

# Server Setup

## Requirements

Here are some minimum requirements you absolutely need to run a node.

### Exits

1024 mb RAM  
5 GB Diskspace  
Dualcore

With this setup you can run a 50 mbit/s relay without running out of space or memory. To run anything faster you need at least twice as much power.

Gigabit servers should at least have the following settings

8 GB RAM  
30 GB Diskspace  
Octocore (with AES-NI enabled)

### Relays

2048 mb RAM  
5 GB Diskspace  
Dualcore

Relay setups need nearly the same requirements as Exit Nodes.

### Bridges

128 mb RAM  
2 GB Diskspace  
Singlecore

A Bridge doesn't need much of anything. Some more RAM could be useful if you decide to run more than one Bridge on the server. Nodes

If you get an SSL/TLS error while using wget to download our scripts from bitbucket, then you have to use the `-no-check-certificate` option!

# Exit Nodes

**First Step:** <sxh bash;>apt-get install build-essential echo "deb

<http://deb.torproject.org/torproject.org> <DISTRIBUTION> main" » /etc/apt/sources.list gpg -keyserver keyserver.c3l.lu -recv 886DDD89 gpg -export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add - apt-get update apt-get install deb.torproject.org-keyring apt-get install tor </sxh>

**Second Step:** <sxh bash;>apt-get install openssl libssl-dev cpan cpan> install

Net::Address::IP::Local Net::SSLeay IO::Socket::SSL LWP::Protocol::https WWW::Mechanize wget

<https://bitbucket.org/fvde/tor-autoconfig/raw/tip/autoconf.pl> perl autoconf.pl exit [YOUR-NODE-NICKNAME] [NETWORK SPEED] [METERED|UNMETERED] ([TRAFFIC LIMIT]) wget -O

/root/family\_updater.pl [https://bitbucket.org/fvde/tor-autoconfig/raw/tip/family\\_updater.pl](https://bitbucket.org/fvde/tor-autoconfig/raw/tip/family_updater.pl) crontab -e

</sxh> \*/10 \* \* \* \* perl /root/family\_updater.pl <sxh bash;>service tor restart wget -O

/root/update\_server.pl [https://bitbucket.org/virii/update-ennstatus/raw/tip/update\\_server.pl](https://bitbucket.org/virii/update-ennstatus/raw/tip/update_server.pl) perl

/root/update\_server.pl [torrc] ([torrc2]) ([torrc3]) crontab -e </sxh> \*/10 \* \* \* \* perl

/root/update\_server.pl torrc

**Third Step:** <sxh bash;>apt-get install iftop htop python-pip pip install nix wget

<https://download.adamas.ai/dlbase/Stuff/sthttpd/sthttpd-2.27.0.tar.gz> tar -zxvf sthttpd-2.27.0.tar.gz cd sthttpd-2.27.0 adduser thttpd ./configure make make install wget -O /etc/thttpd.conf

<https://exit-install.enn.lu/thttpd.conf> mkdir /var/www wget -O /etc/init.d/thttpd

<https://exit-install.enn.lu/thttpd.init> chmod 0755 /etc/init.d/thttpd service thttpd start </sxh> **Forth**

**Step:** <sxh bash;>apt-get install vnstat cd /var/www touch vnstat.png vnstat\_d.png vnstat\_m.png

vnstat.xml crontab -u thttpd -e </sxh> \*/10 \* \* \* \* /usr/bin/vnstat -vs -o

/var/www/vnstat.png -i eth0 >/dev/null 2>&1 ;

\*/10 \* \* \* \* /usr/bin/vnstat -d -o /var/www/vnstat\_d.png -i eth0 >/dev/null

2>&1 ;

1 3 \* \* \* /usr/bin/vnstat -m -o /var/www/vnstat\_m.png -i eth0 >/dev/null

2>&1 ;

1 3 \* \* \* /usr/bin/vnstat -xml > /var/www/vnstat.xml 2>/dev/null ;

<sxh bash;>vnstat -u -i eth0 wget -O /var/www/index.html <https://exit-install.enn.lu/exit-node.html>

chown -R thttpd:thttpd /var/www </sxh> **Fifth Step:**

Control your torrc. A functional Exit torrc has to look like [this!](#) <sxh bash;>service tor restart</sxh>

# Bridges

**First Step:** <sxh bash;>apt-get install build-essential python-dev python-pip libgmp-dev apt-get

install golang echo "deb <https://deb.torproject.org/torproject.org> <DISTRIBUTION> main" »

/etc/apt/sources.list echo "deb <https://deb.torproject.org/torproject.org> tor <DISTRIBUTION> main" »

/etc/apt/sources.list echo "deb <https://deb.torproject.org/torproject.org> obfs4proxy main" »

/etc/apt/sources.list curl

<https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89.asc> | gpg

-import gpg -export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add - apt-get

update apt-get install deb.torproject.org-keyring apt-get install tor nix obfsproxy obfs4proxy fteproxy

</sxh> **Second Step:** <sxh bash;>apt-get install openssl libssl-dev cpan cpan> install

Net::Address::IP::Local Net::SSLeay IO::Socket::SSL LWP::Protocol::https WWW::Mechanize wget

```
https://bitbucket.org/fvde/tor-autoconfig/raw/tip/autoconf.pl perl autoconf.pl bridge [YOUR-NODE-  
NICKNAME] [NETWORK SPEED] [METERED|UNMETERED] ([TRAFFIC LIMIT]) wget -O  
/root/update_server.pl https://bitbucket.org/virii/update-ennstatus/raw/tip/update_server.pl crontab -e  
</sxh> */10 * * * * perl /root/update_server.pl torrc
```

### Third Step:

Control your torrc. A functional Bridge torrc has to look like [this!](#) <sxh bash;>service tor  
restart</sxh>

## Multiple Tor Processes

```
<sxh bash;>wget https://bitbucket.org/fvde/tor-autoconfig/raw/tip/autoconf.pl perl autoconf.pl bridge  
[TEXTFILE-CONTAINING-NAMES] [NETWORK SPEED] [METERED|UNMETERED] ([TRAFFIC LIMIT]) </sxh>
```

There is a modified initscript that makes it easy to manage multiple Tor configurations on one machine.

```
<sxh bash;> cd /etc/init.d wget -O tor https://exit-install.enn.lu/tor.initd chmod +x tor </sxh>
```

Starts/Stops multiple tor configs. Like torrc0 torrc1 torrc2

From:

<http://wiki.enn.lu/> - **Frënn vun der Ënn A.S.B.L.**

Permanent link:

<http://wiki.enn.lu/doku.php?id=server-setup>

Last update: **2019/03/06 16:40**

