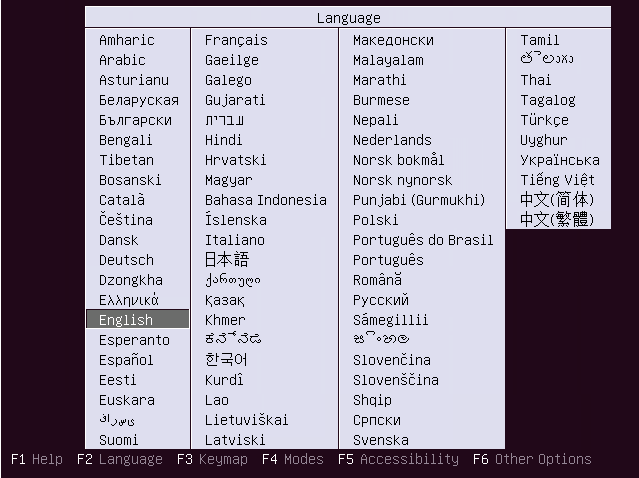
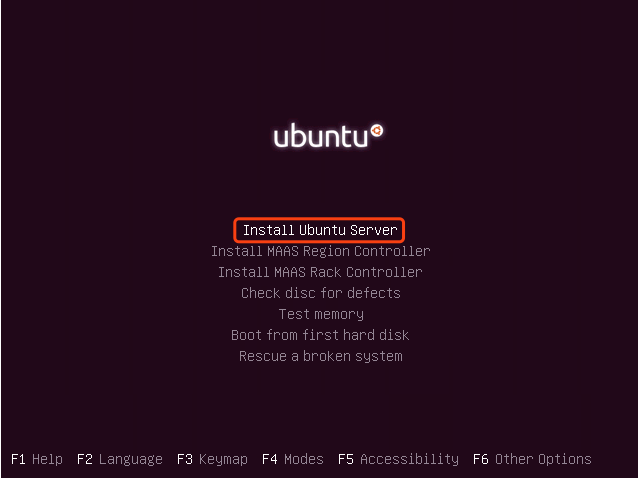
f环境安装

Ubuntu16.04 server 安装

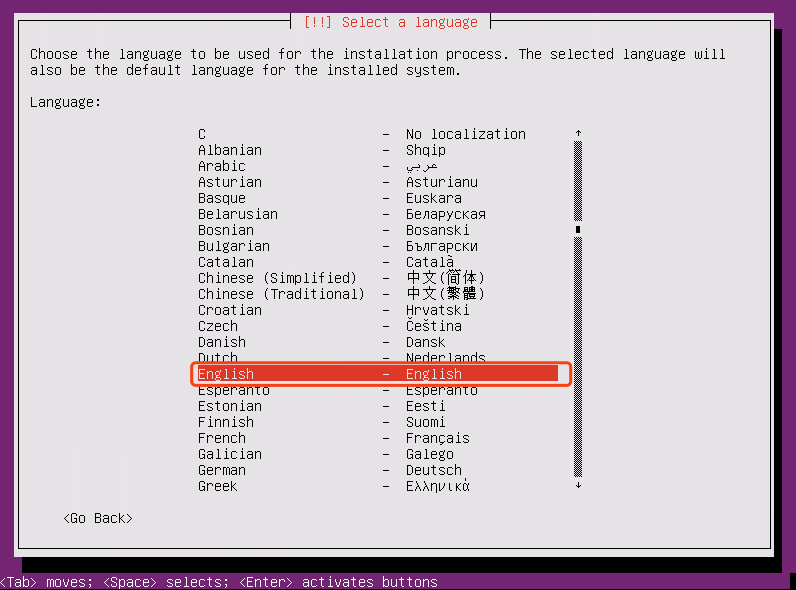
1.选择系统语言-English



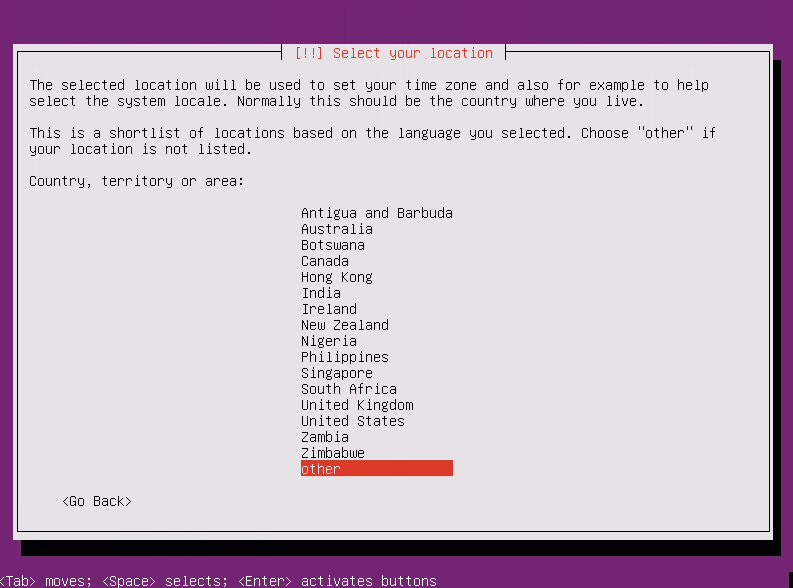
2.选择操作-Install Ubuntu Server



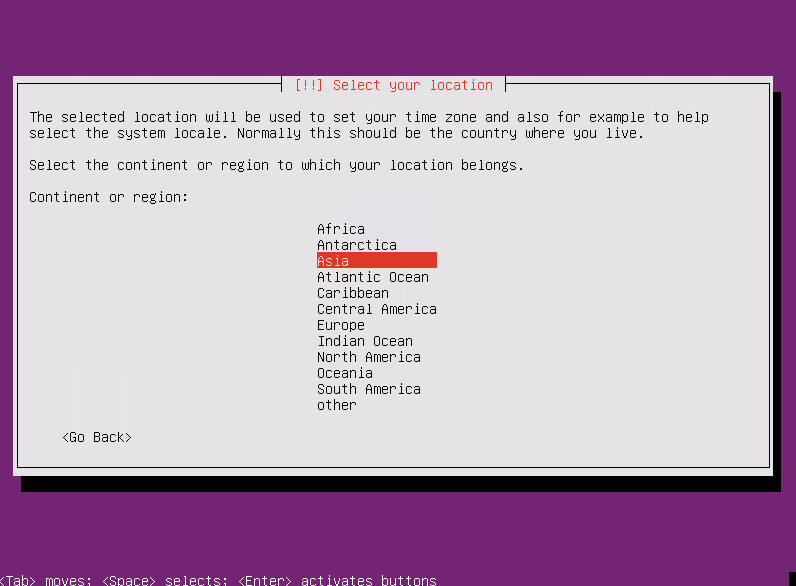
3.选择安装过程和系统的默认语言-English



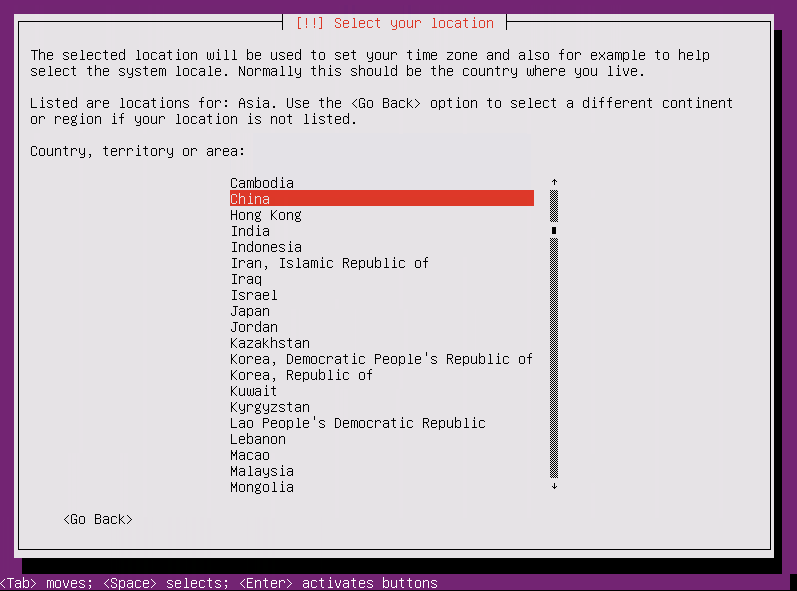
4.选择区域-other



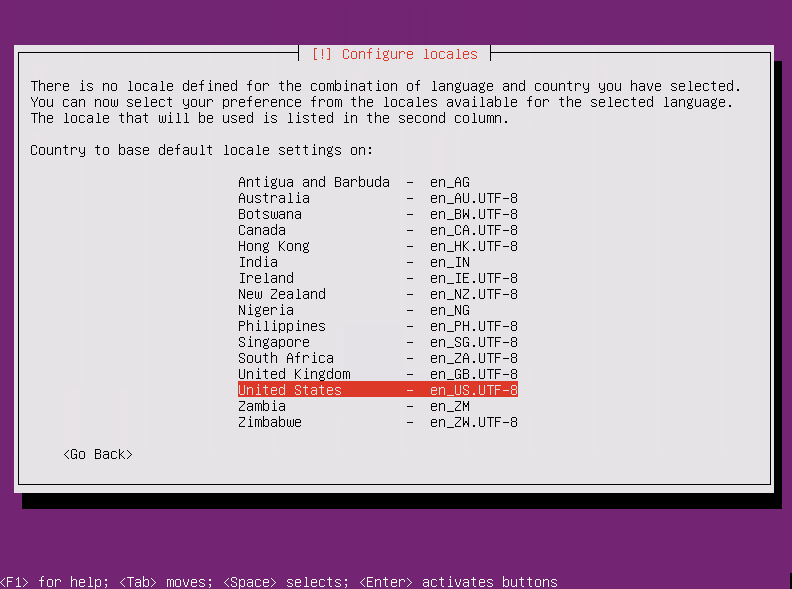
5.选择亚洲-Asia



6.选择国家-China



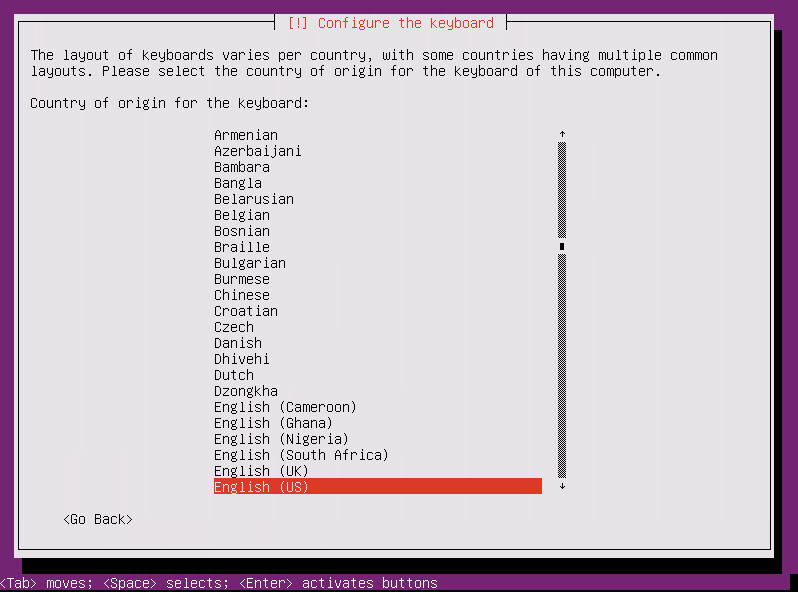
7.选择字符集编码-United States



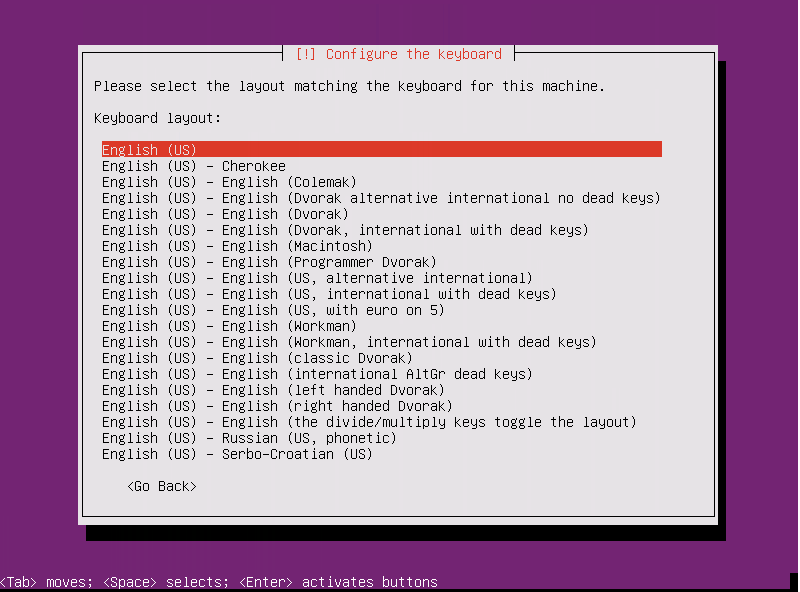
8.是否扫描和配置键盘，选择否-No



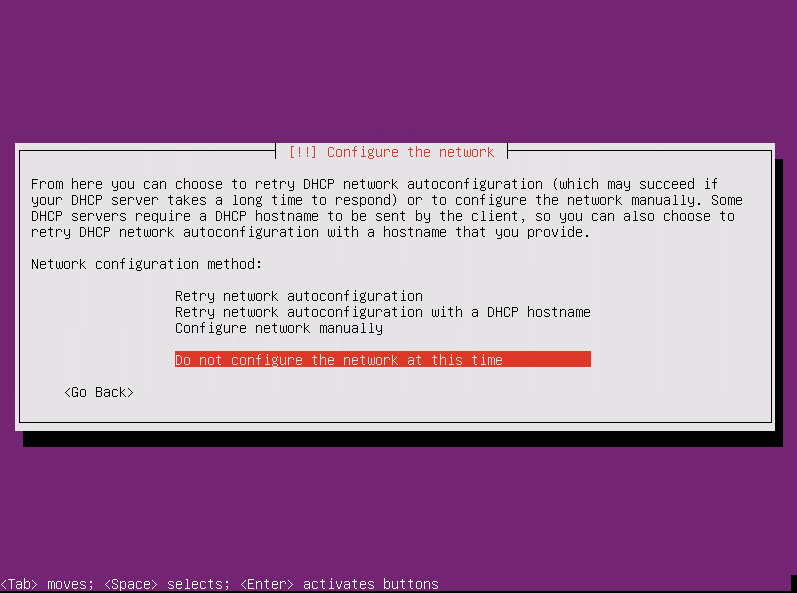
9.选择键盘类型-English (US)



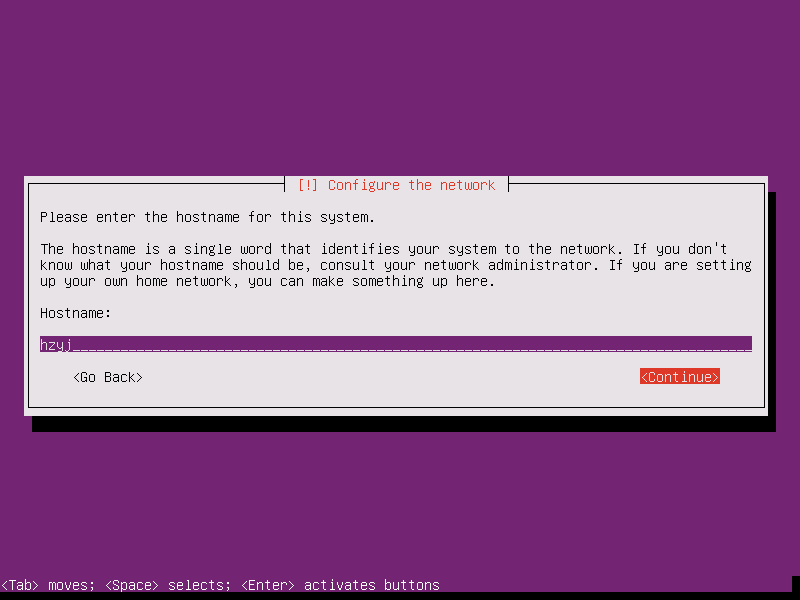
10.选择键盘布局-English (US)



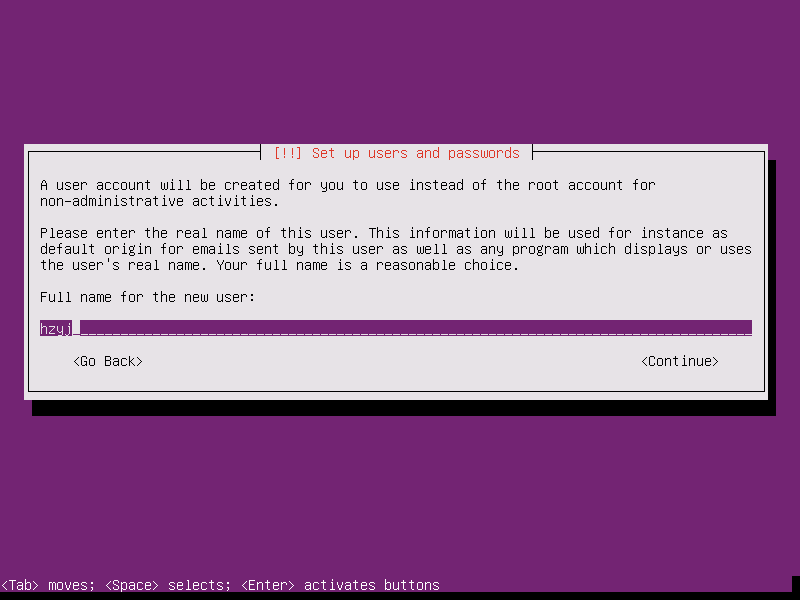
11.配置网络，因为内网没开DHCP ,暂时等装完系统在进行配置



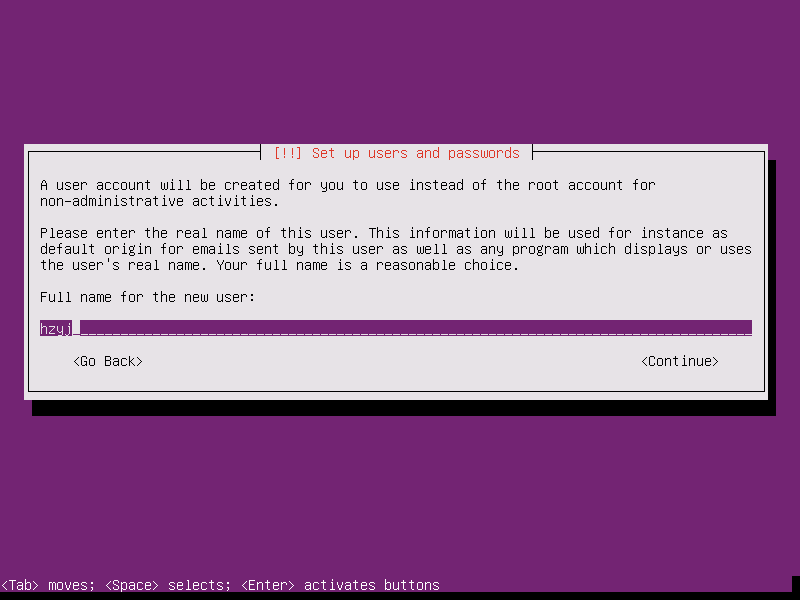
12.设置主机名称(自行设置，这里我设置为“hzyj”)-Continue



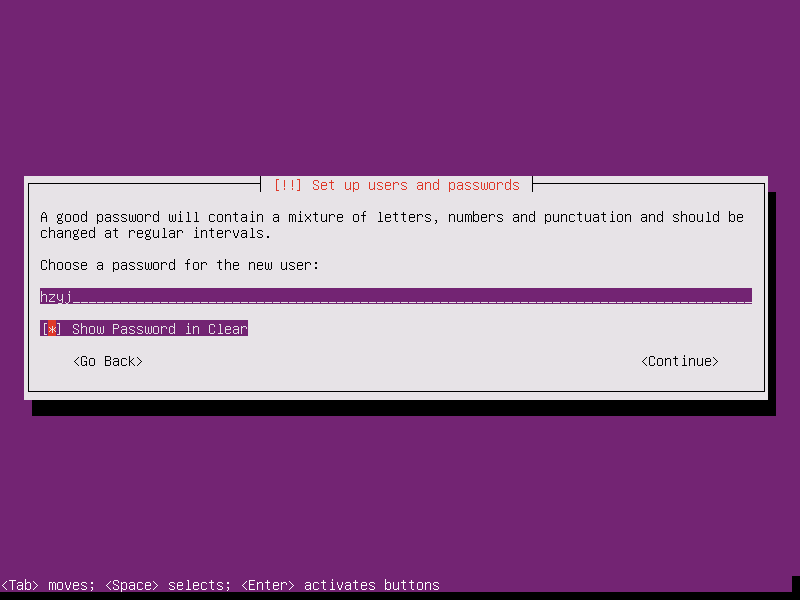
1. 设置用户全名(这里为“hzyj”)-Continue



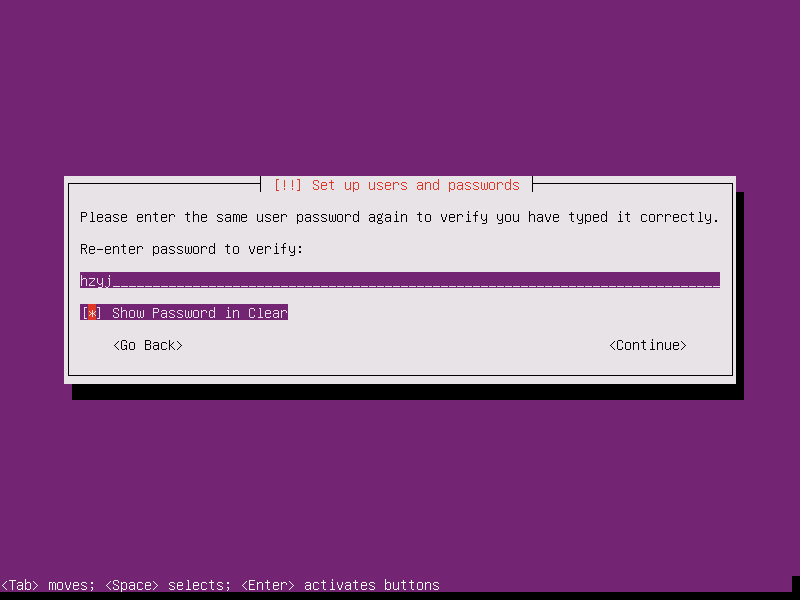
1. 设置登录账号(这里为“hzyj”)-Continue



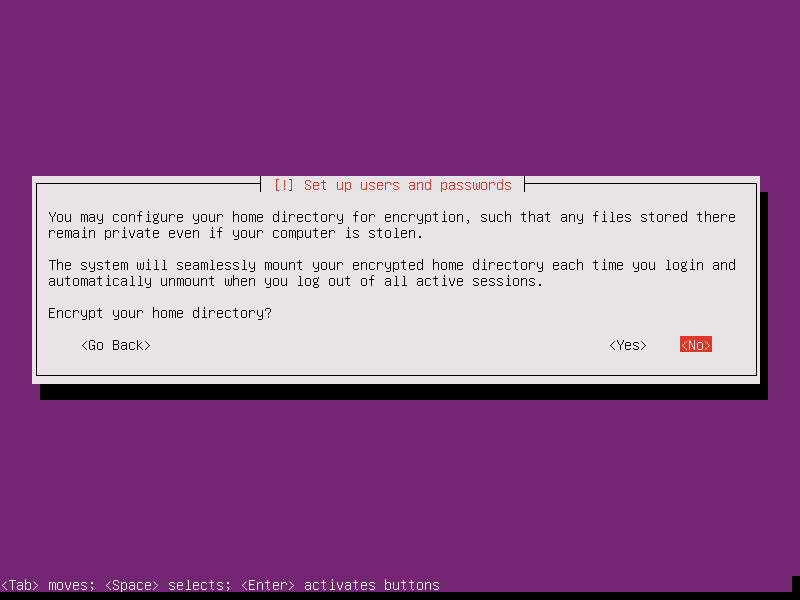
1. 设置登录密码(空格选择“Show Password in Clear”可以显示密码)-Continue；



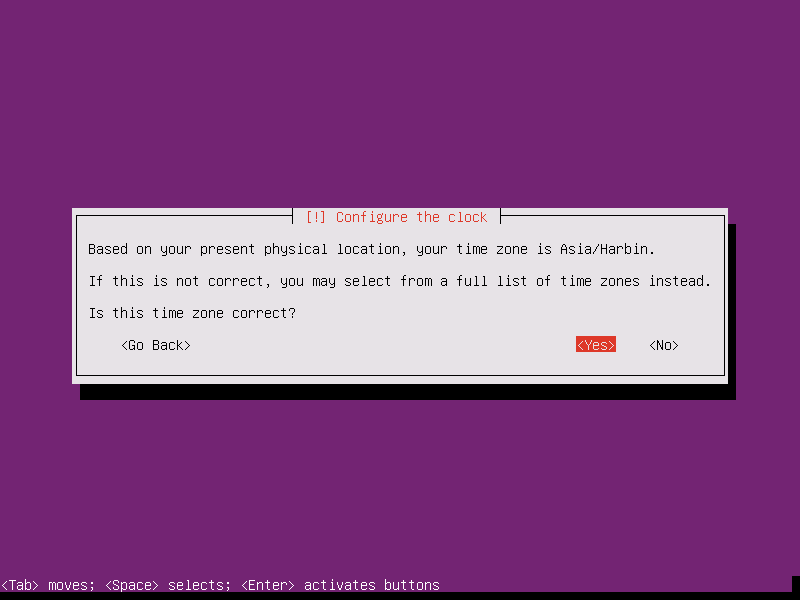
1. 重复上一步设置的登录密码-Continue；



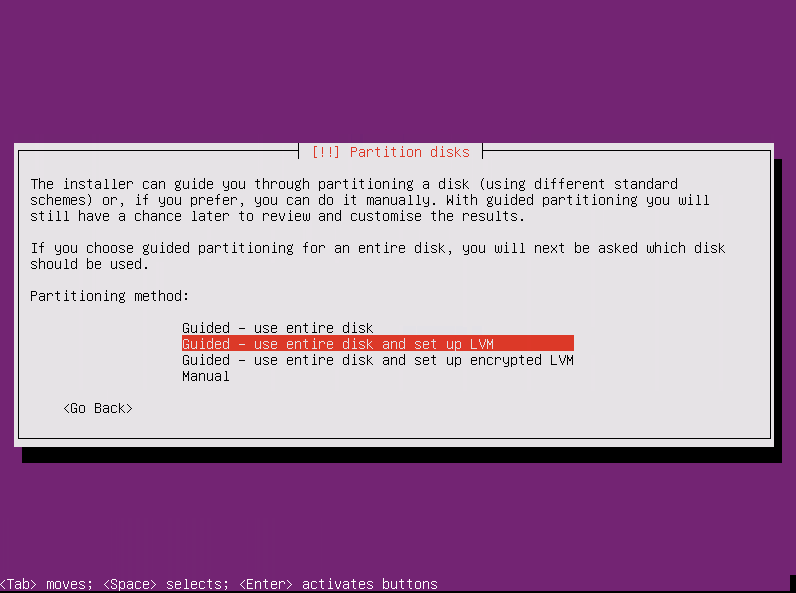
1. 是否使用强度弱的密码，我选择“Yes”
2. 是否加密主目录，选择“No”



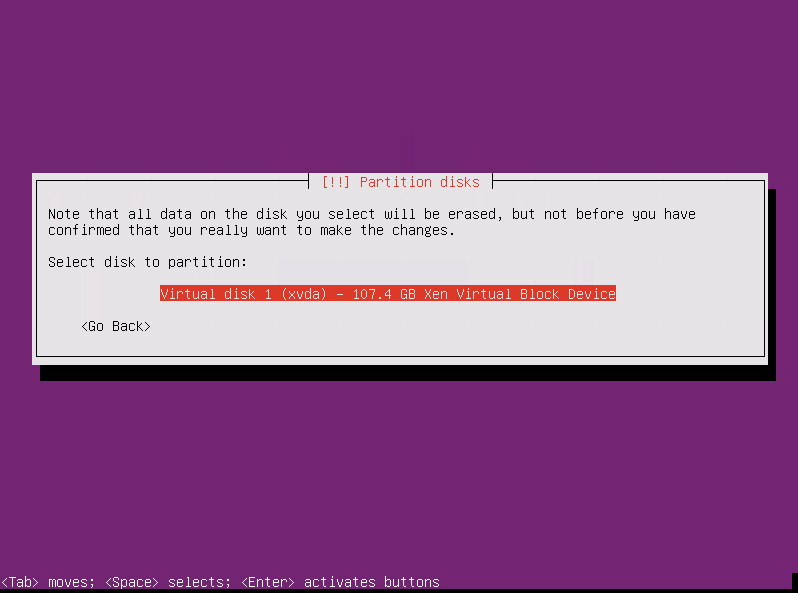
1. 时区是否正确，这里显示的是当前物理位置，我们先选择“Yes”。系统安装好之后我们可以更改为上海。



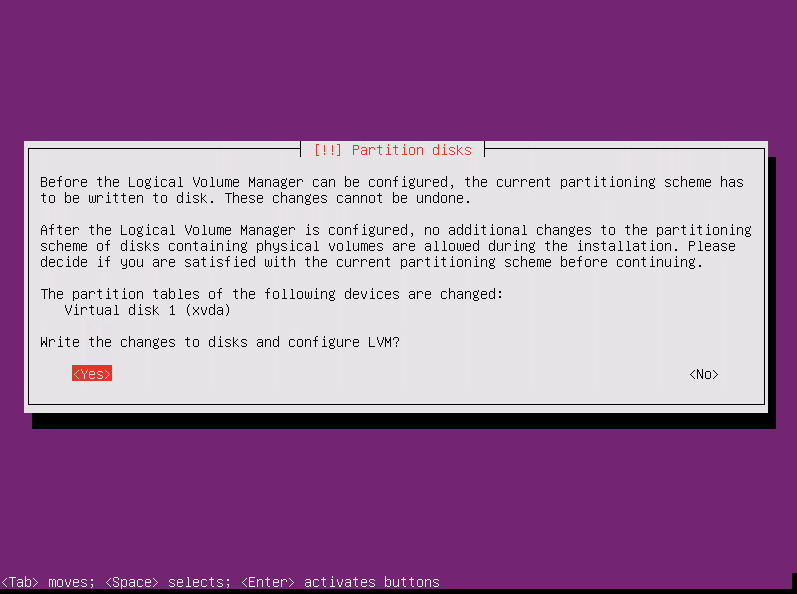
1. 选择分区方式(分区向导-使用整个磁盘)-“Guided - use entire disk and set up LVM”

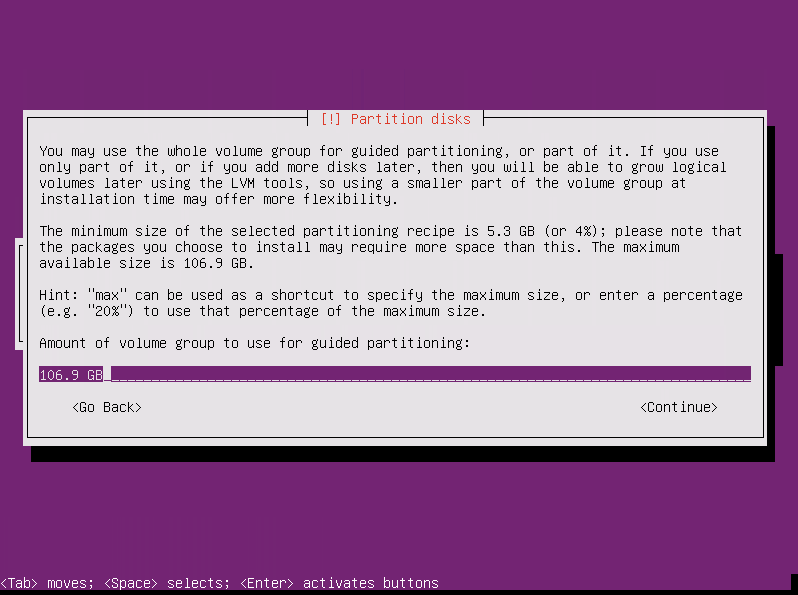


1. 选择要分区的磁盘(这里只有一块)

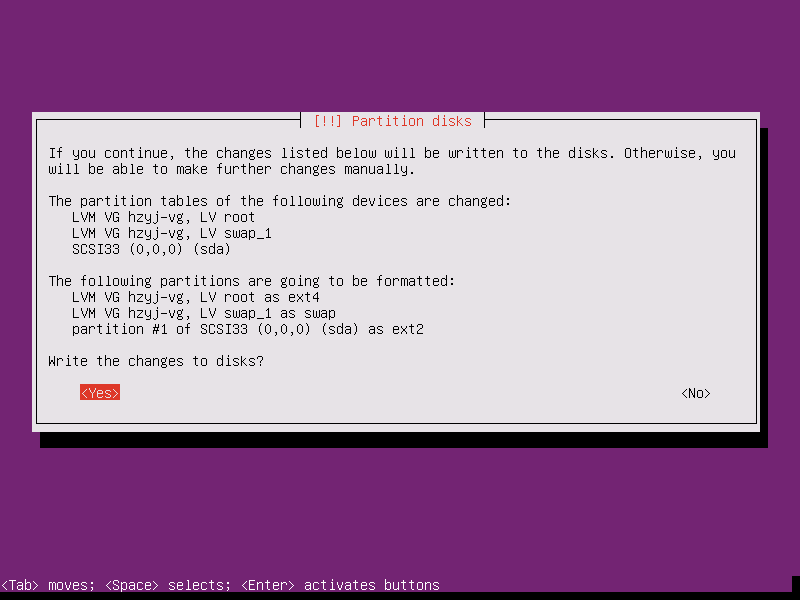


1. 是否将变更写入磁盘，选择是-Yes

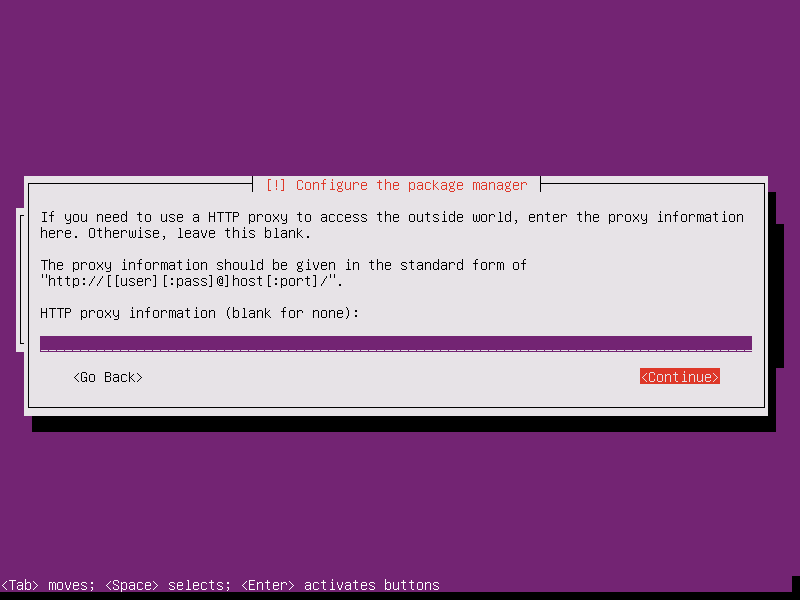




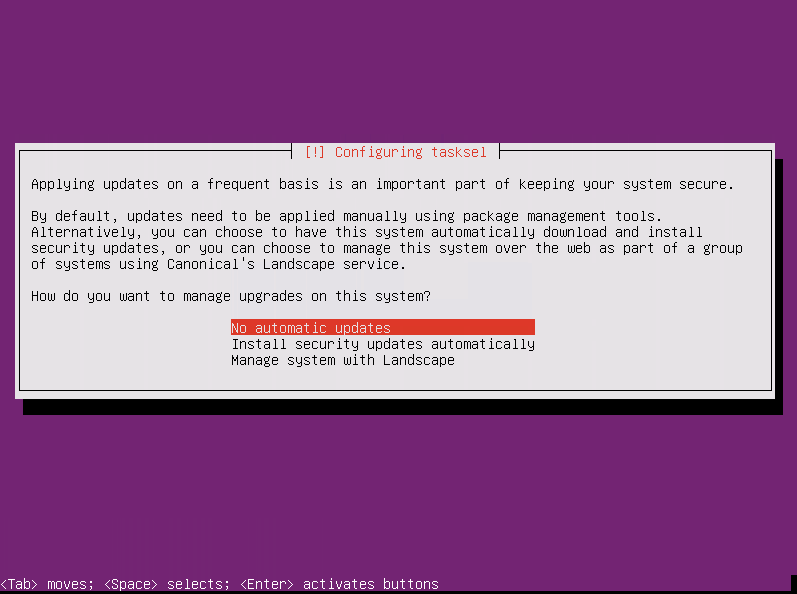
1. 选择Yes，直接格式化，等待安装完成



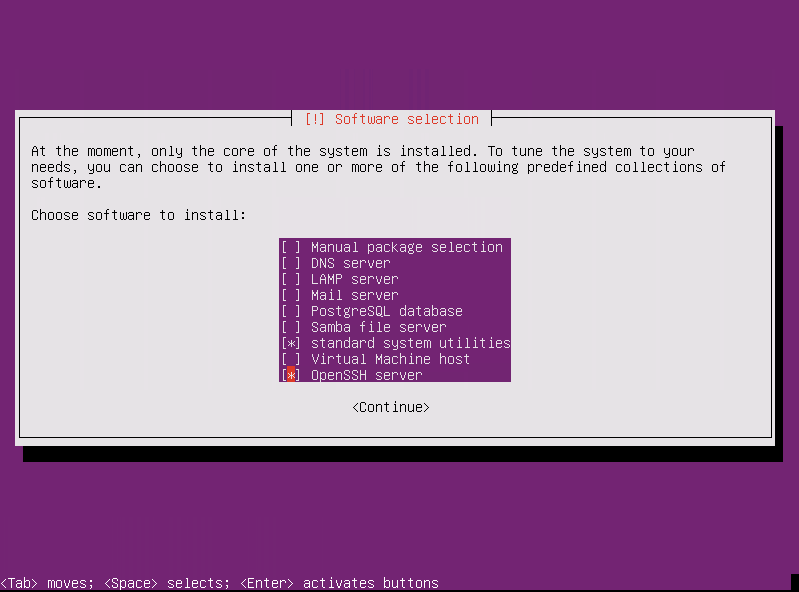
“如果您需要用HTTP代理来链接外部网咯，请在这里输入代理的信息。否则，请置空。”这里我置空，然后继续。



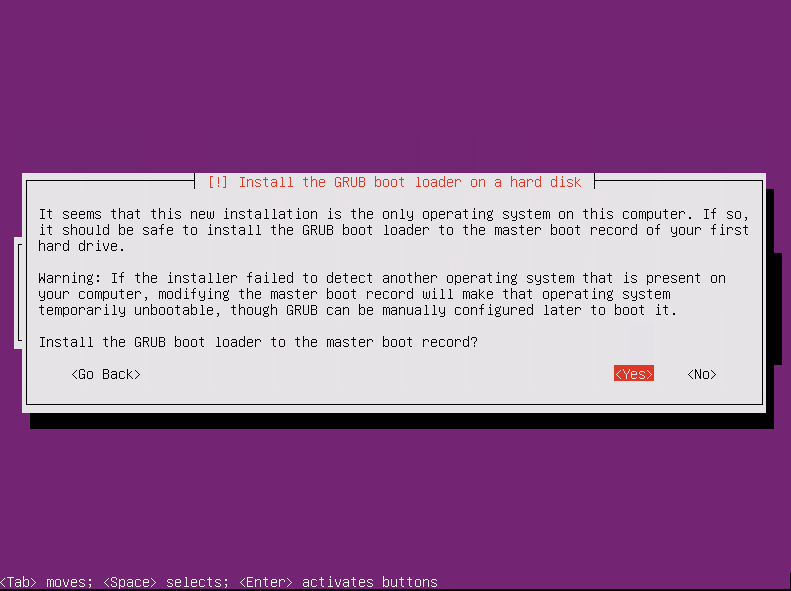
1. 选择升级方式，No automatic updates



1. 选择要安装的软件，OpenSSH Server可不加后面会安装



1. 是否安装GRUB引导程序，选择是-Yes

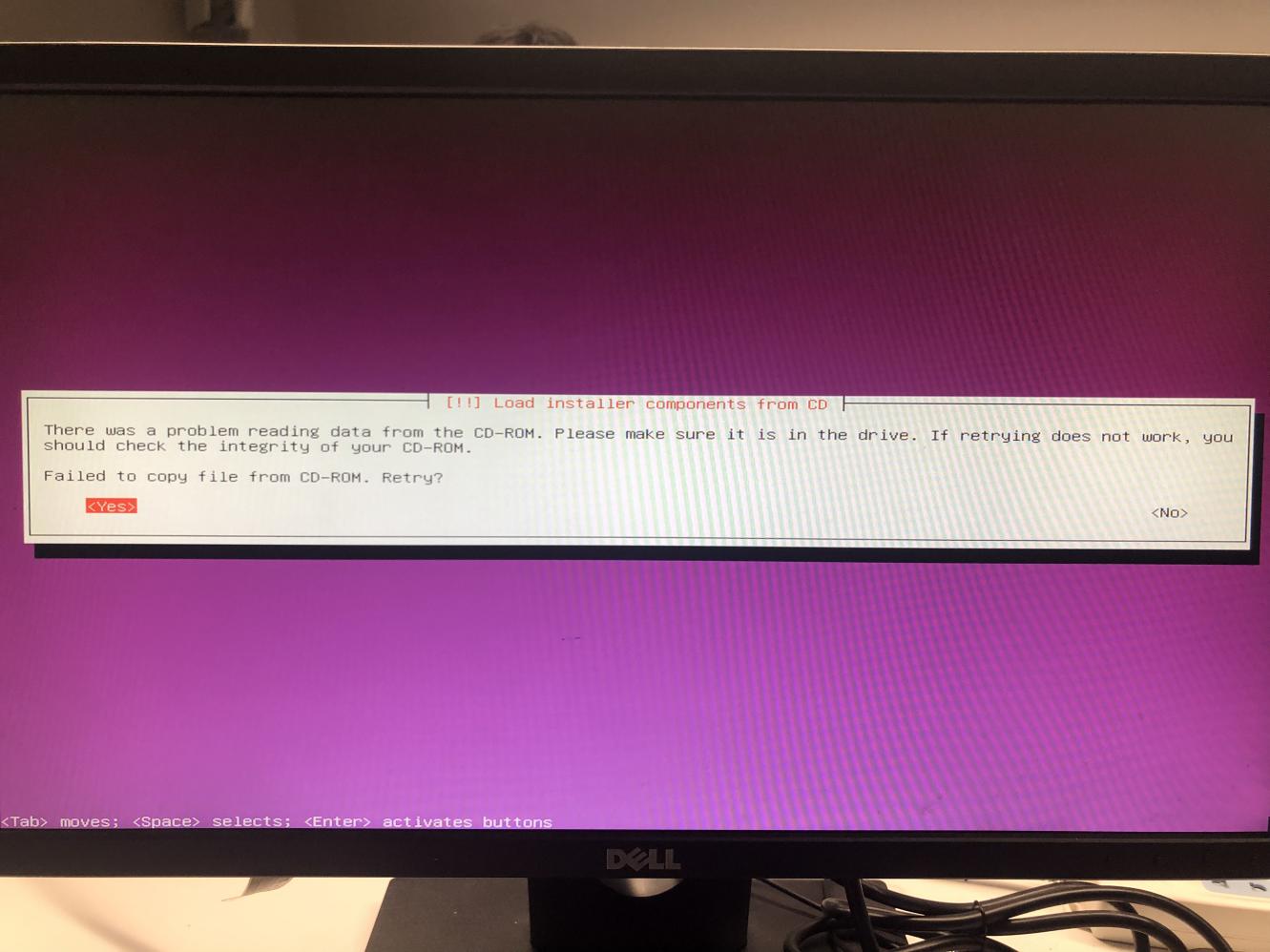


1. 完成安装，选择下一步-Continue



安装系统问题：

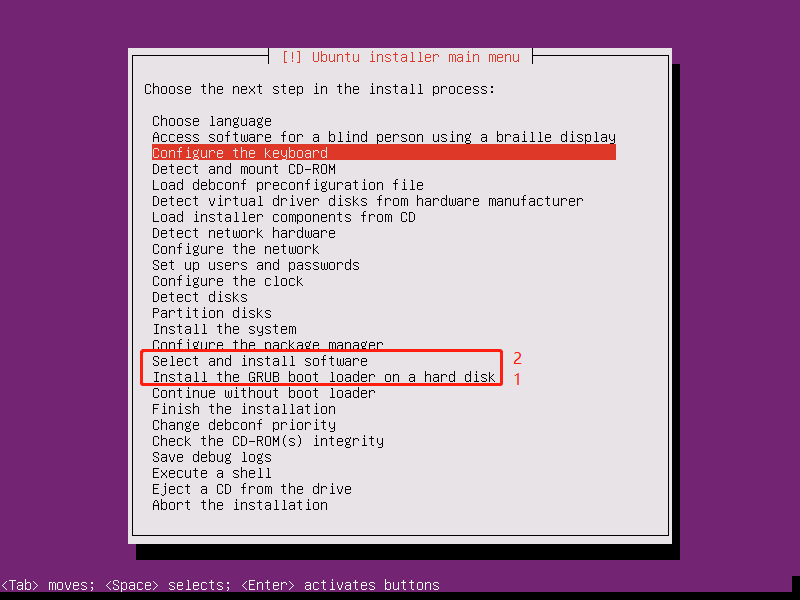
1.挂载U盘镜像参考离线安装中方法



mount /dev/sdb4 /mnt/

mount -t iso9660 -o loop mnt/ubuntu-16.04.4-server-amd64.iso /cdrom

2.磁盘分区好后出错，先安装GRUB然后在安装software



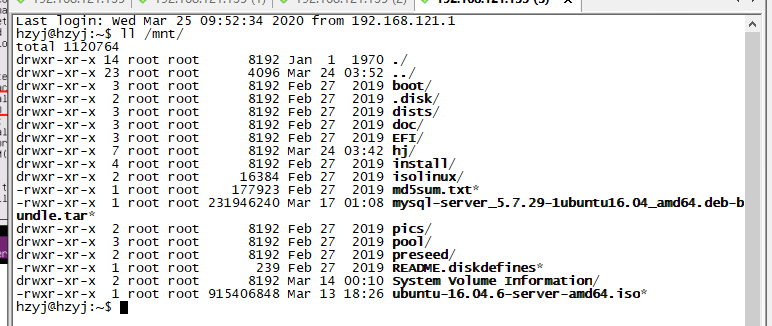
离线安装：

#挂载U盘

# ls /dev/sd\*

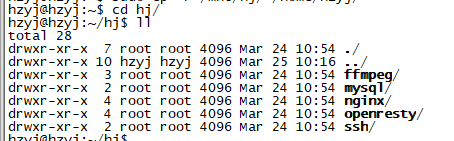


# sudo mount -t vfat /dev/sdb4 /mnt/



# sudo cp -r /mnt/hj/ /home/hzyj/

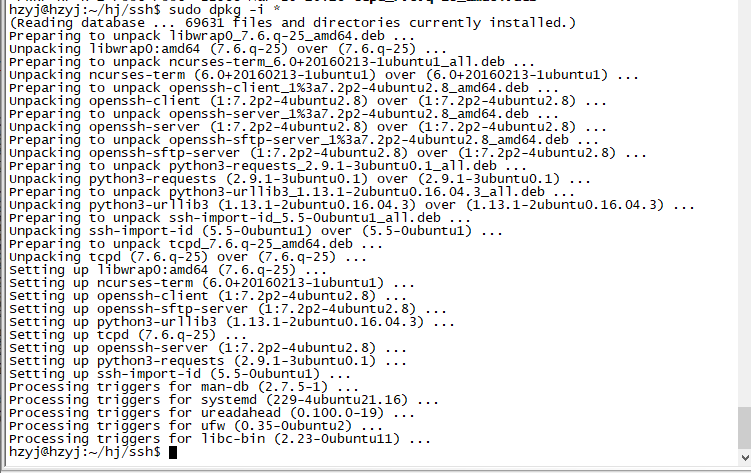
# cd hj



1. 安装ssh：

# cd ssh

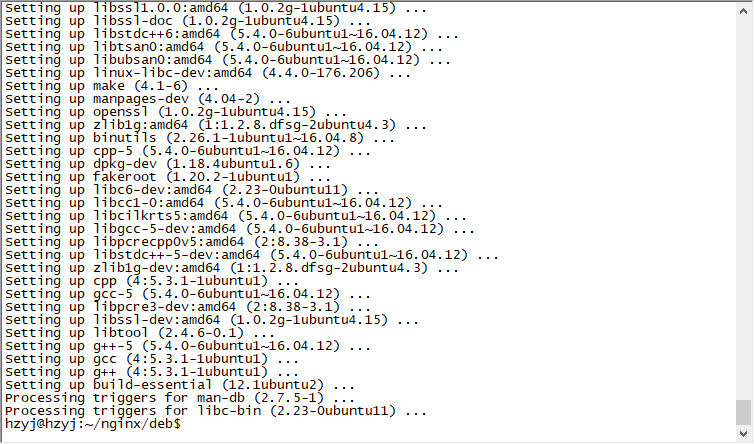
# sudo dpkg -i \*



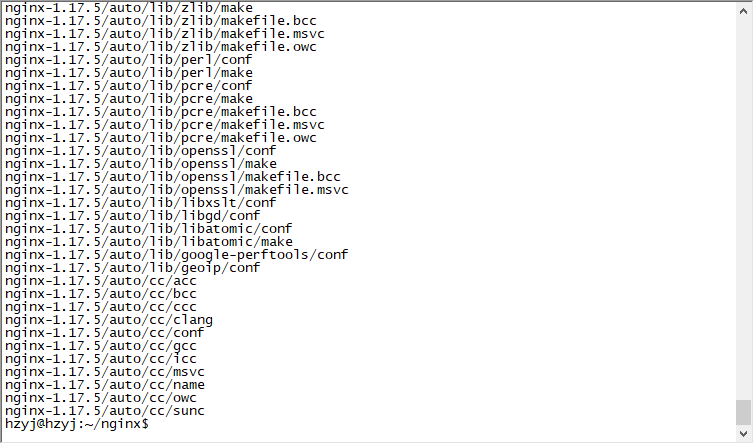
1. 安装Nginx、nginx-http-flv-module

# cd nginx/deb

# sudo dpkg -i \*



# sudo tar -zxvf nginx-1.17.5.tar.gz

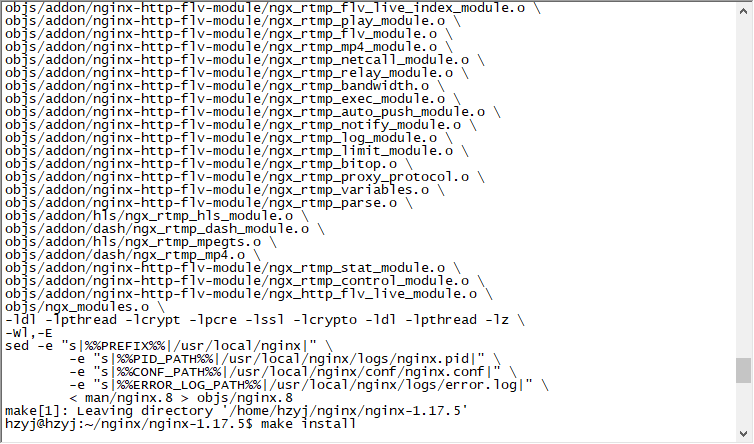


# cd nginx-1.17.5

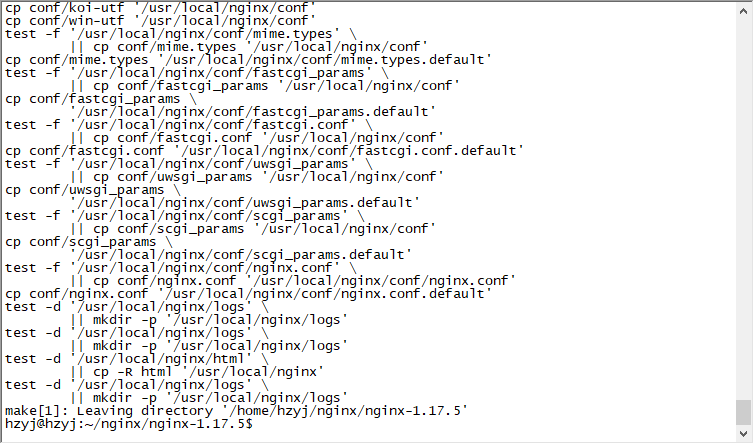
#sudo ./configure --add-module=../nginx-http-flv-module --with-http\_gzip\_static\_module



# sudo make



# sudo make install



配置nginx：

# cd /usr/local/nginx

# sudo vi conf/nginx.conf

文件地址、IP根据自己环境修改

worker\_processes 1; #should be 1 for Windows, for it doesn't support Unix domain socket

#worker\_processes auto; #from versions 1.3.8 and 1.2.5

#worker\_cpu\_affinity 0001 0010 0100 1000; #only available on FreeBSD and Linux

#worker\_cpu\_affinity auto; #from version 1.9.10

error\_log logs/error.log error;

#if the module is compiled as a dynamic module and features relevant

#to RTMP are needed, the command below MUST be specified and MUST be

#located before events directive, otherwise the module won't be loaded

#or will be loaded unsuccessfully when NGINX is started

#load\_module modules/ngx\_http\_flv\_live\_module.so;

events {

worker\_connections 4096;

}

http {

include mime.types;

default\_type application/octet-stream;

keepalive\_timeout 65;

server {

listen 80;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

location /live {

flv\_live on;

chunked\_transfer\_encoding on; #open 'Transfer-Encoding: chunked' response

add\_header 'Access-Control-Allow-Origin' '\*'; #add additional HTTP header

add\_header 'Access-Control-Allow-Credentials' 'true'; #add additional HTTP header

}

location /hls {

types {

application/vnd.apple.mpegurl m3u8;

video/mp2t ts;

}

root /home/hzyj/test/;

add\_header 'Cache-Control' 'no-cache';

}

location /dash {

root /home/hzyj/test/;

add\_header 'Cache-Control' 'no-cache';

}

location /stat {

rtmp\_stat all;

rtmp\_stat\_stylesheet stat.xsl;

}

location /stat.xsl {

root /home/hzyj/test; #specify in where stat.xsl located

}

#if JSON style stat needed, no need to specify

#stat.xsl but a new directive rtmp\_stat\_format

#location /stat {

# rtmp\_stat all;

# rtmp\_stat\_format json;

#}

location /control {

rtmp\_control all; #configuration of control module of rtmp

}

}

}

rtmp\_auto\_push on;

rtmp\_auto\_push\_reconnect 1s;

rtmp\_socket\_dir /tmp;

rtmp {

out\_queue 4096;

out\_cork 8;

max\_streams 128;

timeout 15s;

drop\_idle\_publisher 15s;

log\_interval 5s; #interval used by log module to log in access.log, it is very useful for debug

log\_size 1m; #buffer size used by log module to log in access.log

server {

listen 1935;

server\_name 192.168.121.135; #±¾µØP

application myapp {

live on;

gop\_cache on; #open GOP cache for reducing the wating time for the first picture of video

}

application hls {

live on;

hls on;

wait\_key on;

hls\_path /home/hzyj/test/hls;

hls\_fragment 10s;

hls\_playlist\_length 60s;

hls\_continuous on;

hls\_cleanup on;

hls\_nested on;

}

application dash {

live on;

dash on;

dash\_path /home/hzyj/test/dash;

}

}

server {

listen 1936;

server\_name 192.168.121.135; #for prefix wildcard matching of virtual host name

application myapp {

live on;

gop\_cache on; #open GOP cache for reducing the wating time for the first picture of video

}

}

server {

listen 1937;

server\_name 192.168.121.135; #for completely matching of virtual host name

application myapp {

live on;

gop\_cache on; #open GOP cache for reducing the wating time for the first picture of video

}

}

}

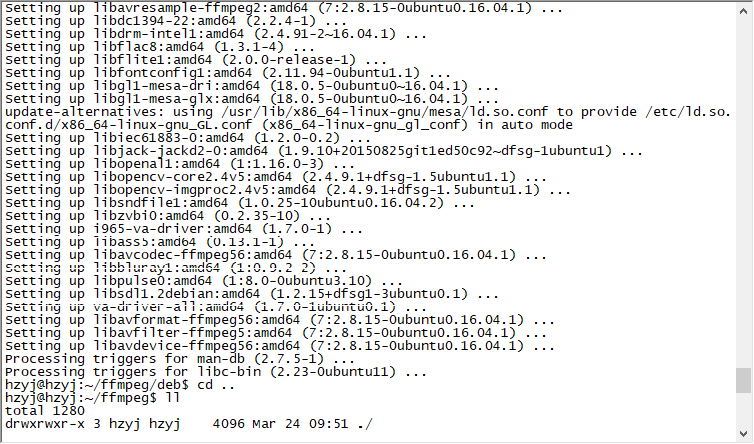
# sudo /usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf

# sudo sbin/nginx -s reload

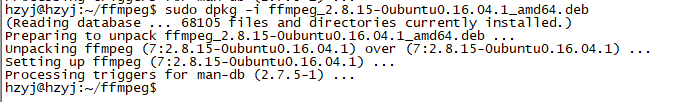
1. 安装ffmpeg

# cd ffmpeg/deb/

# sudo dpkg -i \* && cd ..

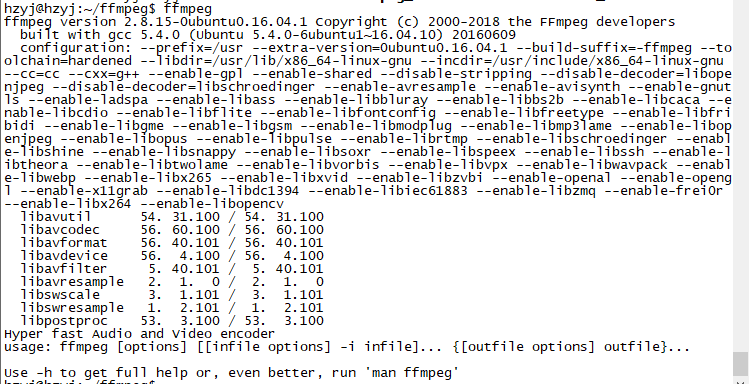


# sudo dpkg -i ffmpeg\_2.8.15-0ubuntu0.16.04.1\_amd64.deb



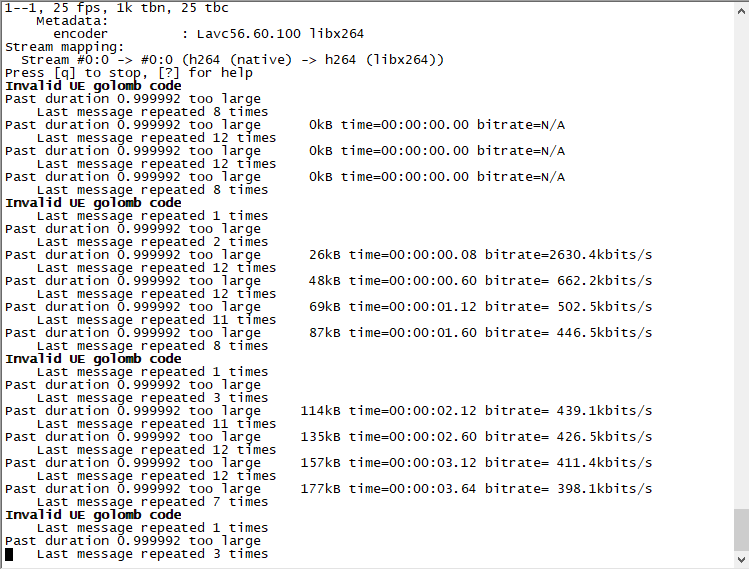
#检查安装版本

# ffmpeg



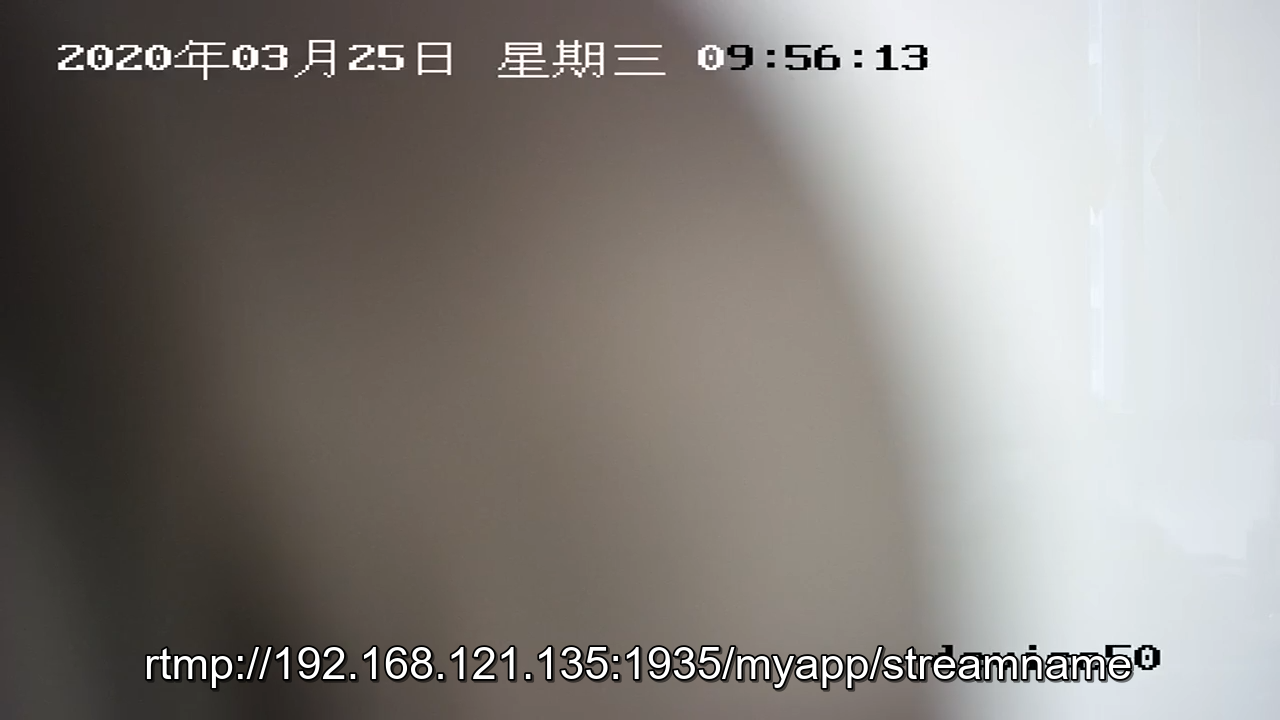
#启动推流RTMP模式

# ffmpeg -re -rtsp\_transport tcp -i rtsp://admin:admin123@192.168.100.50/h264/ch1/sub/av\_stream -vcodec libx264 -vprofile baseline -acodec aac -ar 44100 -strict -2 -ac 1 -f flv -s 1280x720 -q 10 rtmp://192.168.121.135:1935/myapp/streamname

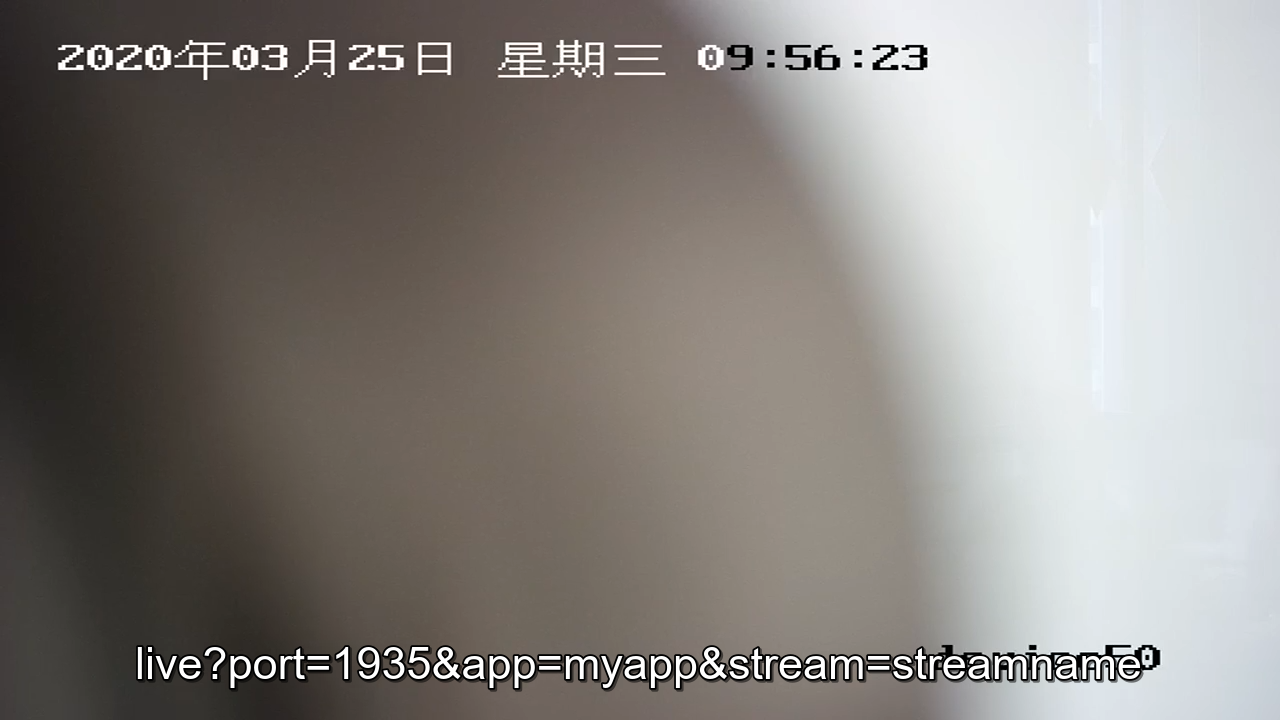


实时浏览：打开VLC----网络串流协议：

rtmp://192.168.121.135:1935/myapp/streamname

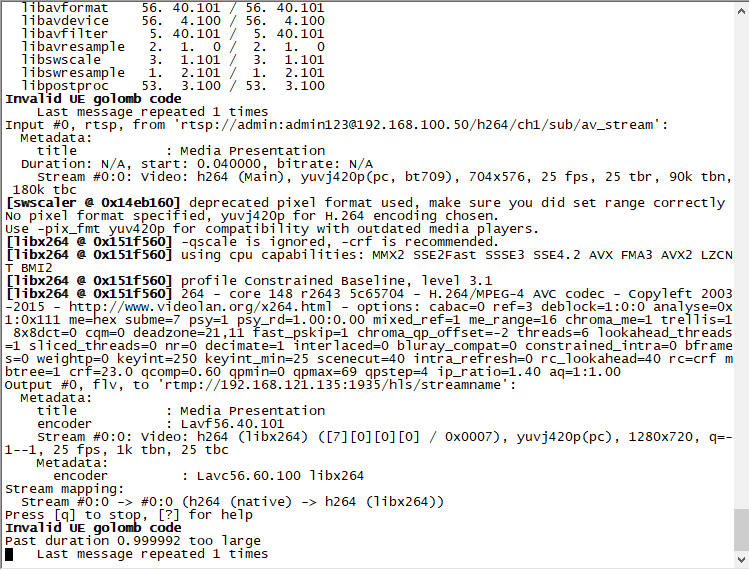


<http://192.168.121.135/live?port=1935&app=myapp&stream=streamname>



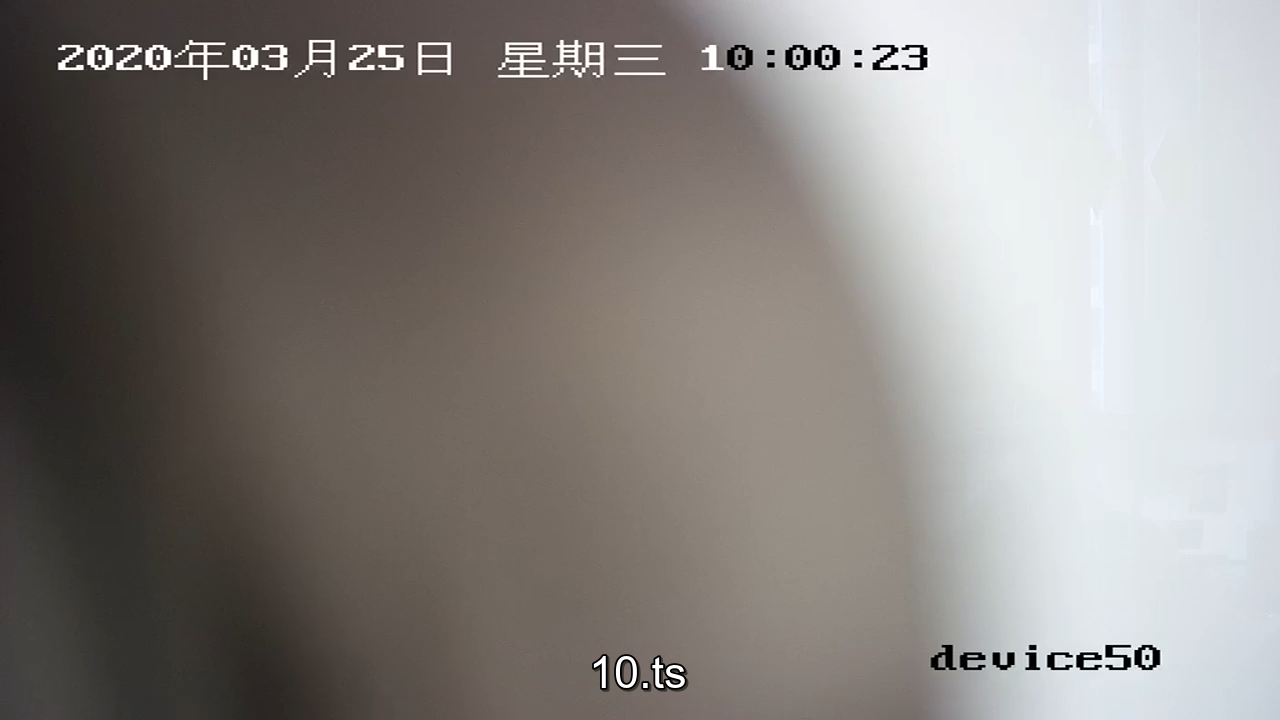
#启动HLS模式

# ffmpeg -re -rtsp\_transport tcp -i rtsp://admin:admin123@192.168.100.50/h264/ch1/sub/av\_stream -vcodec libx264 -vprofile baseline -acodec aac -ar 44100 -strict -2 -ac 1 -f flv -s 1280x720 -q 10 rtmp://192.168.121.135:1935/hls/streamname

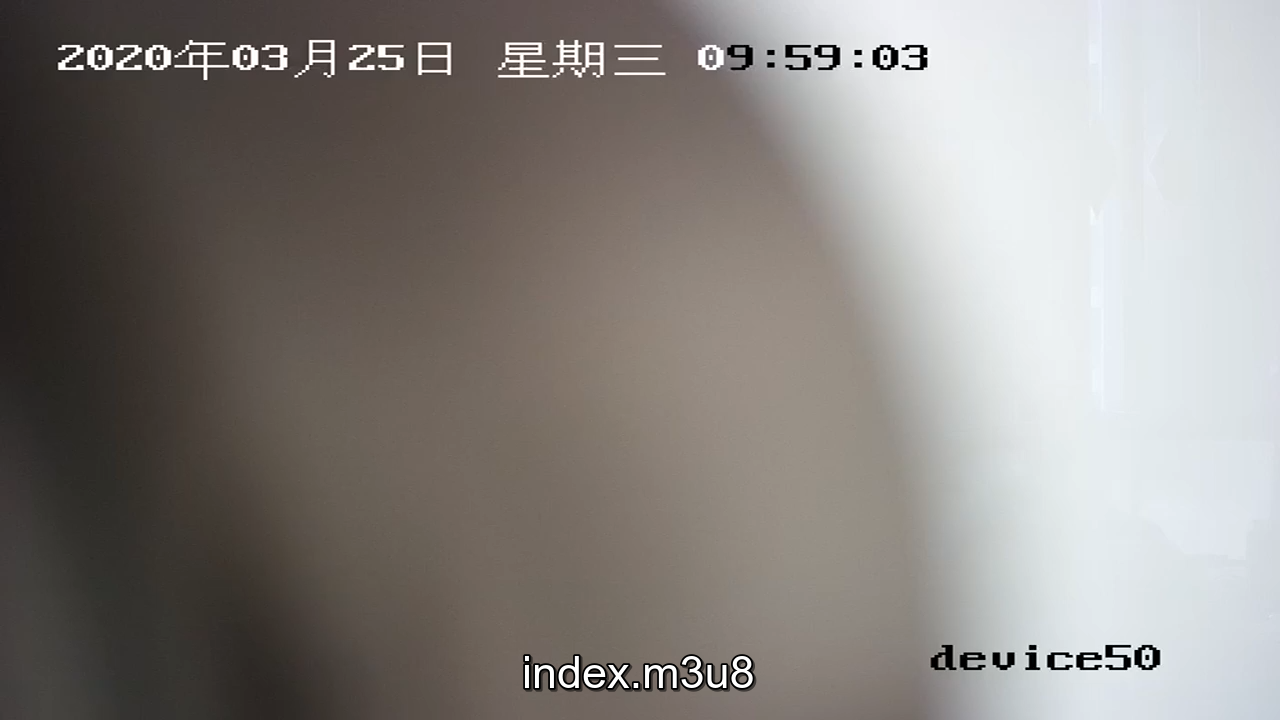


实时浏览：打开VLC----网络串流协议：

<http://192.168.121.135/hls/streamname/10.ts>

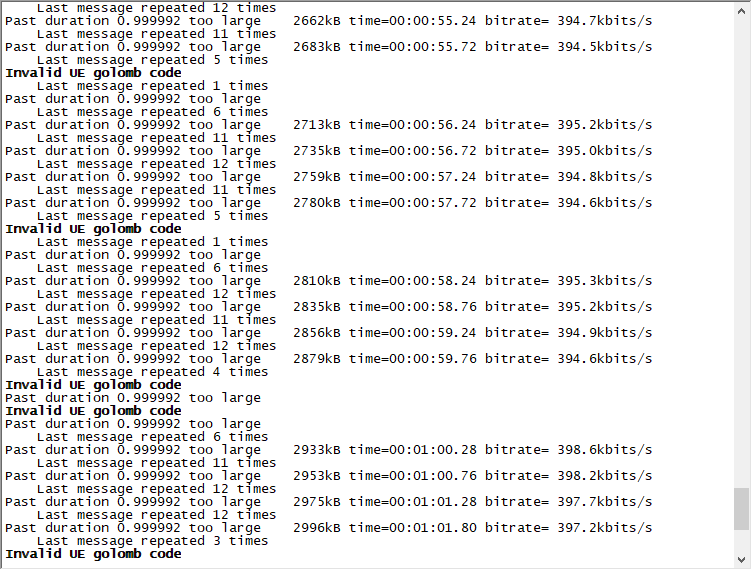


<http://192.168.121.135/hls/streamname/index.m3u8>



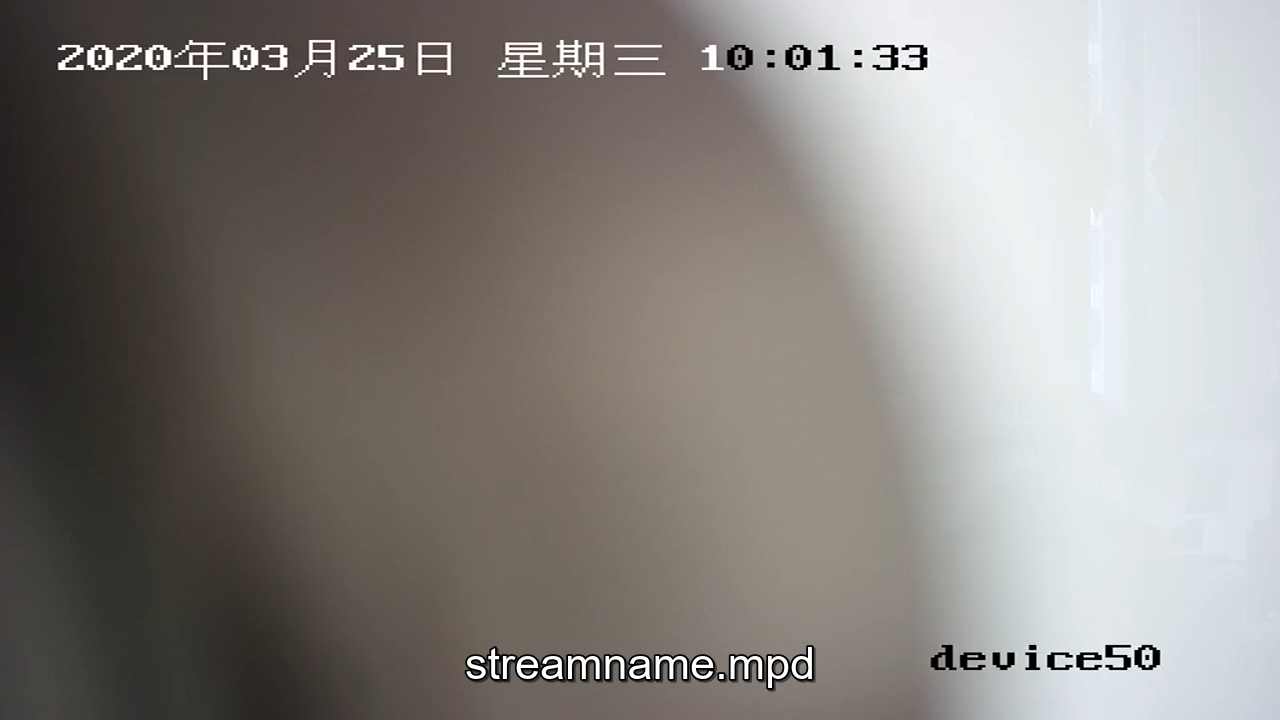
#启动DASH模式

# ffmpeg -re -rtsp\_transport tcp -i rtsp://admin:admin123@192.168.100.50/h264/ch1/sub/av\_stream -vcodec libx264 -vprofile baseline -acodec aac -ar 44100 -strict -2 -ac 1 -f flv -s 1280x720 -q 10 rtmp://192.168.121.135:1935/dash/streamname



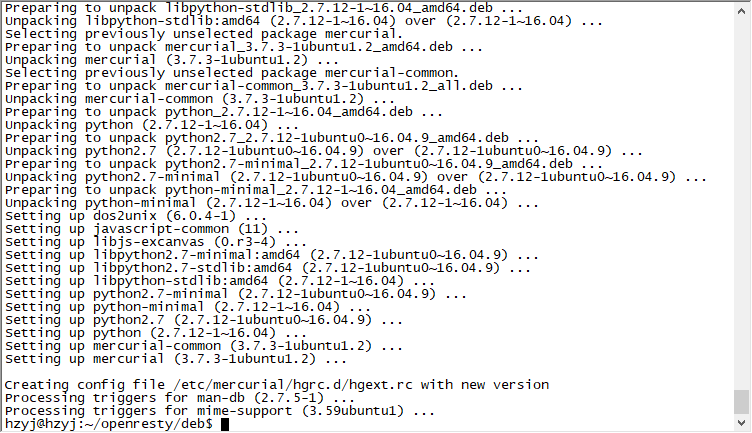
实时浏览：打开VLC----网络串流协议：

http://192.168.121.135/dash/streamname.mpd

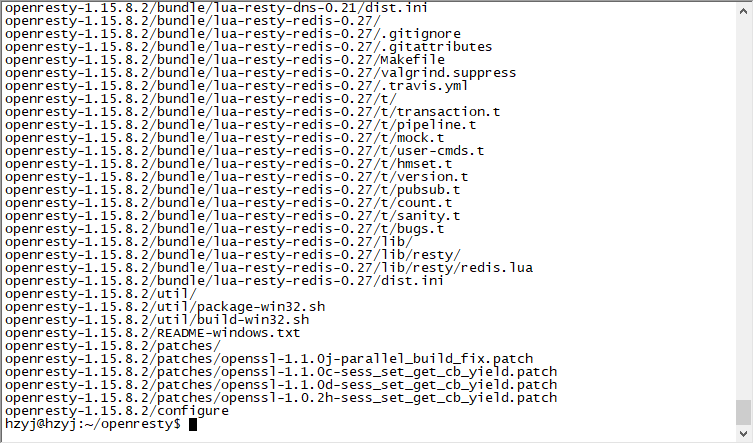


1. openresty

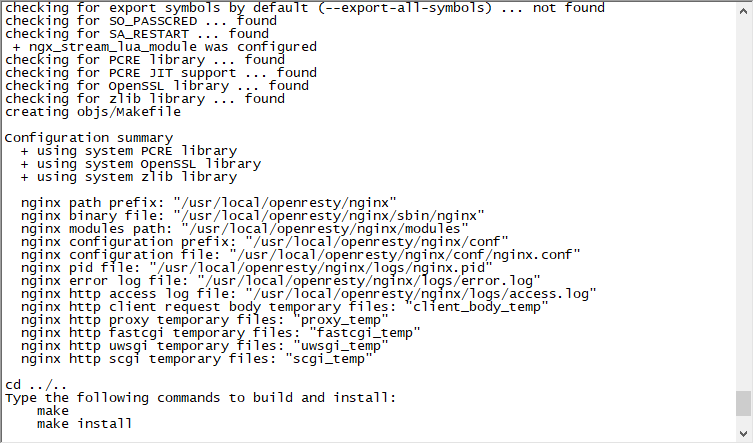
# cd openresty/deb/ && sudo dpkg -i \* && sudo dpkg -i \*



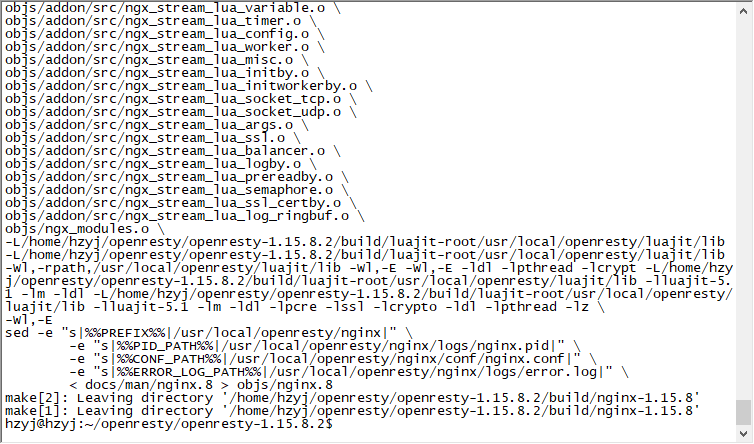
# cd .. && sudo tar -xvf openresty-1.15.8.2.tar.gz



# cd openresty-1.15.8.2/ && ./configure



# sudo make



# sudo make install



# cd /usr/local/openresty/nginx && sudo mkdir html/download && sudo cp -r /home/hzyj/hj/openresty/lua ./conf

# sudo vi conf/nginx.conf

将文本信息替换为：

user root;

worker\_processes 20;

error\_log logs/error.log notice;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

server {

listen 8082;

server\_name localhost;

# 最大允许上传的文件大小

client\_max\_body\_size 200m;

location / {

root html;

index index.html index.htm;

}

set $store\_dir "/usr/local/openresty/nginx/html/download/"; # 文件存储路径

# 文件上传接口：http://xxx:8082/upfile

location /upfile {

default\_type 'text/plain';

content\_by\_lua\_file conf/lua/upload.lua; # 实现文件上传的逻辑

}

# 文件下载入口: http://xxx:8082/download

location /download {

autoindex on;

autoindex\_localtime on;

root html;

index index.html;

}

# redirect server error pages to the static page /50x.html

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

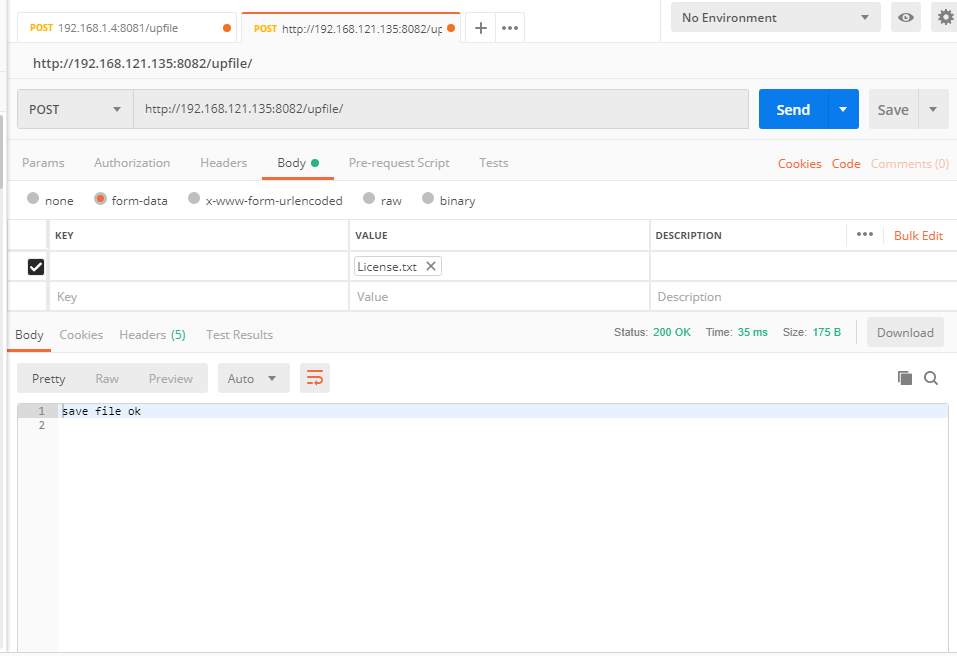
}

}

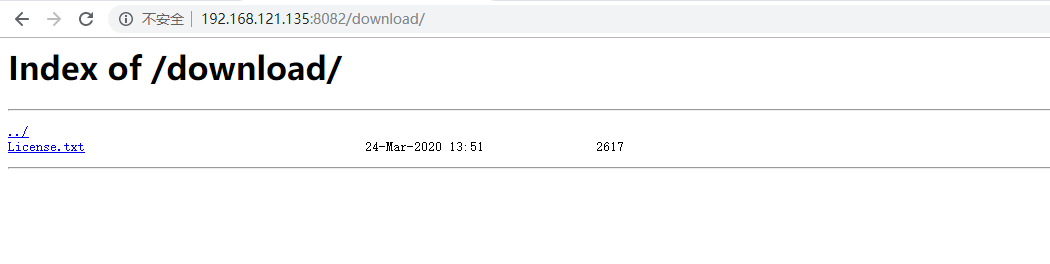
# sudo /usr/local/openresty/nginx/sbin/nginx -c /usr/local/openresty/nginx/conf/nginx.conf

# sudo sbin/nginx -s reload

上传:使用PostMan或其他工具上传测试。

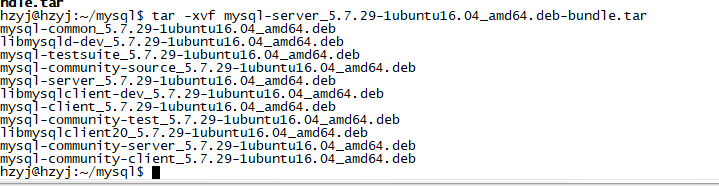


下载:[http://192.168.121.135:8082/download/](http://192.168.121.129:8082/download/)

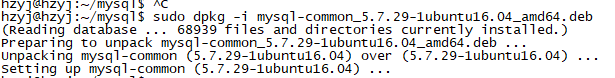


5.Mysql 5.7.29安装：

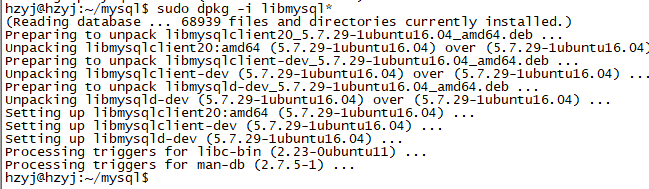
# sudo tar -xvf mysql-server\_5.7.29-1ubuntu16.04\_amd64.deb-bundle.tar

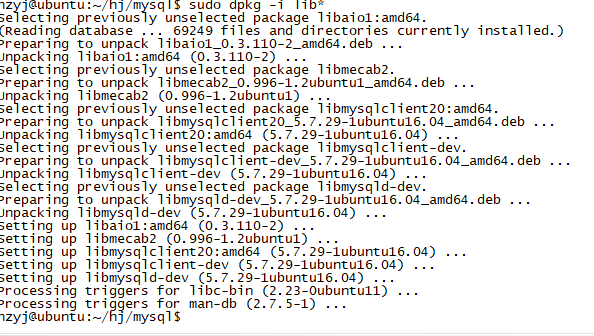


# sudo dpkg -i mysql-common\_5.7.29-1ubuntu16.04\_amd64.deb

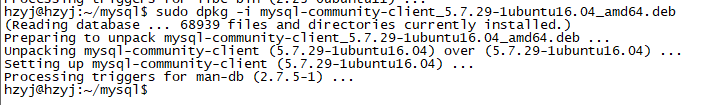


# sudo dpkg -i lib\*

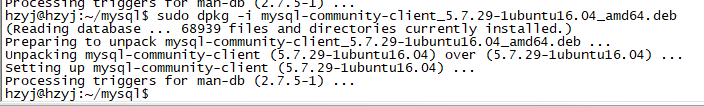




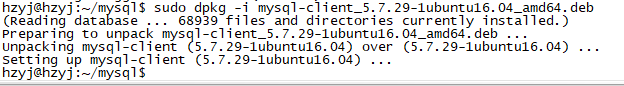
# sudo dpkg -i mysql-community-client\_5.7.29-1ubuntu16.04\_amd64.deb



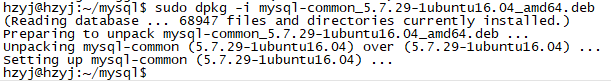
# sudo dpkg -i mysql-community-client\_5.7.29-1ubuntu16.04\_amd64.deb



# sudo dpkg -i mysql-client\_5.7.29-1ubuntu16.04\_amd64.deb

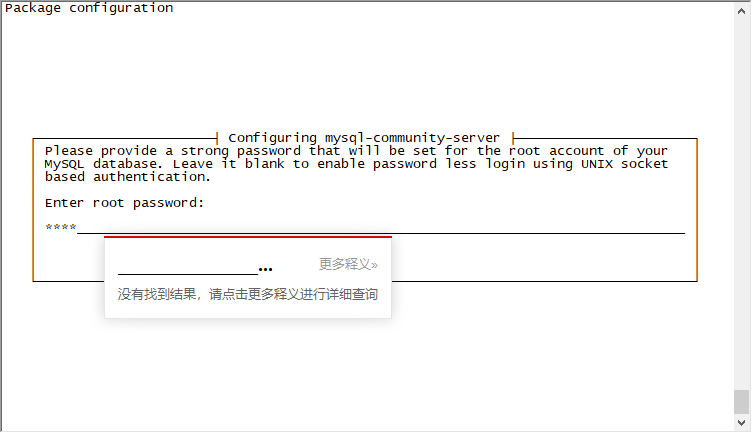


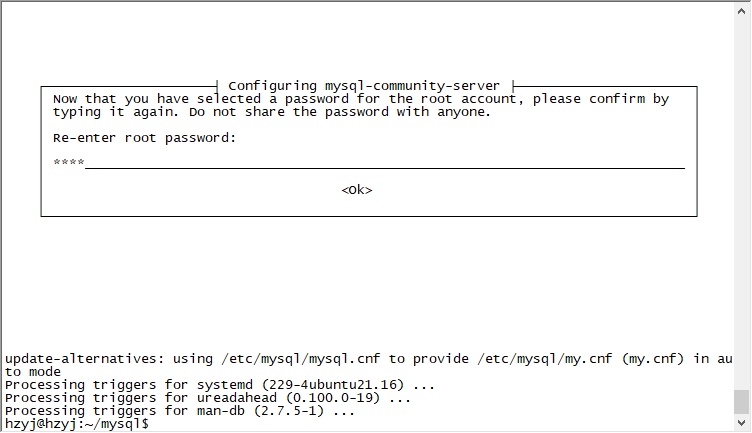
# sudo dpkg -i mysql-common\_5.7.29-1ubuntu16.04\_amd64.deb



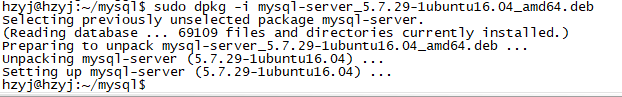
# sudo dpkg -i mysql-community-server\_5.7.29-1ubuntu16.04\_amd64.deb

#输入密码，确认密码

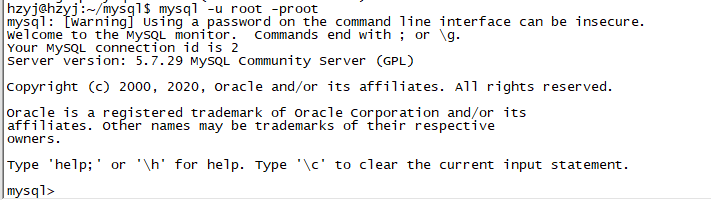




# sudo dpkg -i mysql-server\_5.7.29-1ubuntu16.04\_amd64.deb



#查询版本



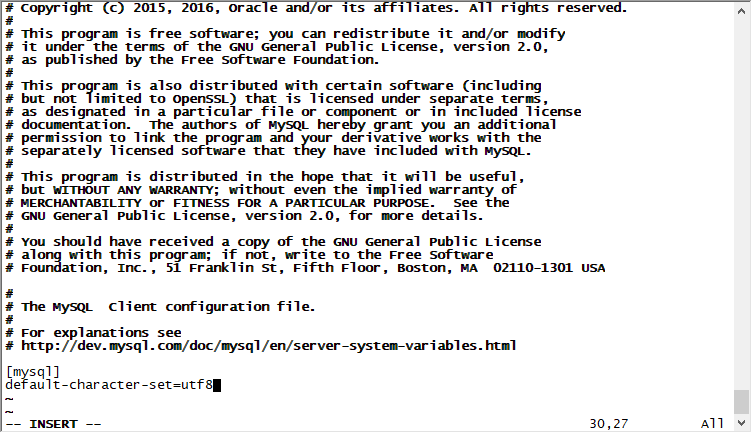
设置utf-8编码：

输入下面的命令，打开第一个配置文件

sudo vi /etc/mysql/conf.d/mysql.cnf

在 [mysql] 标签的下一行添加下面的配置

default-character-set=utf8

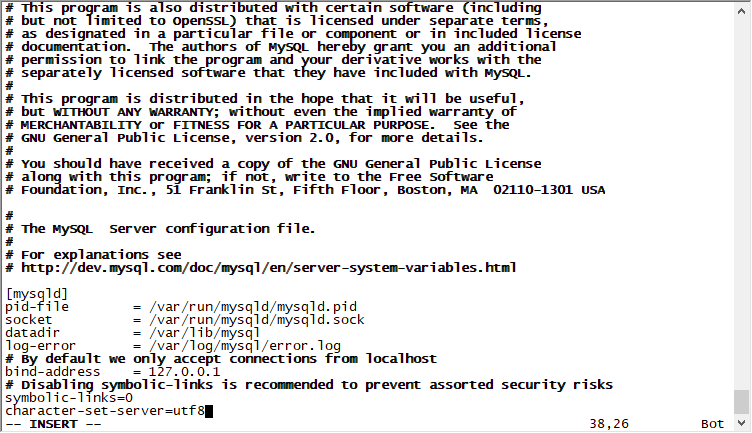


输入下面的命令，打开第二个配置文件

sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

找到 [mysqld] 标签，在其下一行添加下面的配置

character-set-server=utf8

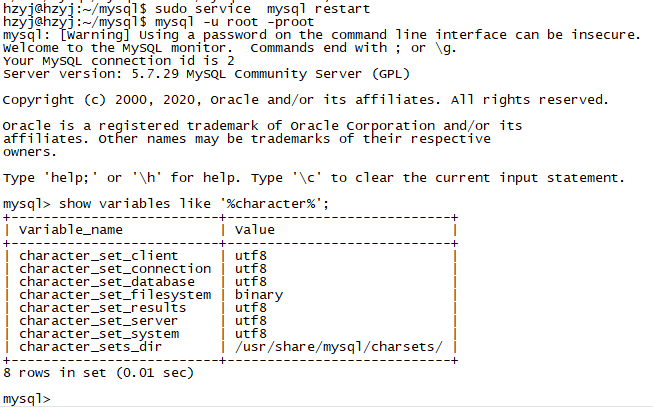


配置文件修改成功之后，输入下面的命令重启mysql服务

# sudo service mysql restart

# mysql -u root -proot

# show variables like '%character%';



在线安装：

Openssl:

sudo apt-get install openssl

sudo apt-get install libssl-dev

Zlib:

sudo apt-get install zlib1g-dev

Gcc:

sudo apt-get install build-essential

apt-get install libtool

sudo apt-get install libpcre3 libpcre3-dev

# wget <http://nginx.org/download/nginx-1.17.5.tar.gz>

# tar -zxvf nginx-1.17.5.tar.gz

cd nginx-1.17.5/

Sudo apt-get install git

Git clone https://github.com/winshining/nginx-http-flv-module.git

./configure --add-dynamic-module=/path/to/nginx-http-flv-module

make

make install

Nginx配置信息，查看环境软件文件

安装

Ffmpeg

Sudo dpgk -i ffmpeg\_2.8.15-0ubuntu0.16.04.1\_amd64.deb

Sudo apt-get install -f

Mysql

# tar -xvf mysql-server\_5.7.29-1ubuntu16.04\_amd64.deb-bundle.tar

# sudo dpkg -i mysql-common\_5.7.29-1ubuntu16.04\_amd64.deb

# sudo dpkg -i libmysql\*

# sudo apt-get install libaio1

# sudo dpkg -i mysql-community-client\_5.7.29-1ubuntu16.04\_amd64.deb

# sudo dpkg -i mysql-community-client\_5.7.29-1ubuntu16.04\_amd64.deb

# sudo dpkg -i mysql-client\_5.7.29-1ubuntu16.04\_amd64.deb

# sudo apt-get install libmecab2\_0

# sudo dpkg -i mysql-common\_5.7.29-1ubuntu16.04\_amd64.deb

# sudo dpkg -i mysql-community-server\_5.7.29-1ubuntu16.04\_amd64.deb

# sudo dpkg -i mysql-server\_5.7.29-1ubuntu16.04\_amd64.deb

设置utf-8编码：

输入下面的命令，打开第一个配置文件

sudo vi /etc/mysql/conf.d/mysql.cnf

在 [mysql] 标签的下一行添加下面的配置

default-character-set=utf8

输入下面的命令，打开第二个配置文件

sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

找到 [mysqld] 标签，在其下一行添加下面的配置

character-set-server=utf8

配置文件修改成功之后，输入下面的命令重启mysql服务

# sudo service mysql restart

# mysql -u root -proot

# show variables like '%character%';

安装openresty

174 sudo apt-get install perl

175 sudo apt-get install dos2unix

176 sudo apt-get install mercurial

wget <https://openresty.org/download/openresty-1.15.8.2.tar.gz>

tar -xvf openresty-1.15.8.2.tar.gz

./configure

make

sudo make install

修改配置文件路径为：/usr/local/openresty/nginx

配置：<https://www.cnblogs.com/lujiango/p/9056680.html>