

# 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.adamas.ai -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;>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 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 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 apt-get install htop apt-get install tor-arm wget ftp://opensource.dyc.edu/pub/sthttpd/sthttpd-2.26.3.tar.gz tar -zxvf sthttpd-2.26.3.tar.gz cd sthttpd-2.26.3 adduser thttpd ./configure make make install wget -O /etc/thttpd.conf http://exit-install.enn.lu/thttpd.conf mkdir /var/www wget -O /etc/init.d/thttpd http://exit-install/enn.lu/thttpd.init chmod 0755 /etc/init.d/thttpd service thttpd start </sxh>` **Forth Step:** `<sxh bash;>apt-get install vnstati cd /var/www touch vnstat.png vnstat_d.png vnstat_m.png vnstat.xml crontab -u thttpd -e </sxh> */10 * * * * /usr/bin/vnstati -vs -o /var/www/vnstat.png -i eth0 >/dev/null 2>&1 ; */10 * * * * /usr/bin/vnstati -d -o /var/www/vnstat_d.png -i eth0 >/dev/null 2>&1 ; 1 3 * * * /usr/bin/vnstati -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 http://exit-install.enn.lu/index.html.ascii 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 echo "deb http://deb.torproject.org/torproject.org <DISTRIBUTION> main" » /etc/apt/sources.list echo "deb http://deb.torproject.org/torproject.org tor-nightly-master-<DISTRIBUTION> main" » /etc/apt/sources.list echo "deb http://fteproxy.org/deb/ stable/" » /etc/apt/sources.list gpg -keyserver keyserver.adamas.ai -recv 886DDD89 gpg -keyserver keyserver.adamas.ai -recv 6B898EE18FBA6390 gpg -export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add - gpg -export 6B898EE18FBA6390 | apt-key add - apt-get update apt-get install deb.torproject.org-keyring apt-get install tor tor-arm obfsproxy fteproxy </sxh>` **Flashproxy: (optional)** `<sxh bash;>apt-get install git golang git clone https://git.torproject.org/flashproxy.git cd flashproxy; make; make install </sxh>` Tor config

```
ExtORPort auto
```

```
ServerTransportPlugin websocket exec /usr/local/bin/pt-websocket-server -port 9901
```

**Second Step:** `<sxh bash;>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>
```

From the [torservers.net](http://torservers.net) wiki:

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://gist.githubusercontent.com/7adietri/9122199/raw/4ed71b894eddbdfb0e241fa06bb583a19f0cc89/tor chmod +x tor </sxh>
```

The relevant tor\*.cfg settings per relay are (change "0" to "1" etc):

```
DataDirectory /var/lib/tor/0  
PidFile /var/run/tor/tor0.pid  
Log notice file /var/log/tor/notices0.log
```

From:

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

Permanent link:

<http://wiki.enn.lu/doku.php?id=server-setup&rev=1406968144>



Last update: **2014/12/04 17:48**