Making a private server Ambulant[*]

[*] Moving or walking about

# Introduction: Who is ROSA?

This booklet covers:

* Adding your server to a Virtual Private Network (VPN) with tinc

* Adding public web access to the machine with a reverse proxy

This is part of documentation of the Rosa server

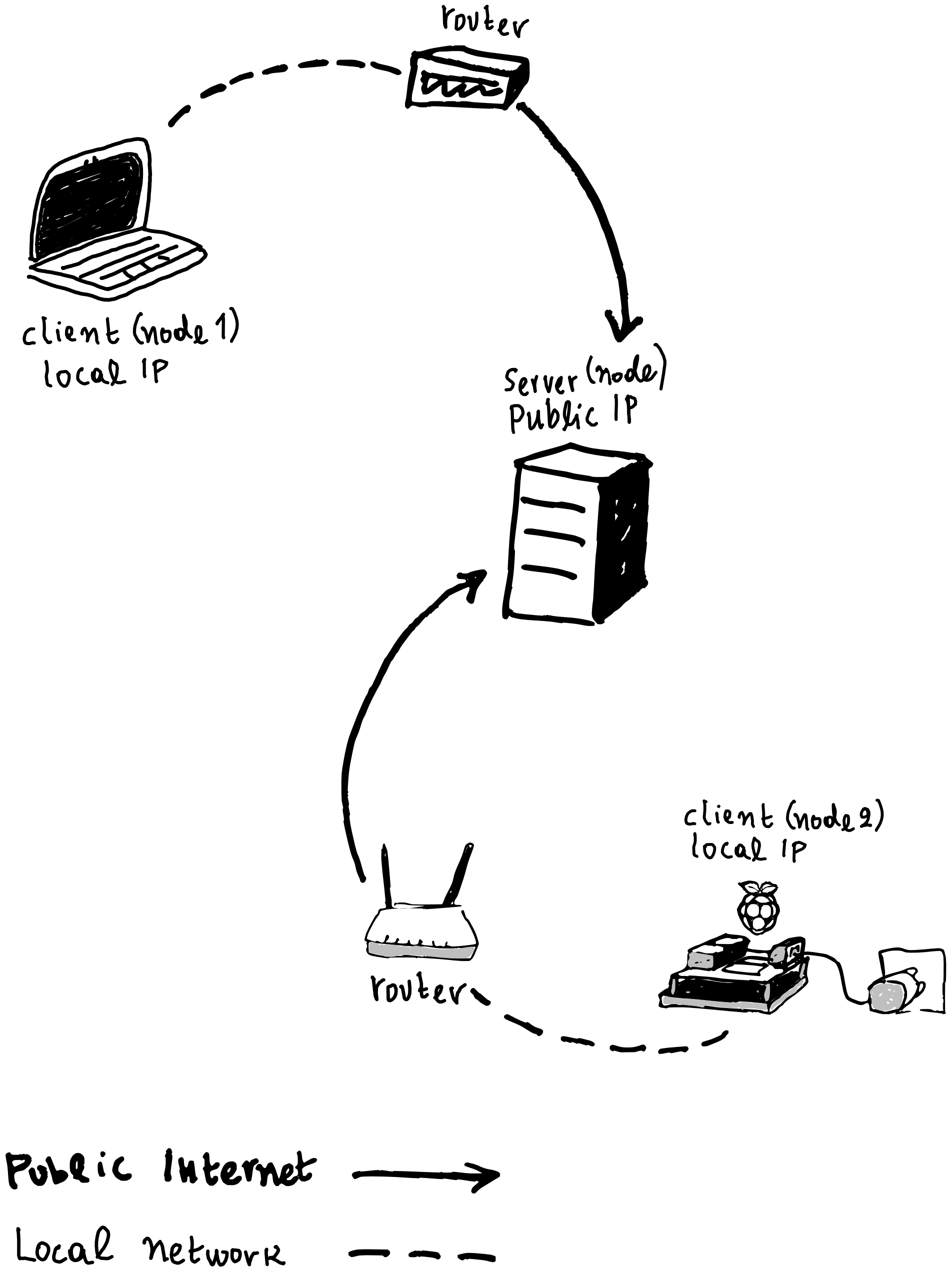
https://hub.vvvvvvaria.org/rosa/pad/p/rosa-log

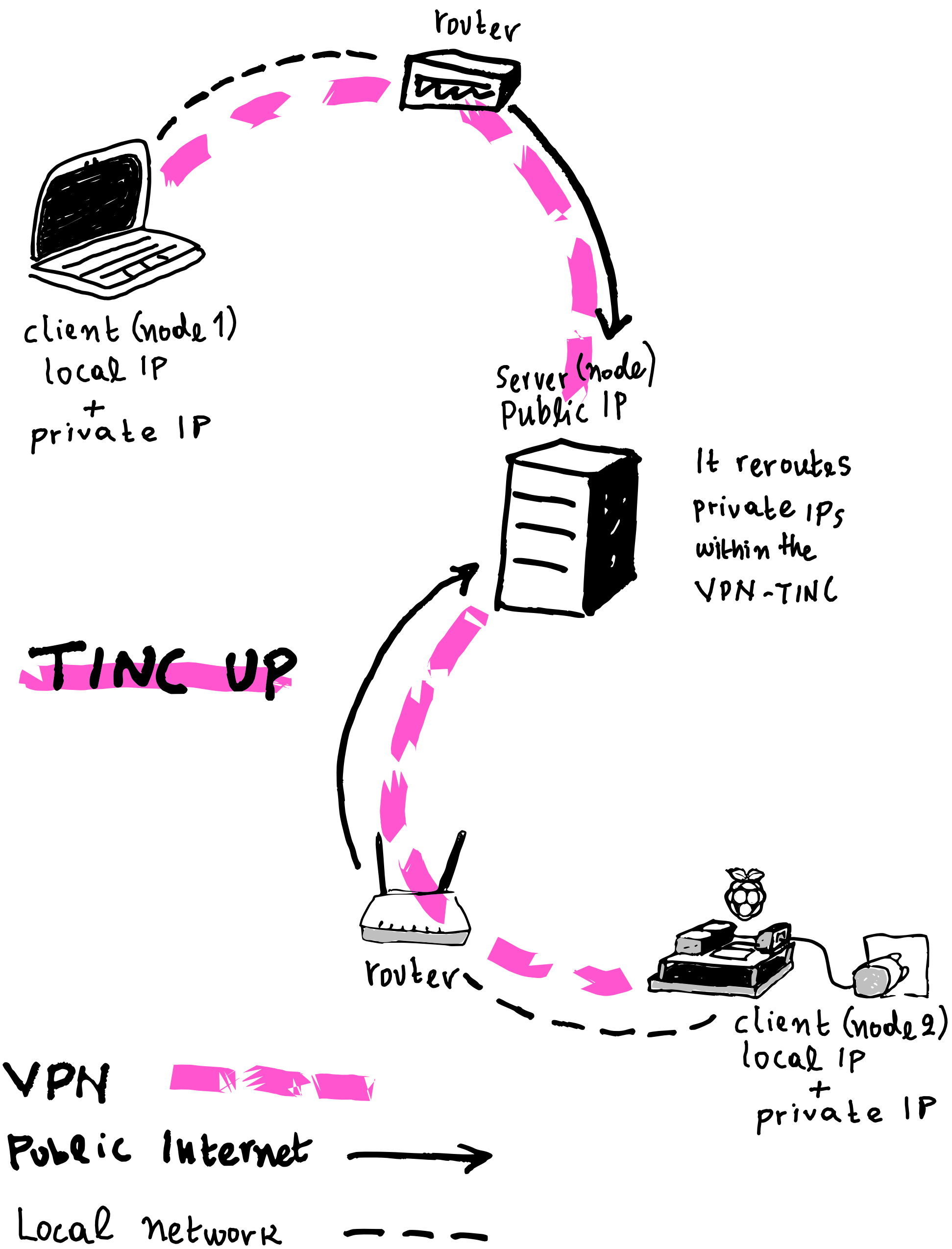
The idea is that this documentation can serve as a part of a larger collection about feminist servers, allowing different choices to be made on how to configure a local server, with the starting point of a private network.

Rosa is a feminist server that has travelled to all the locations of the ATNOFS program (A Traversal Network of Feminist Servers), providing the infrastructure for documenting the 2-day events and for publishing chapters from each of the six partners of the project (Varia, HYPHA, LURK, esc, Feminist Hack Meetings, Constant).

Rosa is not only its constituting hardware or software, but also the multitude of relations which are created around the making, maintening and passing on of this infrastructure: the processes that are performed, the affective charge of their actioning, the community around them.

Using a small situated network is fine for many settings. However, when not all participants can be connected to a single network (or hotspot), things can get complicated. Setting up a larger networks are hard to maintain particularly for situations that are dynamic. This guide proposes the use of the free software tool *tinc* to create a *Virtual Private Network* or *VPN*. VPNs


router
client (node 1)
local IP
Server (node)
Public IP
client (node 2)
local IP
router
Public Internet
Local network


router
client (node 1)
local IP
+
private IP
Server (node)
Public IP
It reroutes
private IPs
within the
VPN-TINC
TINC UP
router
client (node 2)
local IP
+
private IP
VPN
Public Internet
Local network

# Installing the operating system: selecting the distribution

This guide has been developed and tested using a open hardware single board computer called the Olimex, and the GNU-Linux software distrubution called [Armbian](https://www.armbian.com/). In principle the other steps in this guide should work for other linux-based platforms.

## Installing Armbian on Olimex

Install instructions for Debian on Olimex A20 Oluxolino

https://images.olimex.com/release/a20/ -

https://images.olimex.com/release/a20/A20-OLinuXino-bullseye-mini mal-20220928-143706.img.7z

https://github.com/OLIMEX/OLINUXINO/blob/master/DOCUMENTS/

OLIMAGE/Olimage-guide.pdf

Find olimex name or IP from the local router http://192.168.0.1/

```
ssh olimex@192.168.0.78
```

password = olimex

## Update your software

```
$ sudo apt update

$ sudo apt upgrade
```

You now have a working server. The next step is to install a VPN.

# TINC - putting a VPN node on a VPS

Tinc is a peer-to-peer software for virtual networks. This means that it does not distinguish between servers and clients. All computers are equal partipants in the network called *nodes*. There is no strict hierarchy where servers have a special role and clients are somehow limited.

In an ideal world, installing tinc on your laptop and on the rosa server would allow you to connect to each other no matter where you or the rosa physically were located. In practice, it's more complicated. The network works by nodes sharing information between all the other nodes that it can see. In this way information about all the participants are shared on the network. The trouble is that a laptop and the rosa server when connected to the network in different networks aren't directly able to see each other and share information. In this guide, we use a public server that is also a node on the VPN. This server then is visible to other nodes and acts as a bridge between private nodes, however they are connected.

In this guide we refer to the public server as the public node, and all other nodes like the rosa server and individual laptops as private nodes.

This guide assumes that either you or someone you are working with has system administrator access to a public computer (ie a computer that has a public fixed IP address, such as a dedicated server or a virtual private server or VPS) where you can install and run tinc, and create an initial network. Once this step is completed, you can follow instructions for installing + configuring tinc on additional nodes, such as the rosa server, and your own laptop.

This first step is the same for both the public server and additional (private) nodes.

At the time of writing this, the armbian (and other debian) distributions are still using an older 1.0 version of tinc. This guide will compile a *pre-release* version of the tinc 1.1 software. It's essential that all the computers on the virtual network use the same version of tinc.

# Compiling + installing tinc (all nodes: public + private)

You need to do this step for all nodes, both public and private.

Official 1.1 docs: <https://tinc-vpn.org/documentation-1.1/>

As we are compiling the software, we need to use *apt* to install some development tools & dependencies, then we can download the source code of tinc, and follow the standard steps to compile and install the software (configure, make, make install).

```
sudo apt install build-essential automake libssl-dev liblzo2-dev libbz2-dev zlib1g-dev libncurses5-dev libreadline-dev
cd ~
```

```
wget https://www.tinc-vpn.org/packages/tinc-1.1pre17.tar.gz
tar xvf tinc-1.1pre17.tar.gz
cd tinc-1.1pre17
./configure
make
sudo make install
```

Once installed create configuration dir, all the configuration of tinc is in this folder. Using tinc subcommands (like invite / join), result in changes to the files in this folder.

```
$ sudo mkdir -p /usr/local/etc/tinc/
```

The tinc executable is installed in

/usr/local/sbin/tinc

This means that you can only run tinc as sudo, since sbin directory saves binary executables that can be ran only by sudo (s+bin)

## Create a systemd service file

Systemd is a way to manage (start/stop) services like servers. Tinc is such a service. You can create new service files in the folder

/etc/systemd/system. In this case we create a special kind of service file (that has an @ in the name) that allows it to work for multiple network names. In this case the NETNAME we'll use is "constant". Hence the variable "i" inside the unit file would be replaced by the NETWORK name we give after the tinc@<NETWORK)

sudo nano /etc/systemd/system/tinc@.service

```
[Unit]
Description=Tinc (%i)
After=network.target
[Service]
Type=simple
WorkingDirectory=/usr/local/etc/tinc
ExecStart=/usr/local/sbin/tincd -D -n %i
ExecReload=/usr/local/sbin/tincd -D -n %i -kHUP
TimeoutStopSec=5
Restart=always
RestartSec=60
[Install]
WantedBy=multi-user.target
```

Tinc stores all configuration in /usr/local/etc/tinc

TINC allows multiple private networks to be defined, each is a folder

with the name of the network in /usr/local/etc/tinc

(If you mess something up, you can delete the files that are there).

# tinc init: Create the intial network (public node, once only)

NB: This step only has to happen once

We will create a virtual network named "constant" on the public node. Once this network is created, we will use the public node to "invite" other nodes, including the rosa server, and optinally your laptop into this network. Invited nodes can then use tinc's *join* command to use the invite link.

The generic form for initializing a new network named NETNAME

```
sudo tinc -n NETNAME init NODENAME
```

SO I DID

```
sudo tinc -n constant init hub



Generating 2048 bits keys:

.................................+++++ p

........................................+++++ q

Done.

Generating Ed25519 keypair:

Done.
```

# tinc invite/join: Join an existing VPN: A dialog betwen you and your sys admins

Make sure tinc is already installed (described above).

Here are the steps:

* Request from the VPN public node sysadmin: Tell them your desired name for your machine

* The VPN server sysadmin will assign you a private IP address (on the VPN). Sysadmin will run *tinc invite* on the public node to produce an invite link. They should give these two things to you (invite link + IP address).

* run tinc join with the invite code on the local machine

* set the ip address on the VPN with tinc add subnet

* edit the tinc up

## invite + join

Example:

Wendeline wants to request access to the VPN called "constant"

from sys admin Michelle.

Wendeline: Hey Michelle, my machine is called "wendeline"

Michelle: Let me check for the next available number...

The subnet of the constant VPN is "10.10.12.x", I can give you

10.10.12.53.

This is the list of private addresses (handy to save it a file, e.g under the /usr/local/etc/tinc/):

```
10.10.12.1       hub
10.10.12.52       rosamex
10.10.12.53       wendeline
```

(they record in the file wendeline = 10.10.12.53)

Then Michelle runs:

```
 sudo tinc -n constant invite wendeline
```

Michelle (replying to Wendeline): OK Wendeline, here's your invite code, just run:

```
$ sudo tinc -n constant join

79.99.202.57/zVublahX7LaCWXJdBzd03jNn48bxuN83jVE_26VnL
```

```
 and then[sudo] password for wendy:
 Connected to 79.99.202.57 port 655...
 ...................+++++ p
 ...................................................+++++-.................................
...........+++++ q
 Configuration stored in: /usr/local/etc/tinc/constant
 Invitation successfully accepted.
```

## Set the VPN ip address in the 10.10.12.0 subnet

Wendeline then runs the command to set their IP address on the VPN:

```
    sudo tinc -n constant add subnet 10.10.12.53
```

## Edit the tinc-up file

Wendeline then edits their tinc-up file to also include the new VPN IP address, they make the following replacements inside the tinc-up file:

    <your vpn IP address> => 10.10.12.53

    <netmask of whole VPN> => 255.255.255.0

```
$ sudo nano /usr/local/etc/tinc/constant/tinc-up
```

```
#!/bin/sh
#echo 'Unconfigured tinc-up script, please edit '$0'!'
ifconfig $INTERFACE 10.10.12.53 netmask 255.255.255.0
```

## Start tinc for the new network

```
$ sudo systemctl start tinc@constant
```

To make the VPN start automatically (important for the rosa server):

```
$ sudo systemctl enable tinc@constant
```

# Installing a webserver: NGINX

nginx [engine x] is an HTTP (web) server that can also act as a reverse proxy server, a mail proxy server, and a generic TCP/UDP proxy server

```
# apt install nginx
```

Then you can use systemctl to check the status, reload (when settings change) and restart (when there's troubles) the server.

```
# systemctl status nginx
# systemctl reload nginx
# systemctl restart nginx
```

## Make the web server files writable by the main user

Make Olimex user part of the www-data group, including all the files to be able to edit the files in the html folder

```
$ sudo chown -R olimex:www-data html
```

Adding write mode for the group

```
# chmod -R g+w /var/www/html
```

Edit the welcome page

```
$ cd /var/www/html
$ less index.nginx-debian.html # copy the info of the welcome page of Debian and tweak it
$ vi hello-rosas.html # you can use nano instead of vi :-)
```

In the next step you will configure nginx to use the custom page as an index.

# Configure the ROSA site (Nginx)

```
$ cd /etc/nginx/

$ sudo nano sites-available/rosamex.conf
```

```
server {
    listen 80;
    listen [::]:80;
    server_name _;
    root /var/www/html;
    index hello-rosas.html;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

```
$ sudo nginx -t

$ # rm /etc/nginx/sites-enabled/default

$ sudo ln -s /etc/nginx/sites-available/rosamex.conf

/etc/nginx/sites-enabled/rosamex.conf

$ sudo systemctl reload nginx
```

# HOWTO REACH ROSAMEX FROM THE PUBLIC INTERNET

Once the VPN is setup, we can use the fact that the public server can see the rosa server (via the VPN) to create a public portal to the rosa server. The public server again acts as a bridge, with this time a reverse proxy configured on the public server's web server, that points to the rosa server's IP address on the VPN.

## Installing a Reverse Proxy (with apache2)

On the public server, we create a reverse proxy to have a public URL that points to the rosamex server.

Add an apache config at /etc/apache2/sites-available/rosamex.conf

online example

```
<VirtualHost *:80>
    ServerName rosamex.constantvzw.org:80
    ProxyPass / http://10.10.12.52/
    ProxyPassReverse / http://10.10.12.52/
</VirtualHost>
```

## Installing a Reverse Proxy (with nginx)

Add an nginx config at /etc/nginx/sites-available/rosamex.conf

```
server {
    listen       80;
    listen  [::]:80;
    server_name  rosamex.constantvzw.org;
    location / {
      proxy_set_header        Host $host;
      proxy_set_header        X-Real-IP $remote_addr;
      proxy_set_header        X-Forwarded-For
$proxy_add_x_forwarded_for;
      proxy_set_header        X-Forwarded-Proto $scheme;
      proxy_pass           http://10.10.12.51;
      proxy_read_timeout  90;
    }
    location / {
        rewrite ^ https://$host$request_uri? permanent;
    }
}
```

# Troubleshooting

It may be necessary to open port 655 if your firewall is configured to block this port.

You may need to install net-tools if you see problems finding ifconfig.

```
$ sudo apt install net-tools
```

Is tinc port open? check with:

```
$ sudo lsof -i -P -n
```

You can run tinc manually if there are problems...

```
$ sudo tincd -n constant -D -d3
```

Press ctrl-c to stop

Check your tinc-up file, make sure the right lines are comments out with the hash character.

```
$ sudo systemctl status tinc@constant
```

check ip addr or ifconfig to see the new private network is up

```
$ sudo ifconfig
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc
noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
[...]
8: constant:
<POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu
1500 qdisc pfifo_fast state UNKNOWN group default qlen 500
    link/none
    inet 10.10.12.50/24 scope global constant
       valid_lft forever preferred_lft forever
    inet6 fe80::9b4f:9037:c97e:38b7/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

NOTE: now that you are on the VPN, you can ssh to other machines on the network

```
    ssh username-on-other-machine@vpn-address-of-other-machine
```

for example if we want to ssh to olimex machine (assigned to rosemex subnet) we run:

```
$ ssh olimex@10.10.12.52 -o PasswordAuthentication=yes
```

Check web server errors, e.g the nginx error log:

```
$ tail -f /var/log/nginx/error.log
```

# Extra tools

## INSTALL TMUX

TMUX allows multiple terminal sessions to be accessed simultaneously in a single window. Different users acting as one shell user.

Trust is essentiel here

for common terminal based work

```
$ sudo apt-get install tmux
```

become sudo with `sudo su` and create a new tmux session

```
# tmux new -s mysession
# tmux new -s rosamex
```

Join an existing session

```
# tmux attach -t mysession
```

https://phoenixnap.com/kb/tmux-tutorial-install-commands

https://tmuxcheatsheet.com/

# Future editions

The Rosa project is currently documented, though in a fully concise way. The plan is to produce and publish future volumes that cover:

* Etherpad

* Etherdump

* Octomode

* Customisation