How to Install Coturn

Long Term Credentials Mechanism vs Time Limited Credentials (Short Term Credentials) Mechanism

MeetrixIO team is well experienced with WebRTC realated technologies. We provide **commercial support** for Jitsi Meet, Kurento, OpenVidu, BigBlue Button, Coturn Server and other webRTC related opensource projects.

Coturn (https://github.com/coturn/coturn) is an opensource turn server. This guide has been tested on Ubuntu 18.04.

Firewall Rules

First Make sure that you have opened up following ports in your firewall. You can always change the ports you want to use for the setup.

```
80 : TCP # if you need to setup coturn with SSL
443 : TCP # if you need to setup coturn with SSL
3478 : UDP
10000-20000 : UDP
```

Installing Coturn

Login to Ubuntu shell and enter following command to install Coturn

```
sudo apt-get -y update
sudo apt-get -y install coturn
```

Start the Coturn Daemon at Startup

To setup coturn start at system startup

```
sudo vim /etc/default/coturn
```

Uncomment the following line by removing the `#` at the beginning to run Coturn as an automatic system service daemon

```
TURNSERVER_ENABLED=1
```

Configuration Mechanisms

For any application or to user use the turn server, they need to have a username and a password. Depending on how the username and passoword are created there are two main methods that we can configure the turn server.

Long Term Credentials Mechanism

The Long Term Credentials Mechanism (https://tools.ietf.org/html/rfc5389#section-10.2) is simple. A pair of username and password is shared between the Turn Server and the application/user. This credentials will not expire. Anyone who has these credentials can use the turn server. This mechanism is suitable for applications where the turn server credentials are not exposed to end users but used only by the servers.

Time-Limited Credentials Mechanism

In Time-Limited Credentials Mechanism (described here (https://tools.ietf.org/html/draft-uberti-behave-turn-rest-00#section-4)), a static key is shared between the turn server and the appliaction. This shared secred then will be used to generate dynamic usernames and passwords by the application which can be . These dynamic usernames and passwords then can be used by the applications and they will expire within a predefined time period. This mechanism is more suitable for the applications where the turn server credentials should be exposed to the end users. Jitsi, SimpleWebRTC, SpreedWebRTC supports Time-limited Credentials Mechanism.

More information can be found in Coturn Wiki (https://github.com/coturn/coturn/wiki/turnserver).

Configuring Coturn

With Long Term Credential Mechanism

This method should work with most of the versions of Coturn. Open (or create) `/etc/turnserver.conf` file and past the following content. Replace `<YOUR_USERNAME>` , `<YOUR_PASSWORD>` , `<INTERNAL_IP>` and `<YOUR_PUBLIC_IP_ADDRESS>` values with your own ones.

```
realm=coturn.meetrix.io
fingerprint
listening-ip=0.0.0.0
external-ip=<EXTERNAL_IP>/<INTERNAL_IP> #or just the external ip
listening-port=3478
min-port=10000
max-port=20000
log-file=/var/log/turnserver.log
verbose

user=<YOUR_USERNAME>:<YOUR_PASSWORD>
lt-cred-mech
```

Now restart the coturn service

```
sudo service coturn restart
```

With Time-Limited Credentials Mechanism

When a turn server is installed, we can start the turn server with Time-limited Credentials Mechanism using `static-auth-secret` flag and we can pass the shared secret.

```
realm=coturn.meetrix.io
fingerprint
listening-ip=0.0.0.0
external-ip=<EXTERNAL_IP>/<INTERNAL_IP> #or just the external ip
listening-port=3478
min-port=10000
max-port=20000
log-file=/var/log/turnserver.log
verbose

static-auth-secret=<YOUR_SECRET>
```

Now restart the coturn service

```
sudo service coturn restart
```

Pro TIP : Setting up Coturn with SSL

Some firewalls do not allow traffic from ports other than 80 or 443. And some rules might enforce TLS or SSL security over the transport. To support these usecases we can run turn server on port 443 with letsencrypt certificates.

For this you need a domain which is pointed to the server that you are going to install the turn srever

eg: `coturn.meetrix.io` .

And your port 80 and 443 should be open to the public (both inbound and outbount).

First you have to install `Certbot` certificate client from certbot.eff.org (https://certbot.eff.org). On an `Ubuntu 18.04` box, copy and past following commands. Otherwise, follow the official guide.

```
sudo apt-get -y  update &&\
sudo apt-get -y install software-properties-common &&\
sudo add-apt-repository -y universe &&\
sudo add-apt-repository -y ppa:certbot/certbot &&\
sudo apt-get -y update &&\

sudo apt-get -y install certbot
```

Then you have to request the certificates using certbot.

```
sudo certbot certonly --standalone
```

You will be prompted to provide your domain name. If everything goes well, your certificates will be installed in `/etc/letsencrypt/live/<YOUR_COTURN_DOMAIN>` directory. Once you are done with installing the certificates, you have to add following additional lines to the config.

```
server-name=<YOUR_COTURN_DOMAIN>
cert=/etc/letsencrypt/live/<YOUR_COTURN_DOMAIN>/cert.pem
pkey=/etc/letsencrypt/live/<YOUR_COTURN_DOMAIN>/privkey.pem
```

Then you can change the port to `443` as well.

For example, config for the long term crendetials mechanism will look like this.

```
server-name=coturn.meetrix.io
cert=/etc/letsencrypt/live/coturn.meetrix.io/cert.pem
pkey=/etc/letsencrypt/live/coturn.meetrix.io/privkey.pem
realm=coturn.meetrix.io
fingerprint
listening-ip=0.0.0.0
external-ip=<EXTERNAL_IP>/<INTERNAL_IP> #or just the external ip
listening-port=443
min-port=10000
max-port=20000
log-file=/var/log/turnserver.log
verbose

user=<YOUR_USERNAME>:<YOUR_PASSWORD>
lt-cred-mech
```

Testing

For testing we can use Trickle-Ice testing tool. Go to trickle-ice (https://webrtc.github.io/samples/src/content/peerconnection/trickle-ice/) page and enter following details.

```
STUN or TURN URI : turn:<YOUR_PUBLIC_IP_ADDRESS>:3478
TURN username: <YOUR_USERNAME>
TURN password: <YOUR_PASSWORD>
```

If you have configured the turn server in Long-Term Credentials Mechanism, you can directly use the credentials.

But if you are using Time-Limited Credentials Mechanism, you can use following script to generate a username and a password

```
secret=mysecret && \
time=$(date +%s) && \
expiry=8400 && \
username=$(( $time + $expiry )) &&\
echo username:$username && \
echo password : $(echo -n $username | openssl dgst -binary -sha1 -hmac $secret | openssl base64)
```

output of this script would be some thing like following

```
username:1525325424
password : YuzkH/Th9BBaRj4ivR03PiCfr+E=
```

Then click `Add Server` and then `Gather candidates` button. If you have done everything correctly, you should see `Done` as the final result. If you do not get any response or if you see any error messages, please double check if you have followed this guide as it is.

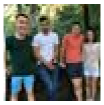
That’s it !

Looking for commercial support ? Please contact us via hello@meetrix.io

Updated: July 14, 2019

LEAVE A COMMENT

0 Comments Sort by **Oldest**



Add a comment...

Facebook Comments Plugin