DIY Telephone Line

a beginner's guide to
running your own VoIP server

This was first written in June 2025 so things may be a bit out of date!

If you'd like to reach out get in touch:
@harriethw on Mastodon
machinestreams@protonmail.com email

What is this?

This is a quick beginner's guide to get up and running with self-hosting your own telephone server - using (mostly) free, open source, software.

You can find the digital version at

https://github.com/Harriethw/home-phone/blob/main/notes/how_to.md

What you could build

- Your own voicemail app

- A telephone exchange for a small business or organisation

- Art projects that incorporate sound and/or branching narratives

- A telephone line for news or announcements that you want to share with your community

And, once you get more advanced

- Sending SMS for free across the world (https://github.com/MatejKovacic/RasPBX-install)

- Setting up free payphones in your town (https://philtel.org/about/)

Why??

Although there are many third-party tools and products to help you do this, if you are the type of person that likes making (and breaking!) things, or learning more about how something is made, or maybe you have a very specific vision for a project and a tight budget, then it's worth spending some time learning about self hosting!

More philosophically, we count every act of creation, education, modification or intervention that moves us away from the confines defined by Big Tech as a Big Win for humanity.

Things you need

Knowledge

- These instructions should hopefully be beginner-friendly, but if you have a very basic grasp of these it will be a lot easier:

 - a little bit of an understanding of [command line interfaces](https://www.freecodecamp.org/news/command-line-for-beginners/)

 - a basic mental model of what a server is and how they computers talk to each other

I am a complete novice so if I can do it you can too!

Time

- Approx 1-3 hours in total, including down time while you wait for stuff to download

Hardware

1. Broadband and an ethernet cable to plug into your modem - it will work over WiFi but service may drop out randomly as WiFi doesn't provide a consistent signal

2. A Raspberry Pi. I'm using a Raspberry Pi 3B for this

3. A Raspberry Pi power cable and plug - I'd recommend going for the official one as I've tried with other cables before and the power source wasn't consistent enough

4. A micro SD card that fits the Pi. Needs to be at least 8GB - I'm using a 32GB Amazon basics one.

5. A laptop to configure your server and download the OS

6. An SD card slot or adapter to put your SD card into your laptop

Software

1. Balena Etcher - (https://etcher.balena.io/)

2. If you don't already have your own telephone line, an account with a telephone provider - this will cost around £6 a month, I use voipfone - https://www.voipfone.co.uk/support/guides/pbx-services/third-party-pbx/freepbx

3. Terminal or other preferred app to `ssh` into your server.

Build your server

1. Download the OS to your SD Card

There are lots of mini-steps to this, so please look up on the computer these instructions and follow step by step (it can take between 10 mins - 1 hour)

https://github.com/Harriethw/freepbx-raspberrypi

^ These are instructions I copied over from a forum after hitting a wall trying to get FreePBX running following [the RasPBX project](http://www.raspbx.org/) and this awesome guide to [setting it up with a 3G Dongle](https://github.com/MatejKovacic/RasPBX-install)

2. Sign into your server

Once you've followed those instructions you should have your `root` and `pi` users setup, and you should be able to login in two ways, using the username and passwords you setup for them.

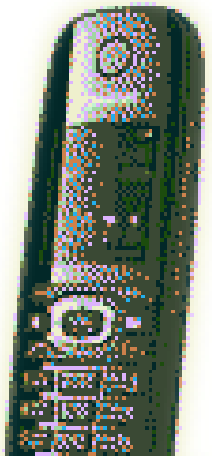
- **Command Line Interface (CLI)**

Via the terminal and an ssh command, replace with your own username (either `root` or `pi` if using default usernames) and IP address:

```
`ssh {user}@{pi.ip.address}`
```

- **Through the UI / Browser**

Via the browser by navigating to the pi's IP address



Good practices

- keep saving and applying config as you go in the UI.
- if you make changes via the CLI, restart the asterisk server `asterisk -rx "core restart now"`
- run the software updates regularly, using the CLI commands `apt update` and `apt upgrade`

Setting up email forwarding

I wanted emails from the server to be sent to my gmail address, which is super easy to do. The piece of software used for emails is called `Exim4`.

1. Setup Exim4 to use Gmail as an external SMTP server, by running this in terminal:

```
dpkg-reconfigure exim4-config
```

2. Choose the answers:

- Choose “mail sent by smarthost; received via SMTP or fetchmail”
- Set to “localhost” for “System mail name:”.
- Set to “127.0.0.1” for “IP-addresses to listen on for incoming SMTP connections” to
refuse external connections.
- Leave as empty for “Other destinations for which mail is accepted:”.
- Leave as empty for “Machines to relay mail for:”.
- Set to “smtp.gmail.com::587” for “IP address or host name of the outgoing smarthost:”.
- Choose “NO” for “Hide local mail name in outgoing mail?”.
- Choose “NO” for “Keep number of DNS-queries minimal (Dial-on-Demand)?”.
- Choose “mbox format in /var/mail/” for “Delivery method for local mail”.
- Choose “YES” for “Split configuration into small files?”.
- Enable 2-factor authentication on your Gmail account used for sending emails and follow Google instruction to create app specific password.
- Edit /etc/exim4/passwd.client and add the following line inside:
`smtp.gmail.com:<gmail-sender-account-name>:<your-app-password>`

You should use account name not email address here.

3. If it is new or restored file then ensure the permissions are correct by running this in the terminal:

```
chown root:Debian-exim /etc/exim4/passwd.client
chmod 640 /etc/exim4/passwd.client
```

4. You can also configure replacement of the local sender address to your Gmail one:

```
echo '<local-user-name>: <gmail-sender-account-name>@gmail.com' >> /etc/email-addresses
echo '<local-user-name>@localhost: <gmail-sender-account-name>@gmail.com' >> /etc/email-addresses
echo '<local-user-name>@<hostname>: <gmail-sender-account-name>@gmail.com' >> /etc/email-addresses
```

5. Update Exim4 configuration, remove stale mail:

```
update-exim4.conf
invoke-rc.d exim4 restart
exim4 -qff
```

6. Check that email is sent using:

```
echo "Subj" | mail -s "Test email from Exim4" <email-address>
```

7. Check logs to see if email was sent successfully:

```
less /var/log/exim4/mainlog
```

8. Check if email is received by <email-address>. If it is Gmail address you may need add filtering rule to prevent email be classified as a spam.

9. In order to redirect the local mail to the external address, edit the `/etc/aliases` file and add the following line:

```
<local-user-name>: <email-address>
```

and execute:

```
newaliases
```

in order to update aliases configuration.

Links:

- [Debian Wiki Exim4 for Gmail](https://wiki.debian.org/Exim4Gmail)

- [Gmail mail client setup](https://support.google.com/mail/answer/7104828?hl=en&visit_id=638300159533347627-1330126653&rd=3)

- [Debian Wiki Exim4](https://wiki.debian.org/Exim)

- [How to redirect local root mail](http://blog.bobbyallen.me/2013/02/03/how-to-redirect-local-root-mail-to-an-external-email-address-on-linux/)

Directing calls to your new server

Getting your phone number

If you haven't already, get yourself a phone number from a VoIP provider. These are instructions using the UI that work with Voipfone.co.uk but should hopefully work across providers

Once signed in as an admin in the UI:

Chan pjsip

1. Navigate to Trunks under the Connectivity tab

1. Click 'Add Trunk' and select 'Add SIP (chan_pjsip) Trunk'

1. Under the 'General' tab, set the trunk name and outbound CallerID as your voipfone SIP Username, under your Master Account > Phone Settings.

1. Under the pjsip Settings add the following:
   - Username / Auth username: voipfone SIP Username
   - Secret: the 'phone password' from voipfone, under the 'Phone settings'
   - SIP Server: sip.voipfone.net

1. Under the advances tab add the following:
   - Contact User/From User: voipfone SIP Username
1. Submit the form and click Apply Config

Outbound Route

1. Navigate to Connectivity > Outbound route
1. Add an Outbound Route. Set the name to something descriptive like `out-by-voipfone`
1. Under Dial Patterns tab, set the match field to `X.` to match any pattern
1. Submit the form and click Apply Config

Inbound Route

1. Give it another name (eg: 'in-from-voipfone')
1. DID Number: Your voipfone SIP Username
1. The Destination option will point the caller to any feature you want (e.g announcement/IVR) you wish. Can be a standard Annoucnement for now if you haven't yet set anything up.
1. Submit the form and click Apply Config

Once this is done, you should be able to call your telephone number and be directed to whatever Inbound destination you chose.

Customise your FreePBX

Adding your own audio

UI

Under the 'Admin' tab, the 'System Recordings' feature is where all of your custom recordings will live.
Before you setup any announcements or other apps, record your audio and load them here.

- I've found `.wav` files easiest to work with
- Save time by saving them as MONO,16bits at 8KHz, on your computer before uploading

CLI

System recordings are saved at the path `/var/lib/asterisk/sounds/en/custom/` so if you ever need to copy them or move them after adding you can.

If you think you've saved the audio in correct format already, you could copy them from your local computer to that folder via the terminal. But I'm not always confident I've saved the audio correctly so I prefer to go via UI.

Setting up voicemail

Voicemail inboxes belong to a User, who has an Extension.

1. Go to Connectivity > Extenstions > Add Extension > Add Virtual Extension
2. Under 'User Extension' put a random single number, I just use 1 [1]
3. Link to a default user - select 'Create new user' from the drop down
4. Check 'custom username' if you want to set the username.
5. Under 'Voicemail' tab:
 - click yes to 'Enabled'
 - add your email address to send voicemails
  - change any other settings you'd like to configure
 - You probably don't need VmX Locator
6. Save and Apply Config

You can manage that User later under Admin > User Management.

You can now redirect IVRs, Announcements or calls to your voicemail. For example, try setting your Inbound Route from the last step to point to your user's voicemail as its Destination.

CLI

Voicemail announcement recordings are stored in `/var/spool/asterisk/voicemail/default/{user_id}` and called e.g. `unavail.wav`, `busy.wav`. So you can swap out those files with your own recordings via the terminal if you wish.

[1] Note that this can effect your options when setting up an IVR - e.g. if your extension is `1` and an IVR option asks to press `1` it will auto redirect to that extension, but you can change that in the IVR settings.

Setting up interactive voice responses (IVR)

IVR's can allow the caller to interact with the server by pressing buttons - e.g. "Press 3 to go to the menu".

If you are getting pretty advanced with Asterisk then you can edit the code directly and start adding your own config here `/etc/asterisk/extensions_custom.conf` - but the easiest way is via the FreePBX UI.

Go to Applications > IVR > Add IVR

This gives you a nice menu to programme the behaviour of your IVR. You could have one layer of an IVR that allows users to press a button and play a recording, or you could chain IVRs so that one button leads to another menu of options.

A few key things:

- Announcement - this is the audio played at the beginning of the IVR

- Enable Direct Dial - You will want this to be set to "No" if you don't want users to be able to call a voicemail inbox

- Timeout - there's a few different options for how you handle timeout, e.g. if a caller hasn't pressed any buttons

- Invalid - same for if they press an invalid button

- Returning to the IVR - Once all options have been played out, or a timeout or invalid entry has occurred you can return the user to the beginning of the IVR

- IVR entries and Destinations - I found it easiest to program the IVR destinations as Announcements, rather than directly play system recordings. For some reason the behaviour wasn't super clear when just playing recordings. You can setup Announcements via the Application menu as well, and programme their destinations and behaviour individually.

Example IVR in code

Creating an extension that loops through a set of recordings and plays them

This is quite a specific use case but I thought it might be useful as an example of custom code that can be written.
I wanted to provide a service where users could leave a voicemail, and then press a button to listen to random voicemails from other people.
I put the following in the extensions_custom.conf file

```
;--------------------------------------------------------
; play-voicemail-custom:
;
; Custom extension to play user 1's voicemail
;
[play-voicemail-custom]
exten => playmail,1,Answer()
exten => playmail,2,Set(RANDSOUND=/var/spool/asterisk/
voicemail/default/1/INBOX/${SHELL(ls /var/spool/asterisk/
voicemail/default/1/INBOX | shuf -n 1 | head -n1 | cut -f1 -
d".")})
exten => playmail,3,NoOp(*${RANDSOUND}*)
exten => playmail,4,Background(${RANDSOUND})
exten => playmail,5,Goto(2)
exten => _.,1,Goto(ivr-1,s,1)
exten => playmail,n,Hangup()
;--------------------------------------------------------
```

This code creates a new extension called play-voicemail-custom.
On answering, it grabs a

 random audio file from the Inbox of user 1, plays the file, then once played goes back to step 2.

To have this extensions recognised, run `asterisk -rx "core restart now"` from the terminal

Once this is done, you need to navigate in the UI to setup a Custom Destination, under the Admin tab.

Set the target as `play-voicemail-custom,playmail,1`.

The first part is the name of the extension in square brackets above, then the extensions 'number' is playmail, and it simulates pressing 1, or going to the first step.

Security

Setting up Fail2Ban

This is a free piece of software that provides a little security agains things like web crawlers which could make brute force attacks on your server.

I'm not an expert by any means, but here's what worked from me after looking at a few different resources online:

1. Install Fail2Ban in the terminal with this command `apt -y install fail2ban`
2. Add security logs to Asterisk
 - Edit the file `nano /etc/asterisk/logger.conf`
 - In the [logfiles] section, add `security` to both `messages` and `console`
 - In the [general] section, uncomment the default time format: `dateformat=%F %T ; ISO 8601 date format`
 - Run `asterisk -rx 'logger reload'` to reload

3. Setup a `jail` for Asterisk
 - `nano /etc/fail2ban/jail.d/asterisk-jail.conf` and add the following:
```
 [asterisk]
 enabled = true
 logpath = /var/log/asterisk/security
 action = iptables-allports[name=ASTERISK, protocol=all]
 maxretry = 5
 bantime = 60
```
 - I had to run `touch /var/log/asterisk/security` before that would work to create the log file.
4. Setup a `jail` for ssh
 - `nano /etc/fail2ban/jail.d/ssh-jail.conf` and add
```
 [sshd]
 enabled = true
 mode = normal
 maxretry = 5
 bantime = 60
```

5. Run `systemctl start fail2ban` and `fail2ban-client status` to see that your two jails were setup correctly.

[I found this article quite helpful](https://hotkey404.com/asterisk-security-blocking-network-attacks-with-fail2ban/)


1985/6
This establishment is is an official sponsor of
London
Gay Switchboard
01·837·7324

Links

- Setup FreePBX with a 3G Dongle https://github.com/MatejKovacic/RasPBX-install

- Great summary of top level Asterisk knowledge https://geek-university.com/asterisk-course/

- More about Machine Streams https://www.machine-streams.com/blog/post/queer-phone/