

# 安装步骤

## 安装步骤

1. 配置epel仓库(coturn需要配置这个仓库)
2. 安装coturn和python3
3. 使用pip安装需要的包
4. 覆盖flask的部分源码, 满足网页字符编码GB18030的要求
5. 将webrtc的源码部分复制到/usr/bin/webrtc\_Tony
6. 将webrtc和turnserver的config文件复制到/etc
7. 将用于https和turn服务器的签名文件复制到/etc/webrtc\_Tony/
8. 建立错误日志目录
9. 将用于开机自动启动的脚本和服务复制到对应的位置, 设置开机自启

(当前目录默认是990101/)

## 1. 配置epel仓库(coturn需要配置这个仓库)

rpm文件是 990101/source/epel-release-8-15.el8.noarch.rpm

```
rpm -Uvh --force ./source/epel-release*rpm
```

```
[root@vm-2053402 ExtraHW]# rpm -Uvh epel-release*rpm
警告: epel-release-8-15.el8.noarch.rpm: 头V4 RSA/SHA256 Signature, 密钥 ID 2f86d6a1: NOKEY
Verifying... ##### [100%]
准备中... ##### [100%]
正在升级/安装...
 1:epel-release-8-15.el8 ##### [100%]
[root@vm-2053402 ExtraHW]#
```

## 2. 安装coturn和python3

```
dnf -y -q install python3 coturn
```

```
[root@vm-2053402 ExtraHW]# dnf install -y -q coturn python3
已安装:
coturn-4.5.2-1.el8.x86_64
libpq-13.5-1.el8.x86_64
perl-Compress-Raw-Zlib-2.081-1.el8.x86_64
perl-HTTP-Date-6.02-19.module+el8.6.0+965+850557f9.noarch
perl-I0-Compress-2.081-1.el8.noarch
perl-LWP-MediaTypes-6.02-15.module+el8.6.0+965+850557f9.noarch
python3-pip-9.0.3-22.el8.rocky.0.noarch
telnet-1:0.17-76.el8.x86_64
hiredis-0.13.3-13.el8.x86_64
perl-Compress-Raw-Bzip2-2.081-1.el8.x86_64
perl-Encode-Locale-1.05-10.module+el8.6.0+965+850557f9.noarch
perl-HTTP-Messsage-6.18-1.module+el8.6.0+965+850557f9.noarch
perl-I0-HTML-1.001-11.module+el8.6.0+965+850557f9.noarch
perl-TimeDate-1:2.30-15.module+el8.6.0+965+850557f9.noarch
python36-3.6.8-38.module+el8.5.0+671+195e4563.x86_64
```

## 3. 使用pip安装需要的包

```
pip3 install -q -r ./source/requirements.txt
```

```
[root@vm-2053402 source]# pip3 install -q -r requirements.txt
WARNING: Running pip as the 'root' user can result in broken permissions and c
virement instead: https://pip.pypa.io/warnings/venv
[root@vm-2053402 source]#
```

注: 此处如果pip版本过低, 可能会出现UnicodeError, 需要先将linux的字符集设置为UTF-8并且升级pip

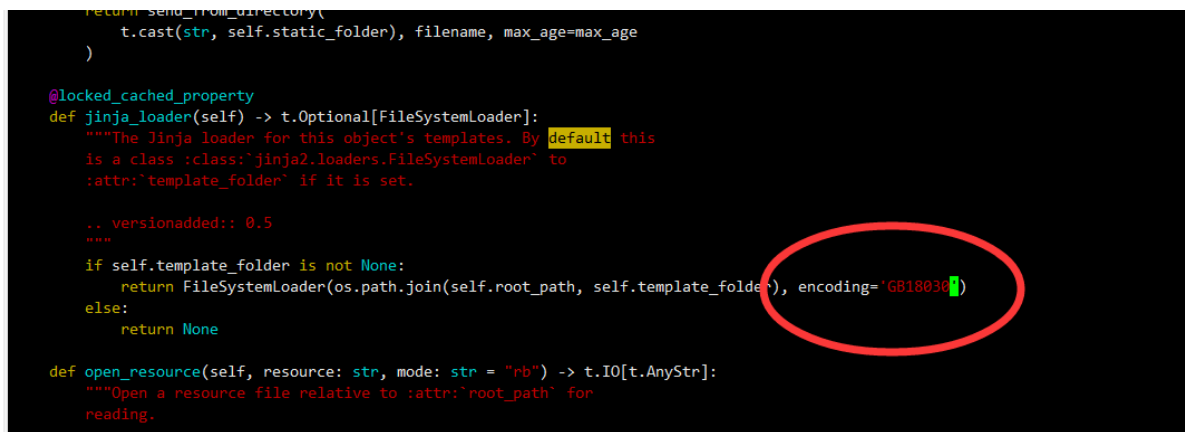
(现象是pip无法执行任何安装/升级的操作, 在GB18030编码下使用 `python3 -m pip install --upgrade pip` 也是UnicodeError)

## 4. 覆盖flask的部分源码, 满足网页字符编码GB18030的要求

修改好的scaffold.py已经放在 `990101/source/scaffold.py`, 执行的操作是

```
\cp ./source/scaffold.py $(pip3 show flask | grep Location | awk -F "Location: " '{printf "%s/flask/", $2}')
```

要覆盖的文件是flask里的scaffold.py, 如下图, 需要在345行左右的位置添加`encoding="GB18030"`, 否则使用默认编码为UTF-8



```
return send_from_directory(
    t.cast(str, self.static_folder), filename, max_age=max_age
)

@locked_cached_property
def jinja_loader(self) -> t.Optional[FileSystemLoader]:
    """The Jinja loader for this object's templates. By default this
    is a class :class:`jinja2.loaders.FileSystemLoader` to
    :attr:`template_folder` if it is set.

    .. versionadded:: 0.5
    """
    if self.template_folder is not None:
        return FileSystemLoader(os.path.join(self.root_path, self.template_folder), encoding='GB18030')
    else:
        return None

def open_resource(self, resource: str, mode: str = "rb") -> t.IO[t.AnyStr]:
    """Open a resource file relative to :attr:`root_path` for
    reading.
```

## 5. 将webrtc的源码部分复制到/usr/bin/webrtc\_Tony

源码部分的位置在 `990101/source/code`

```
mkdir /usr/bin/webrtc_Tony/
\cp -r ./source/code /usr/bin/webrtc_Tony/
```

## 6. 将webrtc和turnserver的config文件复制到/etc

两个config文件都在 `99011/config` 目录下, 都需要复制到/etc/中

```
\cp -r ./config/* /etc
```

## 7. 将用于https和turn服务器的签名文件复制到/etc/webrtc\_Tony/

需要pem格式的签名文件公钥和私钥

提供了默认的签名文件 `990101/source/cert.crt` 和 `990101/source/cert.key`

```
mkdir /etc/webrtc_Tony/
\cp ./source/cert.crt ./source/cert.key /etc/webrtc_Tony
```

## 8. 建立错误日志目录

```
mkdir /var/log/webrtc_Tony/
```

## 9. 将用于开机自动启动的脚本和服务复制到对应的位置, 设置开机自启

```
chmod +x ./source/flask_run.sh
\cp ./source/flask_run.sh /usr/bin/webrtc_Tony/
\cp ./source/flask.service /etc/systemd/system/
systemctl daemon-reload
systemctl start flask.service
systemctl enable flask.service
```

所有需要的文件都已经复制到对应位置

数据库进行了初始化, 数据库的导入在使用手册中介绍

如果需要导入样例数据库信息, 执行 `mysql -u root -proot123 < /etc/webrtc_Tony/user.sql` 即可