

## SOMMARIO

1. Check-list pre installazione
2. Comandi vari
3. Installazione di bitcoincore
4. Configurazione dell'autostart di bitcoincore
5. Installazione di TOR
6. Eventuali aggiornamenti di bitcoincore
7. Installazione di Fulcrum
8. Installazione di Lightning Network Daemon (LND)
9. Installazione di Ride The Lightning (RTL)
10. Installazione di LiT, Loop e Boltz
11. Installazione di Thunderhub
12. Installazione di Mempool

**N.B.:** La guida è derivata dalla serie “Ubuntu Node Box 2022” di Ketan che puoi trovare qui <https://www.youtube.com/c/MinistryofNodes/videos>

---

### Check-list pre installazione

**N.B.:** Si dà per scontata un'installazione “standard” di Ubuntu Server 22.04

**Dotazione HW minima richiesta:**

**Dell Optiplex 3020 SFF/USFF o superiore** (es.: <https://tinyurl.com/55ya8t85>)

**CPU Intel i5**

**RAM 8Gb**

**SSD 1Tb/2Tb**

**(Volendo si può usare anche una macchina virtuale)**

Nome nodo: nodebox

Utente: **satoshi** (cambiare a piacere)

Pwd: password1234 (cambiare a piacere)

rpcauth

Utente: **bitcoin** (o a piacere)

Pwd: **bitcoin** (o a piacere)

**N.B.:** La guida è realizzata per la versione 22.0 di bitcoincore, 1.8.2 di Fulcrum, 0.15.4-beta di LND, 0.12.3 di RTL, 0.8.3-alpha di Lightning Terminal, 1.2.6 di Boltz, 0.13.6 di Thunderhub, 2.12.2 di Docker e 2.5.0-dev di Mempool

**Per installare versioni più recenti fare riferimento ai siti delle varie applicazioni.**

---

### Comandi vari

bitcoin-cli getblockchaininfo (ottiene lo stato in realtime della blockchain)

bitcoin-cli getconnectioncount (ottiene il numero di peer collegati)

bitcoin-cli getnetworkinfo (ottiene l'indirizzo TOR .onion del proprio nodo)

bitcoin-cli getpeerinfo (verifica che i peer siano connessi attraverso TOR)

bitcoin-cli gettxoutsetinfo (verifica il total supply di bitcoin)  
systemctl status nome\_servizio (verifica che il servizio sia girando correttamente)  
systemctl start nome\_servizio (avvia il servizio)  
systemctl stop nome\_servizio (arresta il servizio)  
whereis "nomefile" (mostra la posizione di un file o di un servizio)  
tail -f .bitcoin/debug.log (mostra il log di bitcoincore in realtime)  
journalctl -fu fulcrum.service (mostra il log di fulcrum in realtime)  
sudo systemctl enable nome\_servizio (inserisce un servizio nell'avvio automatico al boot)  
sudo systemctl start nome\_servizio (avvia un servizio)  
sudo systemctl stop nome\_servizio (arresta un servizio)  
sudo systemctl status nome\_servizio (verifica che il servizio stia girando correttamente)

-----

indirizzo\_ip\_nodo:3000 RTL

indirizzo\_ip\_nodo:8443 Lightning Terminal

indirizzo\_ip\_nodo:4000 Thunderhub

indirizzo\_ip\_nodo:4080 Mempool

## Installazione di bitcoincore

Siti di riferimento:

<https://bitcoin.org/en/download>

<https://bitcoincore.org/en/download/>

<https://github.com/bitcoin-core/bitcoincore.org>

```
mkdir downloads
```

```
cd downloads
```

```
wget https://bitcoincore.org/bin/bitcoin-core-22.0/bitcoin-22.0-x86_64-linux-gnu.tar.gz
```

```
wget https://bitcoincore.org/bin/bitcoin-core-22.0/SHA256SUMS
```

```
wget https://bitcoincore.org/bin/bitcoin-core-22.0/SHA256SUMS.asc
```

```
sha256sum --ignore-missing --check SHA256SUMS
```

```
gpg --keyserver hkps://keys.openpgp.org --recv-keys  
E777299FC265DD04793070EB944D35F9AC3DB76A
```

```
gpg --keyserver hkps://keys.openpgp.org --recv-keys  
2688F5A9A4BE0F295E921E8A25F27A38A47AD566
```

```
gpg --keyserver hkps://keys.openpgp.org --recv-keys  
71A3B16735405025D447E8F274810B012346C9A6
```

```
gpg --verify SHA256SUMS.asc
```

```
tar xzf bitcoin-22.0-x86_64-linux-gnu.tar.gz
```

```
sudo install -m 0755 -o root -g root -t /usr/local/bin bitcoin-22.0/bin/*
```

```
bitcoind -daemon
```

```
cd
```

```
tail -f .bitcoin/debug.log (per verificare che il daemon stia girando e scaricando la blockchain)
```

(in .bitcoin) nano bitcoin.conf e copiare il seguente contenuto

```
server=1  
txindex=1  
daemon=1  
rpcport=8332  
rpcbind=0.0.0.0  
rpcallowip=127.0.0.1  
rpcallowip=10.0.0.0/8  
rpcallowip=172.0.0.0/8  
rpcallowip=192.0.0.0/8  
zmqpubrawblock=tcp://0.0.0.0:28332  
zmqpubrawtx=tcp://0.0.0.0:28333  
zmqpubhashblock=tcp://0.0.0.0:28334  
whitelist=127.0.0.1  
rpcauth=bitcoin:[copiare l'output di rpcauth.py – vedi più sotto]
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
cd downloads
```

```
wget https://raw.githubusercontent.com/bitcoin/bitcoin/master/share/rpcauth/rpcauth.py
```

```
chmod +x rpcauth.py
```

```
./rpcauth.py bitcoin bitcoin (user e password scelti sopra per la sezione rpcauth)
```

Copiare il risultato nella riga rpcauth=bitcoin: di bitcoin.conf in .bitcoin (vedi sopra)

```
bitcoin-cli stop
```

```
cd
```

```
tail -f .bitcoin/debug.log (per verificare che il daemon sia stoppato)
```

```
bitcoind (riavvio del daemon)
```

```
tail -f .bitcoin/debug.log (per verificare che il daemon stia girando e scaricando la blockchain)
```

**Lasciare che l'IBD (Initial Blockchain Download) termini correttamente prima di continuare.**

-----

**Configurazione dell'autostart di bitcoincore**

Portarsi in /etc/systemd/system e digitare

```
sudo wget https://raw.githubusercontent.com/bitcoin/bitcoin/master/contrib/init/bitcoind.service
```

sudo nano bitcoind.service e sistemare le seguenti sezioni come sotto riportato

```
[Service]
```

```
ExecStart=/usr/local/bin/bitcoind -daemon \  
          -pid=/run/bitcoind/bitcoind.pid \  
          -conf=/satoshi/.bitcoin/bitcoin.conf \  
          -datadir=/satoshi/.bitcoin
```

```
#ExecStartPre=/bin/chgrp bitcoin /etc/bitcoin
```

```
# Run as bitcoin:bitcoin
```

```
User= satoshi
```

```
Group= satoshi
```

```
# Deny access to /home, /root and /run/user
```

```
#ProtectHome=true
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
bitcoin-cli stop
```

```
sudo systemctl enable bitcoind
```

```
sudo systemctl start bitcoind
```

```
tail -f .bitcoin/debug.log (per verificare che il daemon stia girando e scaricando la blockchain)
```

-----

Siti di riferimento:

<https://bitnode.io/nodes>

```
cd ~ (Alt+126)
```

```
sudo apt install tor
```

```
sudo systemctl status tor (per verificare che TOR sia installato e funzionante)
```

```
sudo nano /etc/tor/torrc
```

Andare in fondo al file e aggiungere le seguenti righe

```
ControlPort 9051
```

```
CookieAuthentication 1
```

```
CookieAuthFileGroupReadable 1
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
sudo systemctl restart tor
```

```
sudo usermod -a -G debian-tor satoshi
```

```
nano .bitcoin/bitcoin.conf
```

Andare in fondo al file e aggiungere le seguenti righe

```
proxy=127.0.0.1:9050
```

```
listen=1
```

```
bind=127.0.0.1
```

```
onlynet=onion
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
sudo reboot
```

Verificare che tutto funzioni correttamente digitando

```
tail -f .bitcoin/debug.log (per verificare che il daemon stia girando e scaricando la blockchain)
```

```
bitcoin-cli getblockchaininfo
```

`bitcoin-cli getconnectioncount` – deve essere maggiore di 0. In caso contrario aggiungere manualmente un nodo prendendolo dal sito [bitnodes.io](https://bitnodes.io) “Networks – Tor Network (TOR)” e copiandolo in fondo al file `.bitcoin/bitcoin.conf` (esempio)

```
addnode=https://bitnodes.io/nodes/ygtqbpqwhkyy4yj6v2znog3let3ihjingvwo5bs4g5obj2ak3ufbnnid.onion-8333/
```

Quindi

```
sudo systemctl restart bitcoind
```

```
bitcoin-cli getconnectioncount (Per verificare che esistano delle connessioni)
```

Quando le connessioni superano le 5, si può rimuovere la riga appena aggiunta sopra e riavviare bitcoind

```
bitcoin-cli getnetworkinfo (Per ottenere l'indirizzo TOR .onion del proprio nodo)  
ipv4 e ipv6 devono avere "reachable": false mentre onion deve avere true
```

```
bitcoin-cli getpeerinfo (Per verificare che i peer siano connessi attraverso TOR)
```

```
bitcoin-cli gettxoutsetinfo (Per verificare il total supply di bitcoin)
```

---

**Aggiornamenti di bitcoincore**

Siti di riferimento:

<https://bitcoin.org/en/download>

<https://bitcoincore.org/en/download/>

<https://github.com/bitcoin-core/bitcoincore.org>

`bitcoin-cli --version` (Per verificare la versione che sta girando)

`cd downloads`

`rm SHA256SUMS`

`rm SHA256SUMS.asc`

Andare su [bitcoincore.org/bin](https://bitcoincore.org/bin) e copiare il link della versione aggiornata e delle firme

`wget https://bitcoincore.org/bin/bitcoin-core-23.0/bitcoin-23.0-x86_64-linux-gnu.tar.gz`

`wget https://bitcoincore.org/bin/bitcoin-core-23.0/SHA256SUMS`

`wget https://bitcoincore.org/bin/bitcoin-core-23.0/SHA256SUMS.asc`

`sha256sum --ignore-missing --check SHA256SUMS`

`gpg --verify SHA256SUMS.asc` (Verifica integrità e firme)

`tar xzf bitcoin-23.0-x86_64-linux-gnu.tar.gz` (Estrazione dello .zip)

`bitcoin-cli stop` (arresta i servizi)

`sudo install -m 0755 -o root -g root -t /usr/local/bin bitcoin-23.0/bin/*`

`sudo systemctl start bitcoind`

-----

**Installazione di Fulcrum**



Siti di riferimento:

<https://github.com/cculianu/Fulcrum/releases/>

Scaricare in /downloads il file da github

```
wget https://github.com/cculianu/Fulcrum/releases/download/v1.8.2/Fulcrum-1.8.2-x86_64-linux.tar.gz
```

```
cd ..
```

```
mkdir fulcrum
```

```
mkdir fulcrum_db
```

```
cd downloads
```

```
tar xvf Fulcrum-1.8.2-x86_64-linux.tar.gz (estrae lo .zip)
```

```
mv Fulcrum-1.8.2-x86_64-linux/* /home/satoshi/fulcrum (sposta tutto nella cartella creata)
```

```
cd ..
```

```
cd fulcrum
```

```
openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem
```

Premere invio a ogni domanda fino alla fine

(Crea il certificato e le relative chiavi di sicurezza per la comunicazione criptata)

```
mv fulcrum-example-config.conf fulcrum.conf (rinomina un file di esempio nel file di config)
```

```
nano fulcrum.conf (e editare di conseguenza le varie sezioni con le info seguenti)
```

```
datadir = /home/satoshi/fulcrum_db  
rpcuser = bitcoin  
rpcpassword = bitcoin  
ssl = 0.0.0.0:50002  
cert = /home/satoshi/fulcrum/cert.pem  
key = /home/satoshi/fulcrum/key.pem  
peering = false  
fast-sync = 5000
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
sudo nano /etc/systemd/system/fulcrum.service e inserire le righe di seguito
```

```
[Unit]  
Description=Fulcrum  
After=network.target
```

```
[Service]  
ExecStart=/home/satoshi/fulcrum/Fulcrum /home/satoshi/fulcrum/fulcrum.conf  
User= satoshi  
LimitNOFILE=8192  
TimeoutStopSec=30min
```

[Install]

WantedBy=multi-user.target

Salvare e uscire (Ctrl+X--Y--Invio)

cd ..

sudo systemctl enable fulcrum.service

sudo systemctl start fulcrum.service

sudo systemctl status fulcrum.service

journalctl -fu fulcrum.service (log)

**Lasciare che Fulcrum termini correttamente la sincronizzazione prima di continuare.**

df -h (verifica lo stato del disco e delle varie partizioni)

cd /home/**satoshi** fulcrum\_db

du -h (verifica lo stato della directory di fulcrum)

---

## Installazione di Lightning Network Daemon (LND)

Siti di riferimento:

<https://raspbolt.org/guide/lightning/lightning-client.html>

<https://github.com/lightningnetwork/lnd/releases>

```
cd downloads
```

```
wget https://github.com/lightningnetwork/lnd/releases/download/v0.15.4-beta/lnd-linux-amd64-v0.15.4-beta.tar.gz
```

```
curl https://raw.githubusercontent.com/lightningnetwork/lnd/master/scripts/keys/guggero.asc | gpg -import
```

```
wget https://github.com/lightningnetwork/lnd/releases/download/v0.15.4-beta/manifest-guggero-v0.15.4-beta.sig
```

```
wget https://github.com/lightningnetwork/lnd/releases/download/v0.15.4-beta/manifest-v0.15.4-beta.txt
```

```
gpg --verify manifest-guggero-v0.15.4-beta.sig manifest-v0.15.4-beta.txt
```

```
sha256sum --check manifest-v0.15.4-beta.txt --ignore-missing
```

```
tar -xzf lnd-linux-amd64-v0.15.4-beta.tar.gz
```

```
sudo install -m 0755 -o root -g root -t /usr/local/bin lnd-linux-amd64-v0.15.4-beta/*
```

```
lnd --version
```

```
cd ..
```

```
mkdir .lnd
```

```
cd .lnd
```

```
wget https://raw.githubusercontent.com/lightningnetwork/lnd/master/sample-lnd.conf
```

```
mv sample-lnd.conf-lnd.conf
```

```
nano-lnd.conf e modificare le righe di seguito indicate
```

```
tlsextraip=0.0.0.0  
tlsextradomain=0.0.0.0
```

```
; Specify the interfaces to listen on for p2p connections. One listen  
; address per line.  
; All ipv4 on port 9735:  
listen=0.0.0.0:9735
```

```
; Specify the interfaces to listen on for gRPC connections. One listen  
; address per line.  
; Only ipv4 localhost on port 10009:  
rpclisten=localhost:10009
```

```
; Specify the interfaces to listen on for REST connections. One listen  
; address per line.  
; All ipv4 interfaces on port 8080:
```

restlisten=0.0.0.0:8080

; The alias your node will use, which can be up to 32 UTF-8 characters in  
; length.

alias=nome\_a\_piacere

:[Bitcoin] **IMPORTANTE IL PUNTO E VIRGOLA!!!**

; If the Bitcoin chain should be active. Atm, only a single chain can be  
; active.

bitcoin.active=1 **IMPORTANTE METTERE 1!!!**

; Use Bitcoin's main network.

bitcoin.mainnet=1 **IMPORTANTE METTERE 1!!!**

; Use Bitcoin's simulation test network

; bitcoin.simnet=true

; NOTE: Please note that switching between a full back-end (btcd/bitcoind) and  
; a light back-end (neutrino) is not supported.

; bitcoin.node=btcd

bitcoin.node=bitcoind

; bitcoin.node=neutrino

; Username for RPC connections to bitcoind. By default, lnd will attempt to  
; automatically obtain the credentials, so this likely won't need to be set  
; (other than for a remote bitcoind instance).

bitcoind.rpcuser= **bitcoin**

; Password for RPC connections to bitcoind. By default, lnd will attempt to  
; automatically obtain the credentials, so this likely won't need to be set  
; (other than for a remote bitcoind instance).

bitcoind.rpcpass= **bitcoin**

; ZMQ socket which sends rawblock and rawtx notifications from bitcoind. By  
; default, lnd will attempt to automatically obtain this information, so this  
; likely won't need to be set (other than for a remote bitcoind instance).

bitcoind.zmqpubrawblock=tcp://127.0.0.1:28332

bitcoind.zmqpubrawtx=tcp://127.0.0.1:28333

; bitcoind.zmqreaddeadline=10s

Salvare e uscire (Ctrl+X—Y--Invio)

lnd (verificare che lnd parta)

lncli create (da un altro terminal)

Normalmente scegliere “n” e non cifrare il seed

Trascrivere il seed

cd .lnd

nano password.txt (inserire la password inserita in fase di creazione del wallet)

nano lnd.conf e modificare le righe di seguito indicate

```
; The full path to a file (or pipe/device) that contains the password for
; unlocking the wallet; if set, no unlocking through RPC is possible and lnd
; will exit if no wallet exists or the password is incorrect; if
; wallet-unlock-allow-create is also set then lnd will ignore this flag if no
; wallet exists and allow a wallet to be created through RPC.
wallet-unlock-password-file=/home/satoshi/.lnd/password.txt
```

```
; Don't fail with an error if wallet-unlock-password-file is set but no wallet
; exists yet. Not recommended for auto-provisioned or high-security systems
; because the wallet creation RPC is unauthenticated and an attacker could
; inject a seed while lnd is in that state.
wallet-unlock-allow-create=true
```

```
; Allow outbound and inbound connections to be routed through Tor
tor.active=true
```

```
; This option may not be used while direct connections are enabled, since direct
; connections compromise source IP privacy by default.
tor.streamisolation=true
```

```
; Automatically set up a v3 onion service to listen for inbound connections
tor.v3=true
```

Salvare e uscire (Ctrl+X--Y--Invio)

ctrl+c per stoppare LND nella finestra dove sta girando

lnd (per rilanciarlo)

cd /etc/systemd/system (in altra finestra)

sudo wget https://raw.githubusercontent.com/lightningnetwork/lnd/master/contrib/init/lnd.service

stoppare lnd nell'altra finestra come sopra

whereis lnd per verificare il percorso

sudo nano lnd.service e modificare le righe di seguito indicate

```
# Replace these with the user:group that will run lnd
```

```
User= satoshi
```

```
Group= satoshi
```

Salvare e uscire (Ctrl+X--Y--Invio)

sudo systemctl enable lnd.service

journalctl -fu lnd.service (in una finestra)

sudo systemctl start Ind.service (in un'altra finestra)

sudo systemctl status Ind.service (verifica che Ind stia girando correttamente)

-----

## **Installazione di Ride The Lightning (RTL)**

Siti di riferimento:

<https://github.com/ride-the-lightning/rtl>

<https://nodejs.org/en/download/>

```
curl -fsSL https://deb.nodesource.com/setup_19.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs
```

```
mkdir ~/.npm-global
```

```
npm config set prefix '~/.npm-global'
```

```
nano .profile e aggiungere in fondo la seguente riga  
export PATH=~/.npm-global/bin:$PATH
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
source .profile
```

```
cd downloads
```

```
wget https://github.com/Ride-The-Lightning/RTL/archive/refs/tags/v0.12.3.tar.gz
```

```
tar -xvf v0.12.3.tar.gz
```

```
mv RTL-0.12.3/ ~
```

```
cd ..
```

```
mv RTL-0.12.3/ RTL
```

```
cd RTL
```

```
npm install --only=prod --legacy-peer-deps (ASPETTARE!)
```

In presenza di errori potrebbe apparire il messaggio seguente, procedere “a senso”.

To address issues that do not require attention, run:

```
npm audit fix
```

To address all issues (including breaking changes), run:

```
npm audit fix --force
```

Run `npm audit` for details.

```
npm notice
```

```
npm notice New patch version of npm available! 8.19.2 -> 8.19.3
```

```
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.19.3
```

```
npm notice Run npm install -g npm@8.19.3 to update!
```

```
cp Sample-RTL-Config.json RTL-Config.json
```

nano RTL-Config.json e modificare le righe di seguito indicate

```
"multiPass": "password_a_piacere",
```

```
"lnNode": "Node 2",
  "lnImplementation": "LND",
  "Authentication": {
    "macaroonPath": "/home/satoshi/.lnd/data/chain/bitcoin/mainnet",
    "configPath": "/home/satoshi/.lnd/lnd.conf",
    "swapMacaroonPath": "",
    "boltzMacaroonPath": ""
```

```
"channelBackupPath": "/home/satoshi/lndbackup",
(la directory lndbackup va creata)
```

Salvare e uscire (Ctrl+X—Y--Invio)

```
cd RTL
```

```
node rtl (lancia RTL)
```

```
indirizzo_ip_nodo:3000 (per verificare che l'interfaccia grafica funzioni)
```

Stappare RTL (Ctrl+C)

sudo nano /etc/systemd/system/RTL.service e copiare il seguente contenuto

```
[Unit]
Description=RTL daemon
Wants=ln.service
After=ln.service
```

```
[Service]
ExecStart=/usr/bin/node /home/satoshi/RTL/rtl
User=satoshi
Restart=always
TimeoutSec=120
RestartSec=30
```

```
[Install]
WantedBy=multi-user.target
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
sudo systemctl enable RTL.service (abilita l'autostart al reboot)
```

```
sudo systemctl start RTL.service (avvia il servizio)
```

```
sudo systemctl status RTL.service (verifica il corretto avvio del servizio)
```

**N.B.: Per raggiungere RTL via TOR procedere come segue:**

Aggiungere le seguenti tre linee nella sezione "location-hidden services" nel file /etc/tor/torrc.

```
sudo nano /etc/tor/torrc
```



```
##### This section is just for location-hidden services ###  
# Hidden Service RTL  
HiddenServiceDir /var/lib/tor/hidden_service_RTL/  
HiddenServiceVersion 3  
HiddenServicePort 80 127.0.0.1:3000 (può cambiare a seconda dell'installazione di RTL)
```

Salvare e uscire (Ctrl+X--Y--Invio)

Riavviare TOR e recuperare l'indirizzo .onion come segue

```
sudo systemctl reload tor
```

```
sudo cat /var/lib/tor/hidden_service_RTL/hostname
```

Deve ritornare qualcosa tipo abcdefg.....xyz.onion

---

## **Installazione di Lightning Terminal, Loop e Boltz**

Siti di riferimento:

<https://github.com/lightninglabs/lightning-terminal/releases>

<https://github.com/boltzexchange/boltz-lnd>

<https://lnd.docs.boltz.exchange/en/latest/>  
<https://boltz.exchange/>

cd downloads

wget <https://github.com/lightninglabs/lightning-terminal/releases/download/v0.8.3-alpha/lightning-terminal-linux-amd64-v0.8.3-alpha.tar.gz>

tar -xvf lightning-terminal-linux-amd64-v0.8.3-alpha.tar.gz

cd lightning-terminal-linux-amd64-v0.8.3-alpha/

sudo cp \* /usr/local/bin

whereis litd (verifica l'esatto percorso di LiT)

litd --uipassword=password\_a\_piacere (verifica che LiT si avvii correttamente)

Stoppare LiT (Ctrl+C)

cd ~

cd .lit

litd --uipassword=password\_inserita\_sopra --httpslisten=0.0.0.0:8443

[https://indirizzo\\_ip\\_nodo.8443](https://indirizzo_ip_nodo.8443) (per verificare che l'interfaccia grafica funzioni)

Stoppare LiT (Ctrl+C)

sudo nano /etc/systemd/system/litd.service e inserire quanto segue

[Unit]

Description=Lightning Loop daemon

Wants=lightningd.service

After=lightningd.service

[Service]

ExecStart=/usr/local/bin/litd --uipassword=password\_a\_piacere --httpslisten 0.0.0.0:8443

User= **satoshi**

Restart=always

TimeoutSec=120

RestartSec=30

StandardError=journal

[Install]

WantedBy=multi-user.target

Salvare e uscire (Ctrl+X--Y--Invio)

sudo systemctl enable litd.service

```
sudo systemctl start lited.service
```

```
cd ~
```

```
cd RTL
```

nano RTL-Config.json ed editare le righe seguenti come indicato

```
"swapMacaroonPath": "/home/satoshi/.loop/mainnet",
```

```
"swapServerUrl": "https://localhost:8443",
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
sudo systemctl restart RTL.service
```

**---BOLTZ---**

```
cd downloads
```

```
wget https://github.com/BoltzExchange/boltz-lnd/releases/download/v1.2.6/boltz-lnd-linux-amd64.zip
```

```
sudo apt install unzip
```

```
unzip boltz-lnd-linux-amd64.zip
```

```
sudo cp boltzcli boltzd /usr/local/bin
```

```
cd /usr/local/bin
```

```
sudo chmod +x boltzcli boltzd
```

```
cd ~
```

boltzd (darà un errore)

```
cd .boltz-lnd/
```

nano boltz.toml e inserire quanto segue

```
[LND]
```

```
macaroon = "/home/satoshi/.lnd/data/chain/bitcoin/mainnet/admin.macaroon"
```

```
certificate = "/home/satoshi/.lnd/tls.cert"
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
boltzd
```

```
cd ~ (In altra finestra)
```

```
cd RTL
```

nano RTL-Config.json ed editare le righe seguenti come indicato

```
"boltzMacaroonPath": "/home/satoshi/.boltz-lnd/macaroons"
```

Salvare e uscire (Ctrl+X--Y--Invio)

Stappare boltz (CTRL+C)

```
boltzd
```

sudo systemctl restart RTL (in altra finestra)

indirizzo\_ip\_nodo:3000 (per verificare che boltz funzioni)

Stappare boltz (CTRL+C)

sudo nano /etc/systemd/system/boltzd.service e inserire le righe seguenti

```
[Unit]
Description=Boltz daemon
Wants=lightningd.service
After=lightningd.service
```

```
[Service]
ExecStart=/usr/local/bin/boltzd
User= satoshi
Restart=always
TimeoutSec=120
RestartSec=30
StandardError=journal
```

```
[Install]
WantedBy=multi-user.target
```

Salvare e uscire (Ctrl+X--Y--Invio)

sudo systemctl enable boltzd.service

sudo systemctl start boltzd.service

indirizzo\_ip\_nodo:3000 (per verificare che boltz funzioni)

**N.B.: Per raggiungere Lightning Terminal via TOR procedere come segue:**

Aggiungere le seguenti tre linee nella sezione “location-hidden services” nel file /etc/tor/torrc.

```
sudo nano /etc/tor/torrc
```

```
##### This section is just for location-hidden services ###
# Hidden Service Lightning Terminal
HiddenServiceDir /var/lib/tor/hidden_service_LiT/
HiddenServiceVersion 3
HiddenServicePort 443 127.0.0.1:8443 (può cambiare a seconda dell’installazione di RTL)
```

Salvare e uscire (Ctrl+X--Y--Invio)

Riavviare TOR e recuperare l'indirizzo .onion come segue

```
sudo systemctl reload tor
```

```
sudo cat /var/lib/tor/hidden_service_LiT/hostname
```

Deve ritornare qualcosa tipo abcdefg.....xyz.onion



## **Installazione di Thunderhub**

Siti di riferimento:

<https://thunderhub.io/>

<https://github.com/apotdevin/thunderhub/releases/tag/v0.13.16>

<http://docs.thunderhub.io/>  
<https://docs.thunderhub.io/setup/#server-accounts>

cd ~

git clone <https://github.com/apotdevin/thunderhub.git>

cd thunderhub

npm install

In presenza di errori nell'installazione digitare

npm install --legacy-peer-deps e/o npm audit fix

npm run build

cp .env .env.local

nano .env.local e aggiungere in fondo al file

PORT=4000

Salvare e uscire (Ctrl+X--Y--Invio)

npm start

indirizzo\_ip\_nodo:4000 (per verificare che Thunderhub funzioni)

Stappare il servizio con CTRL+C

cd ~

mkdir .thunderhub

cd .thunderhub

nano thubConfig.yaml e inserire quanto segue

```
masterPassword: "password_a_piacere" # Default password unless defined in account
accounts:
```

```
- name: "Account 1"
```

```
  serverUrl: "localhost:10009"
```

```
  macaroonPath: "/home/satoshi/.lnd/data/chain/bitcoin/mainnet/admin.macaroon"
```

```
  certificatePath: "/home/satoshi/.lnd/tls.cert"
```

Salvare e uscire (Ctrl+X—Y--Invio)

cd ~

cd thunderhub

nano .env.local e inserire quanto seguente nella sezione indicata

```
# -----  
# Account Configs  
# -----  
ACCOUNT_CONFIG_PATH='/home/satoshi/.thunderhub/thubConfig.yaml'
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
npm start
```

indirizzo\_ip\_nodo:4000 (per verificare che Thunderhub funzioni)

Stappare il servizio con CTRL+C

sudo nano /etc/systemd/system/thunderhub.service e inserire quanto segue

```
[Unit]  
Description=Thunderhub  
Wants=Ind.service  
After=Ind.service  
  
[Service]  
WorkingDirectory=/home/satoshi/thunderhub  
ExecStart=/usr/bin/npm start  
User= satoshi  
Group= satoshi  
TimeoutSec=120  
Restart=always  
RestartSec=60
```

```
[Install]  
WantedBy=multi-user.target
```

Salvare e uscire (Ctrl+X--Y--Invio)

```
sudo systemctl enable thunderhub.service
```

```
sudo systemctl start thunderhub.service
```

**N.B: Per aggiornare periodicamente Thunderhub procedere come segue:**

```
cd thunderhub  
git pull  
npm install  
npm run build
```

**N.B.2: Per raggiungere Thunderhub via TOR procedere come segue:**

Aggiungere le seguenti tre linee nella sezione “location-hidden services” nel file /etc/tor/torrc.

```
sudo nano /etc/tor/torrc
```

```
##### This section is just for location-hidden services ###
```

```
# Hidden Service Thunderhub
HiddenServiceDir /var/lib/tor/hidden_service_thunderhub/
HiddenServiceVersion 3
HiddenServicePort 80 127.0.0.1:4000 (può cambiare a seconda dell'installazione di Thunderhub)
```

Salvare e uscire (Ctrl+X--Y--Invio)

Riavviare TOR e recuperare l'indirizzo .onion come segue

```
sudo systemctl reload tor
```

```
sudo cat /var/lib/tor/hidden_service_thunderhub/hostname
```

Deve ritornare qualcosa tipo abcdefg.....xyz.onion

-----

## **Installazione di Mempool**

Siti di riferimento:

<https://mempool.space/it/>

<https://github.com/mempool/mempool>

<https://docs.docker.com/engine/install/ubuntu/>



cd ~

sudo apt-get update

sudo apt-get install \  
ca-certificates \  
curl \  
gnupg \  
lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg|sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg

echo \  
"deb [arch=\$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
\$(lsb\_release -cs) stable"|sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

sudo usermod -aG docker \$USER (evita di usare "sudo")

Chiudere terminal e rientrare

sudo curl -SL https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-  
linux-x86\_64 -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

git clone https://github.com/mempool/mempool.git

cd mempool/docker

nano docker-compose.yml e sostituire il contenuto con quello seguente

version: "3.7"

services:

web:

environment:

FRONTEND\_HTTP\_PORT: "8080"

BACKEND\_MAINNET\_HTTP\_HOST: "api"

image: mempool/frontend:latest

user: "1000:1000"

restart: always

stop\_grace\_period: 1m

command: "./wait-for db:3306 --timeout=720 -- nginx -g 'daemon off;'"

ports:

- 4080:8080

api:

environment:

ELECTRUM\_HOST: "indirizzo\_ip\_nodo"

ELECTRUM\_PORT: "50002"

ELECTRUM\_TLS\_ENABLED: "true"

MEMPOOL\_BACKEND: "electrum"

CORE\_RPC\_HOST: "indirizzo\_ip\_nodo"

CORE\_RPC\_PORT: "8332"

CORE\_RPC\_USERNAME: "**bitcoin**"

CORE\_RPC\_PASSWORD: "**bitcoin**"

DATABASE\_ENABLED: "true"

DATABASE\_HOST: "db"

DATABASE\_DATABASE: "mempool"

DATABASE\_USERNAME: "mempool"

DATABASE\_PASSWORD: "mempool"

STATISTICS\_ENABLED: "true"

image: mempool/backend:latest

user: "1000:1000"

restart: always

stop\_grace\_period: 1m

command: "./wait-for-it.sh db:3306 --timeout=720 --strict -- ./start.sh"

volumes:

- ./data:/backend/cache

db:

environment:

MYSQL\_DATABASE: "mempool"

MYSQL\_USER: "mempool"

MYSQL\_PASSWORD: "mempool"

MYSQL\_ROOT\_PASSWORD: "admin"

image: mariadb:10.5.8

user: "1000:1000"

restart: always

stop\_grace\_period: 1m

volumes:

- ./mysql/data:/var/lib/mysql

networks:

default:

driver: bridge

ipam:

config:

- subnet: 172.16.57.0/24 (variare in presenza di più nodi)

Salvare e uscire (Ctrl+X--Y--Invio)

docker-compose up (aspettare che il servizio parta completamente)

indirizzo\_ip\_nodo:4080 (per verificare che Mempool funzioni)

Stoppare Mempool (ctrl+c)

docker-compose up -d (fa girare Mempool in background)

indirizzo\_ip\_nodo:4080 (per verificare che Mempool funzioni)

**N.B.: Per raggiungere Mempool via TOR procedere come segue:**

Aggiungere le seguenti tre linee nella sezione “location-hidden services” nel file /etc/tor/torrc.

```
sudo nano /etc/tor/torrc
```

```
##### This section is just for location-hidden services ###
```

```
# Hidden Service Mempool
```

```
HiddenServiceDir /var/lib/tor/hidden_service_mempool/
```

```
HiddenServiceVersion 3
```

```
HiddenServicePort 80 127.0.0.1:4080 (può cambiare a seconda dell'installazione di Thunderhub)
```

Salvare e uscire (Ctrl+X--Y--Invio)

Riavviare TOR e recuperare l'indirizzo .onion come segue

```
sudo systemctl reload tor
```

```
sudo cat /var/lib/tor/hidden_service_mempool/hostname
```

Deve ritornare qualcosa tipo abcdefg.....xyz.onion

---