

启动相关服务

```
1 ./janus -b
2 sudo turnserver -o -c /usr/local/etc/turnserver.conf
3 nohup java -jar janus_dispatch-0.0.1-SNAPSHOT.jar >log.file &
```

新建用户

```
1 useradd -m ubuntu -s /bin/bash
2 passwd ubuntu
3 adduser ubuntu sudo
4 su ubuntu
5
6 scp -i ./证书.pem 目录/software.tar.gz root@服务器 ip 地址:/home/ubuntu/
```

安装基础软件

```
1 tar -zxvf software.tar.gz
2 cd ~
3 sudo apt update
4 sudo apt install aptitude
5
6 sudo aptitude install libjansson-dev libssl-dev libsrtp-dev libsofia-s
ipua-dev libglib2.0-dev \
7 libopus-dev libogg-dev libcurl4-openssl-dev \
8 liblua5.3-dev libconfig-dev pkg-config gengetopt libtool automake cmake
e unzip wget
9
10 sudo apt-get install apt-transport-https ca-certificates
11 sudo apt install cmake
12 sudo apt-get install gtk-doc-tools
```

libsrtp

```
1 cd ~/software/libsrtp-1.5.4/
2 ./configure --prefix=/usr/local/janus --enable-openssl
3 sudo make clean && make && make shared_library && sudo make install
```

usrctp

```
1 cd ~/software/usrctp/
2 ./bootstrap
3 ./configure --prefix=/usr/local/janus
```

```
4 make && sudo make install
```

libwebsockets

```
1 cd ~/software/libwebsockets-3.0.1
2 sudo apt install cmake
3 mkdir build && cd build
4 cmake -DCMAKE_INSTALL_PREFIX:PATH=--prefix=/usr/local/janus -DCMAKE_C_
  FLAGS="-fpic" ..
5 sudo make clean && make && sudo make install
```

libconfig

```
1 cd ~/software/libconfig-1.7.2
2 ./configure --prefix=/usr/local/janus
3 sudo make clean && make && sudo make install
```

安装libffi-3.3

```
1 tar -zxvf libffi-3.3.tar.gz
2 cd libffi-3.3/
3 ./configure
4 make && sudo make install
```

安装libmount

```
1 sudo apt install libmount-dev
```

安装glib-2.54

```
1 tar -xvf glib-2.54.0.tar.xz
2 cd glib-2.54.0/
3 ./autogen.sh
4 make -j8 && sudo make install
```

libnice

```
1 cd libnice-latest-release/
2 sh ./autogen.sh
```

```
3 ./configure --prefix=/usr/local/janus
4 sudo make install
5 完成后执行看看是不是自己需要的版本
6 cat /usr/local/janus/lib/pkgconfig/nice.pc
```

libmicrohttpd

```
1 cd ~/software/libmicrohttpd-0.9.70/
2 ./configure --enable-shared
3 sudo make -j8 && sudo make install
4
5 #添加动态库查找路径
6 sudo vim /etc/ld.so.conf
7 添加一行:
8 /usr/local/lib
9 然后 sudo ldconfig 生效 // sudo /sbin/ldconfig
```

```
1 ii libavcodec-ffmpeg56:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg library
  with de/encoders for audio/video codecs - runtime files
2 ii libavdevice-ffmpeg56:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg library
  for handling input and output devices - runtime files
3 ii libavfilter-ffmpeg5:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg library
  containing media filters - runtime files
4 ii libavformat-ffmpeg56:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg library
  with (de)muxers for multimedia containers - runtime files
5 ii libavresample-ffmpeg2:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg compat
  ibility library for resampling - runtime files
6 ii libavutil-ffmpeg54:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg library w
  ith functions for simplifying programming - runtime files
7 ii libpostproc-ffmpeg53:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg library
  for post processing - runtime files
8 ii libswresample-ffmpeg1:amd64 7:2.8.17-0ubuntu0.1 amd64 FFmpeg librar
  y for audio resampling, rematrixing etc. - runtime files
9 ii libswscale-ffmpeg3:amd64
```

FFmpeg

```
1 卸载自带的ffmpeg 一定要先卸载, 再安装ffmpeg, 否则会出现libavutil版本不匹配
2 dpkg -l|grep ffmpeg
3 sudo apt-get --purge remove libavformat-ffmpeg56:amd64
4
5
6 sudo apt-get install ppa-purge
```

```
7 sudo apt-get update -qq && sudo apt-get -y install autoconf automake \
8 build-essential cmake git-core libass-dev libfreetype6-dev \
9 libsdl2-dev libtool libva-dev libvdpau-dev libvorbis-dev \
10 libxcb1-dev libxcb-shm0-dev libxcb-xf86-dev pkg-config texinfo wget
11 zlib1g-dev
12 sudo apt-get install nasm
13 sudo apt-get install yasm
14 sudo apt-get install libx264-dev
15 sudo apt-get install libx265-dev libnuma-dev
16 sudo apt-get install libvpx-dev
17 sudo apt-get install libfdk-aac-dev
18 sudo apt-get install libmp3lame-dev
19 sudo apt-get install libopus-dev
20
21 卸载之前源码安装的ffmpeg
22 cd ~/software/ffmpeg-3.4.7/
23 sudo make uninstall
24
25 使用apt安装ffmpeg
26 sudo apt-get install software-properties-common
27 sudo apt-get install python-software-properties
28 sudo add-apt-repository ppa:jonathonf/ffmpeg-4
29 sudo apt-get update
30 sudo apt install ffmpeg -y
31 sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev lib
32 avutil-dev libavfilter-dev libswresample-dev
33
34 sudo apt remove ffmpeg -y
35 sudo apt-get remove libavcodec-dev libavformat-dev libswscale-dev liba
36 vutil-dev libavfilter-dev libswresample-dev
37
38 wget https://ffmpeg.org/releases/ffmpeg-4.2.4.tar.bz2
39 tar -xf ffmpeg-4.2.4.tar.bz2
40 cd ffmpeg-4.2.4
41 sudo ./configure --prefix=/usr --enable-shared --enable-pthreads --er
42 able-version3 --enable-hardcoded-tables --enable-swresample --host-
43 cflags= --host-ldflags= --enable-gpl --enable-ffplay --enable-libass --er
44 able-libfreetype --enable-libmp3lame --enable-libopus --enable-libvorbis
45 --enable-libvpx --enable-libx264
```

```
42
43 sudo make -j8 && sudo make install
44 ffmpeg -version
45 sudo apt-get update
```

libyuv

```
1 sudo apt-get install -y libjpeg-dev
2 cd ~/software/libyuv/
3 mkdir out && cd out
4 cmake -DCMAKE_INSTALL_PREFIX="/usr/lib" -DCMAKE_BUILD_TYPE="Release"
..
5 cmake --build . --config Release
6 sudo cmake --build . --target install --config Release
7
8 sudo vi /usr/lib/pkgconfig/libyuv.pc
```

把下面的内容拷贝进 libyuv.pc

```
1 prefix=/usr/lib/
2 exec_prefix=/usr/lib/
3 libdir=${exec_prefix}/lib
4 includedir=${prefix}/include/
5 Name: libyuv
6 Version: 1.1.4
7 Description: libyuv
8 Libs: -L${libdir} -lyuv
9 Cflags: -I${includedir}
```

nginx

```
1 sudo apt-get install nginx-full
2 参考附件修改 nginx 的配置，参考文末的样例
3 cd /etc/nginx/sites-available/
4 sudo cp default default.bak
5 sudo vi default
6 sudo service nginx restart
7 sudo service nginx
```

配置nginx

```
1 server {
2     root /home/ubuntu/janus/my/new_html;
3     index videoroom_vp3.html;
4
5     server_name us-live1.smart-leaping.com; # managed by Certbot
6
7     location / {
8         # First attempt to serve request as file, then
9         # as directory, then fall back to displaying a 404.
10        try_files $uri $uri/ =404;
11    }
12
13    location /janus {
14        proxy_pass http://127.0.0.1:8088/janus;
15    }
16
17    location /admin {
18        proxy_pass http://127.0.0.1:7088/admin;
19    }
20
21    # cebort证书
22    listen [::]:443 ssl ipv6only=on; # managed by Certbot
23    listen 443 ssl; # managed by Certbot
24    ssl_certificate /etc/letsencrypt/live/us-live1.smart-leaping.com/fullchain.pem; # managed by Certbot
25    ssl_certificate_key /etc/letsencrypt/live/us-live1.smart-leaping.com/privkey.pem; # managed by Certbot
26    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
27    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
28
29 }
```

certbot HTTPS证书

```
1 sudo apt install snapd
2 sudo snap install --classic certbot
3 sudo certbot --nginx
4
5 删除证书
```

```
6 sudo certbot delete
7
8 如果没有lsb
9 sudo apt-get install lsb-core
```

coturn

```
1 cd ~/software/libevent-2.1.8-stable/
2 ./autogen.sh
3 ./configure && make
4 sudo make install
5
6 安装 turn
7 cd ../coturn/
8 ./configure
9 make -j8
10 sudo make install
11 cd /usr/local/etc
12 sudo cp turnserver.conf.default turnserver.conf
13
14 //启动
15 su
16 中国船员在槟城附近漂流15个月，中领馆安排首批26日回国
17 唐人街
18 2小时前
19
20 龚正为第4期“中青班”作报告，寄语上海年轻干部do turnserver -o -c /usr/local/etc/turnserver.conf
```

配置coturn

```
1 sudo vim /usr/local/etc/turnserver.conf
2
3 listening-port=3478
4 external-ip=[服务器的ip]
5 min-port=50001
6 max-port=65535
7 user=demon:demon // user=turnuser1:pass023461
8 realm=stun.wangzhumo.com
```

用openssl生成证书和密钥

```
1 cd /home/ubuntu/cert
2 openssl genrsa > cert.key #生成私钥
3 openssl req -new -x509 -key cert.key > cert.pem #生成证书,需要输入信息
4 Country Name (2 letter code) [XX]: china #国家
5 State or Province Name (full name) []:hunan #省份
6 Locality Name (eg, city) [Default City]:changsha #城市
7 Organization Name (eg, company) [Default Company Ltd]:xxx #公司名
8 Organizational Unit Name (eg, section) []:xxx #单位名
9 Common Name (eg, your name or your server's hostname) []:主机名 #主机名hostname
10 Email Address []:xx@xx.com
```

配置nginx

```
1 server {
2     listen 80 default_server;
3     listen [::]:80 default_server;
4     listen *:443 ssl;
5
6     root /home/ubuntu/janus/my/new_html;
7     index videoroom_vp3.html;
8
9     location /janus {
10     proxy_pass http://127.0.0.1:8088/janus;
11 }
```

安装Python3.5

```
1 使用源码安装Python
2 wget https://www.python.org/ftp/python/3.5.0/Python-3.5.0.tgz
3 tar zxvf Python-3.5.0.tgz
4 ./configure --prefix=/usr/ --enable-shared CFLAGS=-lutil
5 make && make install
```



```

6 重要：需要开启shared的原因：
7 编译安装完成后，libpython3.5m 在/usr/lib中，
8 但是它需要调用/usr/lib/x86_64-linux-gnu/libutil.so
9
10 更换引用
11 cd /usr/bin/
12 mv /usr/bin/python /usr/bin/python_old
13 ln -s /usr/bin/python3 /usr/bin/python
14
15
16 重要： 复制lsb_release.py， 否则无法使用pip安装opencv_python
17 cp /usr/lib/python2.7/dist-packages/lsb_release.py
   /usr/lib/python3.5/site-packages

```

安装opencv_python

```

1 sudo apt-get install python3-pip
2
3 pip3 install --upgrade pip
4 sudo pip install -i https://pypi.tuna.tsinghua.edu.cn/simple opencv-
python
5
6 将#include<Python.h> 改为 #include<python3.5m/Python.h>
7 /usr/lib/python3.5/site-packages/numpy/core/include/numpy/ndarrayobjec
t.h:11:20
8 /usr/lib/python3.5/site-
packages/numpy/core/include/numpy/np_common.h:11:20
9 /usr/include/numpy/ndarrayobject.h:17
10 /usr/include/numpy/np_common.h:11
11

```

ubuntu16.4安装OpenCV 3.2

1 安装依赖

```

1 sudo apt-get update # 更新软件源
2 sudo apt-get install build-essential # 安装编译所需的库
3 sudo apt-get install libgtk2.0-dev libavcodec-dev libavformat-dev libsw
scale-dev python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-
dev libtiff-dev libtiff4-dev libswscale-dev libjasper-dev libdc1394-22-de
v libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev pkg-config

```

2 准备opencv-3.2.0.zip，解压后复制到OpenCV 3.2下

3 在opencv-3.2.0/modules/videoio/src/cap_ffmpeg_impl.hpp的头部添加，否则可能与ffmpeg版本不兼容

```
1 #define AV_CODEC_FLAG_GLOBAL_HEADER (1 << 22)
2 #define CODEC_FLAG_GLOBAL_HEADER AV_CODEC_FLAG_GLOBAL_HEADER
3 #define AVFMT_RAWPICTURE 0x0020
```

4 先编译一次

```
1 mkdir build && cd build
2
3 cmake -D CMAKE_BUILD_TYPE=Release -D CMAKE_INSTALL_PREFIX=/usr/local/j
  anus \
4   -D OPENCV_EXTRA_MODULES_PATH=../opencv_contrib-3.2.0/modules/ \
5   -DWITH_CUDA=OFF \
6   -DBUILD_DOCS=OFF \
7   -DBUILD_EXAMPLES=OFF \
8   -DBUILD_TESTS=OFF \
9   -DBUILD_PERF_TESTS=OFF .. \
10  --enable-shared
```

5 准备依赖文件

复制protobuf-cpp-3.1.0.tar.gz到以下目录

```
1 /root/software/opencv-3.2.0/opencv_contrib-
  3.2.0/modules/dnn/.download/bd5e3eed635a8d32e2b99658633815ef/v3.1.0
```

复制以下文件到 /root/software/opencv-3.2.0/opencv_contrib-3.2.0/modules/xfeatures2d/src

```
1 boostdesc_lbgm.i
2 boostdesc_bgm.i
3 boostdesc_bgm_bi.i
4 boostdesc_bgm_hd.i
5 boostdesc_binboost_064.i
6 boostdesc_binboost_128.i
7 boostdesc_binboost_256.i
```

```
8 vgg_generated_64.i
9 vgg_generated_48.i
10 vgg_generated_80.i
11 vgg_generated_120.i
```

将ippicv_linux_20151201.tgz复制到以下路径

```
1 /root/software/opencv-3.2.0/3rdparty/ippicv/downloads/linux-808b791a6e
ac9ed78d32a7666804320e/
```

然后重新make

或者在opencv/3rdparty/ippicv/ippicv.cmake中修改文件的路径为本地路径

```
1 47行 "https://raw.githubusercontent.com/opencv/opencv_3rdparty/${IPPICV
_COMMIT}/ippicv/"
2 "file:///root/software/opencv-4.3.0/ipvc"
```

安装其它依赖并修改文件

```
1 sudo apt-get install liblapacke-dev checkinstall
2 在/build文件夹中找到opencv_lapack.h文件, 把#include "LAPACKE_H_PATH-NOTFO
UND/lapacke.h"改为#include "lapacke.h"
```

6 安装

```
1 sudo make -j2 && sudo make install
```

```
1 cd ..
2 mkdir release && cd release
3 sudo cmake ..
4 sudo make && sudo make install
5
6 vim /etc/bash.bashrc
7 添加以下两行
8 export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig
9 export LD_LIBRARY_PATH=<opencvSourceDir>/release/lib:$LD_LIBRARY_PATH
10
```

```
11 source /etc/bash.bashrc
12
```

安装janus其它依赖

```
1 sudo apt-get install libconfig-dev
2
3 sudo aptitude install libmicrohttpd-dev libjansson-dev libnice-dev \
4 libssl-dev libsrtplib-dev libsofia-sip-ua-dev libglib2.0-dev \
5 libopus-dev libogg-dev libcurl4-openssl-dev pkg-config gengetopt \
6 libtool automake
7
8 包含libImlmf, opencv的imwrite
9 apt-get install libopenexr-dev
```

修改janus中的configure.ac

```
1 PKG_CHECK_MODULES([JANUS],
2 [
3 glib-2.0 >= $glib_version
4 gio-2.0 >= $glib_version
5 libconfig
6 nice
7 jansson >= $jansson_version
8 libssl >= $ssl_version
9 libcrypto
10 zlib
11 libavutil
12 libavcodec
13 libavformat
14 libswscale
15 libswresample
16 libavfilter
17 ogg
18 libyuv
19 opencv
20 python3
21 ])
```

编译janus

```
1 cd ~/janus
2 sh ./autogen.sh
3 export PKG_CONFIG_PATH=/usr/local/janus/lib/pkgconfig
4
5 //./configure --prefix=/home/ubuntu/janus/build --disable-rabbitmq --d
isable-docs --disable-mqtt --disable-rabbitmq --disable-mqtt --disable-ra
bbitmq-event-handler --disable-mqtt-event-handler --disable-plugin-audio
ridge --disable-plugin-echochat --disable-plugin-recordplay --disable-plu
gin-textroom --disable-plugin-videocall --disable-plugin-voicemail --disa
ble-plugin-streaming --disable-plugin-nosip --disable-plugin-sip --enable
-websockets
6
7 ./configure --prefix=/home/ubuntu/janus/build --disable-websockets &&
make -j8 && sudo make install
8
9 sudo turnserver -o -c /usr/local/etc/turnserver.conf
10 make clean
11
12 //重新生成配置文件
13 make configs
14
```

在janus中配置coturn服务

```
1 vi /home/ubuntu/janus/build/etc/janus/janus.jcfg
2
3 nat: {
4   turn_server = "127.0.0.1"
5   turn_port = 3478
6   turn_type = "udp"
7   turn_user = "demon"
8   turn_pwd = "demon"
9   ice_enforce_list = "eth0"
10
11   nat_1_1_mapping = "47.91.23.199"
12   ice_enforce_list = "eth0"
13 }
14
15 // user=turnuser1:pass023461
```

修改videoRoom的配置文件

```
1 vi ./config/janus.plugin.videoroom.jcfg
2
3 room-1234: {
4     description = "Demo Room"
5     secret = "adminpwd"
6     publishers = 6
7     bitrate = 128000
8     fir_freq = 10
9     audiocodec = "opus"
10    videocodec = "h264"
11    h264_profile = "42e01f"
12    record = "no"
13    merging = "yes"
14    #rec_dir = "/path/to/recordings-folder"
15 }
```

```
1 #ifndef change_bg_hpp
2 #define change_bg_hpp
3
4 #include <stdio.h>
5 #define __STDC_CONSTANT_MACROS
6
7
8 #ifdef __cplusplus
9 extern "C" {
10 #endif
11 #include <libavutil/avutil.h>
12 #include <libavutil/samplefmt.h>
13 #include <libavutil/timestamp.h>
14 #include <libavformat/avformat.h>
15 #include <libavutil/imgutils.h>
16
17
```

```
18 void changeBG(AVFrame *frame);
19
20
21 #ifdef __cplusplus
22 }
23 #endif
24 #endif
```

安装深度学习环境

安装1.8版本的Tensorflow

```
1 pip install tensorflow==1.8 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

安装CPU版本的Tensorflow

```
1 pip3 install --upgrade \
2 https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow_cpu-
1.1.0-cp35-cp35m-manylinux2010_x86_64.whl
```

```
1 - 请求/janus/streaming/5909858260155905参数为 {} 请先登陆
2 ERROR j.C.BaseExceptionHandler - [error,25] - 请求/janus/videoRoom/722
8599844628583参数为 {} 请先登陆
```

```
1 {
2   "janus": "message",
3   "body": {
4     "request": "exists",
5     "room": 5611450586256011,
6     "admin_key": "janusoverlord",
```

```
7  "secret": "videoroom_secret"
8  },
9  "transaction": "qBXefZfC0EaA",
10 "apisecret": "think_jump"
11 }
```

```
1 Failed in creating mixing publisher mp!
2
3 Can't add 'rtp' stream: no mcu_mixer...
```

```
1 var destroy_mixing_params = {
2     "request": "destroy_mcu_mixer",
3     "mcu_mixer": this_instance.mixing_mcu_mixer_id,
4     "secret": "mixing_secret"
5 };
6
7 public String janus;
8 public String transaction;
9 public String admin_secret;
10 public String plugin;
11 public Long user_id;
```

```
1 "request": "destroy_publisher_mp",
2     "mcu_mixer": this_instance.mixing_mcu_mixer_id,
3     "id": this_instance.mixing_publisher_stream_id,
4     "secret": "mixing_secret"
```

从私钥提取公钥

```
1 ssh-keygen -e -f key.pem >> key.pem.pub
2 ssh-keygen -e -f tang-cn1.pem >> tang-cn1.pem.pub
```


