Asterisk with Twilio Elastic SIP Trunking Configuration Guide

This guide shows one way of configuring Asterisk to work with Twilio's Elastic SIP Trunking product. Asterisk is a hugely flexible piece of software and it is quite possible to depart, significantly, from the instructions provided here and still bring up a fully working trunk. I've tried to highlight the essential concepts that you will need to follow.

If you have any feedback on this or any of our other configuration guides, please do email us at sip.interconnectionguides@twilio.com

Contents:

Installing Asterisk

The first thing to do is to set up a server and install Asterisk. I installed Asterisk on an Amazon EC2 instance running Redhat. Wherever you build your Asterisk, remember to configure firewall or security settings so that you can receive SIP and RTP from Twilio

and any other SIP devices (like phones) you will be using. Please see https://www.twilio.com/docs/api/sip-trunking/getting-started#whitelist for all the IP addresses you will need to whitelist.

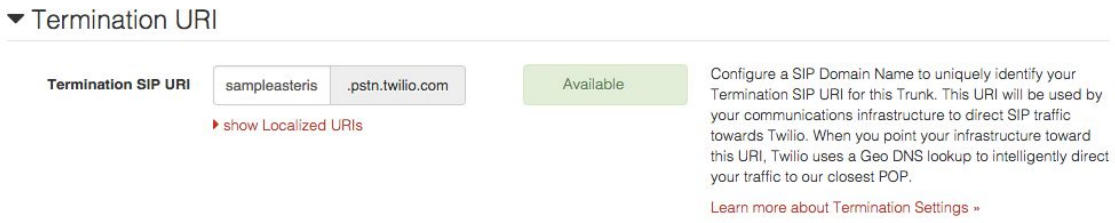
NB: If you know that you want to use TLS and SRTP on this trunk, please read Using Secure Trunking, below.

There's a pretty good guide for how to install Asterisk at http://www.ipcomms.net/sample-device-configurations/41-asterisk/179-asterisk-13-on-centos. We found that we had to install two extra packages before we could install asterisk: ncurses-devel and libuuid-devel.

Setting up your Twilio Elastic SIP Trunk

We have a pretty comprehensive guide on how to configure an Elastic SIP Trunk through your Twilio account portal at https://www.twilio.com/docs/sip-trunking/getting-started. I will highlight 5 aspects here.

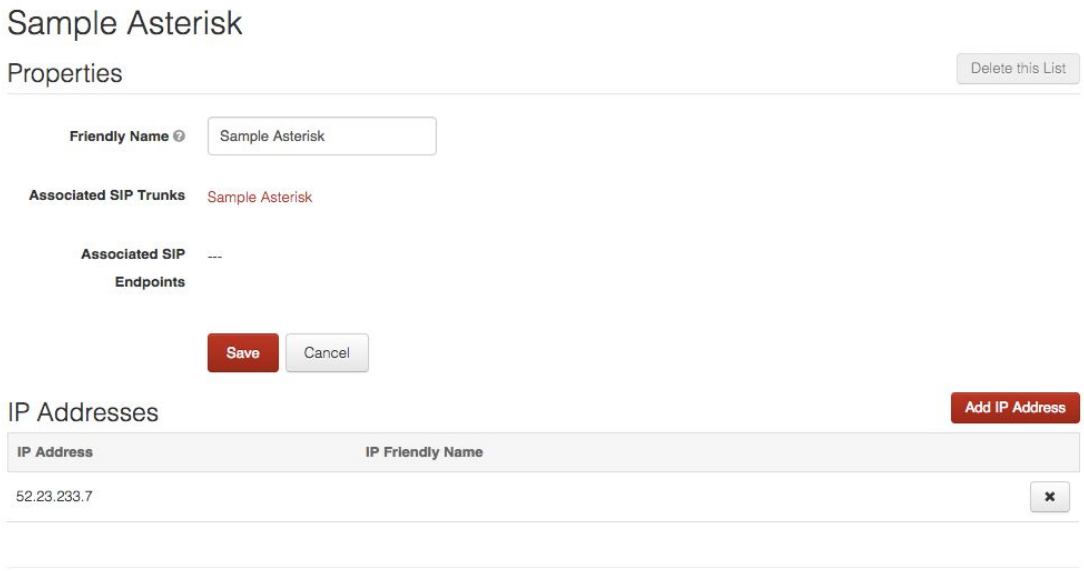
Termination URI



This is where you configure a unique URI that identifies your trunk. You will need to remember this when configuring your Asterisk because we need it to reference this URI in its SIP requests.

The termination URI I've chosen here is sampleasterisk.pstn.twilio.com. It's too long to fit neatly into the text box in the screenshot.

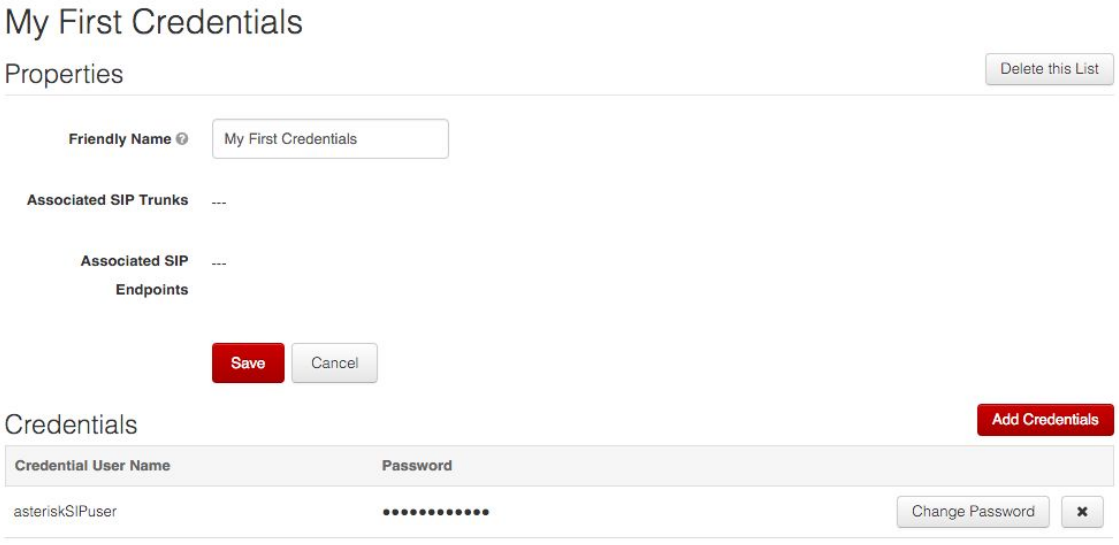
ACL



You can add an ACL that contains the public IP address of your Asterisk. This tells Twilio to ignore SIP from anywhere except for your PBX, reducing the chance of anyone making fraudulent calls on your trunk.

You will need to add either an ACL or SIP credentials to your trunk.

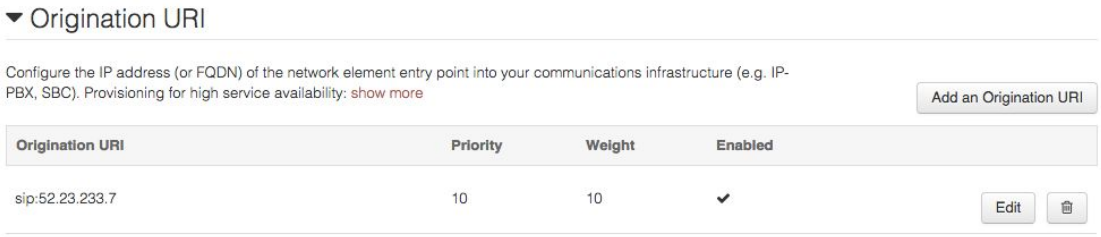
Credentials



You can add credentials to your trunk. If you do, we will expect any server that sends SIP requests to Twilio on this trunk to authenticate using these credentials. Again, this is a way of reducing the chance of someone making fraudulent calls on your trunk.

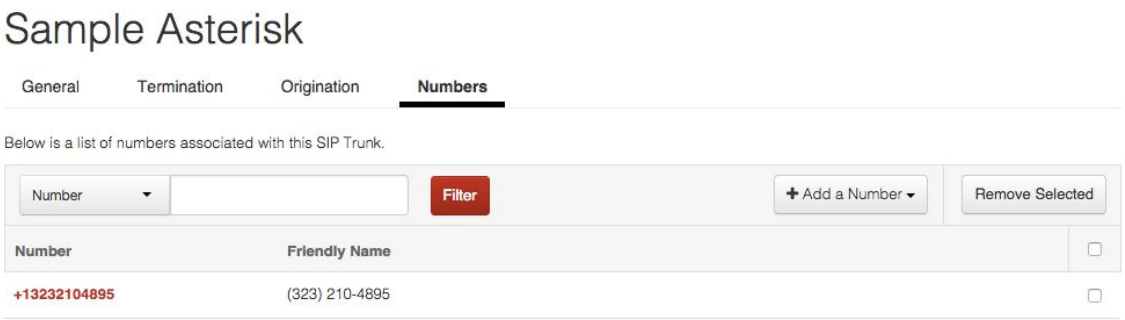
If you do not add credentials to your trunk, you will need to add an ACL.

Origination URI



The easiest way to configure the Origination URI is using “sip:” followed by the public IP address of your Asterisk. In my case, that would be “sip:52.23.233.7”. Although I have used an IP address here, you can also use an FQDN.

Phone Numbers



If you want to receive calls from the PSTN, across your new trunk, you will need to add a phone number to it. If someone calls this number, Twilio will contact your PBX using the Origination URI, above.

Configuring your Asterisk

In this section we’re going to walk through a minimal configuration to use your Asterisk system for inbound and outbound calls over a Twilio SIP Trunk.

sip.conf

If you are using the default SIP driver, chan_sip, you will need to edit sip.conf. First, set up some global parameters. Because I was building on an AWS EC2 instance, my asterisk server was behind NAT so I needed to account for this:

```
[general]
udpenable=yes
tcpenable=yes
preferred_codec_only=yes
disallow=all
allow=ulaw ;Twilio does G.711 only
sipdebug=yes
localnet=172.31.0.0/16   ;My EC2 instance had a private IP in
this range
externaddr=52.23.233.7  ;The public IP of my EC2 instance
```

Then, we need to build a trunk to Twilio.

```
[twilio-trunk](!)
type=peer
context=from-twilio  ;Which dialplan to use for incoming calls
dtmfmode=rfc4733
canreinivite=no
insecure=port,invite

[twilio0](twilio-trunk)
host=sampleasterisk.pstn.twilio.com  ;Our Termination URI
remotesecret=nottelling       ;password,  if  you  are  using
credentials
defaultuser=myuser               ;username,  if  you  are  using
credentials


[twilio1](twilio-trunk)
host=54.172.60.2

[twilio2](twilio-trunk)
host=54.172.60.0

[twilio3](twilio-trunk)
host=54.172.60.1

[twilio4](twilio-trunk)
host=54.172.60.3
```

In the US, Twilio has 4 different IP addresses that it uses for SIP signaling. We need the Asterisk server to recognize a SIP INVITE coming from any of these addresses as coming over this trunk. That means we need to build a trunk for each address. We then need to build another trunk for outgoing calls to Twilio, one that uses the Termination URI we configured earlier. When you are building multiple objects with similar configuration objects on Asterisk, using templates makes life easier. A template is denoted by putting a (!) at the end of the object name. We then use that template by putting its name in () at the end of a new object's name.

If you want to use non-US numbers on your trunk, you will need to authorize other Twilio Regional IP addresses. The full list can be found at https://www.twilio.com/docs/api/sip-trunking/getting-started#whitelist

Then, build objects for the SIP phones you will register with the Asterisk.

```
[office-phone](!)
type=friend
context=from-phones
host=dynamic
secret=Asterisk15
dtmfmode=auto
disallow=all
allow=ulaw

[1001](office-phone)
```

pjsip.conf

If you have installed, and are using pjsip, instead of chan_sip, you will need to edit pjsip.conf.

First, we need to build a transport. Again, I had to account for the fact that my EC2 instance is behind NAT.

```
[transport-udp-nat]
type=transport
protocol=udp
bind=0.0.0.0
local_net=172.31.0.0/16
external_media_address=52.23.233.7
external_signaling_address=52.23.233.7
```

Then, we set up our trunk. Again, I will use templates to make life easier.

```
[twilio-trunks](!)
type=endpoint
transport=transport-udp-nat
```

```
context=from-twilio
disallow=all
allow=ulaw

[auth-out](!)
type=auth
auth_type=userpass

[twilio0](twilio-trunks)
aors=twilio0-aors
outbound_auth=twilio0-auth                    ;if we are using
credentials

[twilio0-aors]
type=aor
contact=sip:sampleasterisk.pstn.twilio.com:5060

[twilio0-ident]
type=identify
endpoint=twilio0
match=54.172.60.0
match=54.172.60.1
match=54.172.60.2
match=54.172.60.3

[twilio0-auth](auth-out)
password=nottelling                           ;password on our
credentials
username=myuser                               ;username on our
credentials
```

And then we build objects for our SIP phones.

```
[endpoint-basic](!)
type=endpoint
transport=transport-udp-nat
context=from-phones
disallow=all
allow=ulaw

[auth-userpass](!)
type=auth
auth_type=userpass

[aor-single-reg](!)
type=aor
max_contacts=1
```

```
[1001](endpoint-basic)
auth=auth1001
aors=1001

[auth1001](auth-userpass)
password=ImNotTellingYou
username=1001

[1001](aor-single-reg)
```

extensions.conf

This is where we tell the Asterisk how to handle incoming calls. The examples here are for very, very basic North American dialing.

We'll start with calls coming in from the Twilio SIP trunk. We're just going to send all incoming calls to the one SIP phone registered as 1001. This is enough for testing but you will want to end up with something more complicated that this, probably. This example assumes you are using chan_sip. If you are using PJSIP, replace the "SIP" with "PJSIP"

```
[from-twilio]
exten => _+1NXXXXXXXXX,1,Dial(SIP/1001)
```

Then, for calls from our SIP phones, we'll just worry about ones that are to go out over the trunk. If you are using chan_sip:

```
[from-phones]
exten => _NXXNXXXXXX,1,Set(CALLERID(all)="David" <7845551234>)
same => n,Dial(SIP/twilio0/+1${EXTEN})
```

If you are using pjsip, the second line should be:

```
same => n,Dial(PJSIP/+1${EXTEN}@twilio0)
```

Using Secure Trunking

Twilio account portal

In the General Settings section for your trunk, you will need to enable secure trunking.

chan_sip

Before you can use secure trunking with chan_sip, you need to install srtp. Before you can install srtp, you need to install words!

```
$cd /usr/src
$sudo yum install -y words
```

The SRTP module uses words for self testing

```
$wget
https://downloads.sourceforge.net/project/srtp/srtp/1.4.4/srtp-
1.4.4.tgz
$tar zxvf srtp-1.4.4.tgz
$cd /usr/src/srtp
$./configure CFLAGS=-fPIC --prefix=/usr/local/lib
$make
```

here, you're going to have to fix up one of the install scripts.

```
cd test
vi rtpw_test.sh
```

change line 7 from "`RTPW=rtpw`" to "`RTPW=./rtpw`"

save and exit (`<esc>` followed by `:wq`)

```
cd ..
$make runtest
$make install
```

Now, when you configure Asterisk, replace the "`./configure`" command with "`./configure --with-srtp=/usr/local/lib`" This tells asterisk to include the SRTP module and where to find the linkable libraries (note that we told the SRTP module to use /usr/local/lib when configuring it, above).

Then, you can follow the instructions in https://wiki.asterisk.org/wiki/display/AST/Secure+Calling+Tutorial to enable TLS and SRTP on your asterisk.

Using TLS with Twilio requires a couple more settings, too. On the transport object for TLS, you should set

```
tlsdontverifyserver=yes
```

On the peer object for the SIP trunk, you should set

```
encryption=yes
media_encryption=sdes
```

PJSIP Channel Driver

Asterisk ships by default with `chan_sip` driver and works well with Twilio. However, if you have some reason to run `pjsip` driver with Asterisk, please note the following:

- Asterisk 13.8 cert2 defaults to `PJSIP 2.5` and it does not work with Twilio for TLS/SRTP purposes. Non-encrypted calls do work.
- Asterisk 13.8 cert2 can also use the latest `PJSIP` driver, which at this time is 2.5.5. Twilio works well with it despite the following message appearing in your log:

```
Sep 27 13:03:56] ERROR[10886]: pjproject:0 <?>:
tlsc0x7f217c03 RFC 5922 (section 7.2) does not allow TLS
wildcard certificates. Advise your SIP provider, please!
```

The following link is a guide to installing a non-bundled version of `PJSIP`. Change the version to 2.5.5 in the steps.
[Installing PJSIP channel driver]