```

任务编排相关：

```
# pip3 install django_celery_results==1.0.1 注意：这样执行是不行的，1.0.1包有缺陷，无法获取结果对应的任务名和任务类型
```

需要通过下面的方法来安装这个结果插件：

<https://github.com/celery/django-celery-results>

下载该插件的源码包，该源码包已经对结果的表结构进行了优化。

执行：

```
# unzip django-celery-results-master.zip
# cd django-celery-results-master
```

```
# python setup.py build  
# python setup.py install
```

上述的安装包在准备环节都已经进行安装了，可以再检查一遍是否已经安装完成。

注意一点：

需要做软链接，否则找不到celery的二进制执行命令

```
# ln -s /home/software/python3.5/bin/celery /usr/bin/celery
```

celery以及celery beat在服务器端的处理：

```
# mkdir /var/opt/itelftool/pid  
# main_dir="/var/opt/itelftool"  
# config_dir="$main_dir/config"  
# mkdir -p $config_dir/celery  
# cd /home/app/itelftool/installfile/server/celery/  
# cp -ar beat.conf $config_dir/celery/beat.conf  
# cp -ar celery.service /usr/lib/systemd/system/  
# cp -ar start_celery.sh $config_dir/celery/start_celery.sh  
# cp -ar beat.service /usr/lib/systemd/system/  
# chmod +x $config_dir/celery/start_celery.sh  
# systemctl daemon-reload  
# chkconfig celery on  
# chkconfig beat on
```

启动celery：

```
# service celery start  
# service beat start
```

10、导入数据库结构

先手动创建itelftool数据库：

```
MySQL [(none)]> create database itelftool default charset utf8;
```

11、创建管理员以及常规操作

```
# cd /home/app/itelftool
```

生成数据库表结构：

```
# python manage.py makemigrations  
# python manage.py migrate
```

创建管理员：

```
# cd /home/app/itelftool  
# python manage.py createsuperuser  
Email address: 420521738@qq.com  
Name: 陈秋飞  
Password:  
Password (again):  
Superuser created successfully.
```

12、字体拷贝

由于验证码需要用到STZHONGS.TTF这个字体，也可以使用其他的字体，看个人喜欢，将字体上传到系统的/usr/share/fonts/下即可

```
# ll /usr/share/fonts/STZHONGS.TTF  
-rw-r--r--. 1 root root 12135284 10月 29 2002 /usr/share/fonts/STZHONGS.TTF
```

13、启动itelftool

```
# cd /home/app/itelftool  
# python manage.py runserver 0.0.0.0:9009
```

启动之后，浏览器访问

<http://192.168.1.241:9009>

即可。