



SatoshiSpritz Treviso
11 Febbraio 2025

Electrum Server: Electrs

Installazione, indicizzazione e configurazione sotto rete Tor

Installazione Rust

Installiamo curl

Electrs è un electrum server basato sul linguaggio di programmazione Rust.

Per questo motivo è necessario installare tale linguaggio nel nostro dispositivo.

Prima di procedere con l'installazione di Rust, dobbiamo installare il comando curl.

Eseguiamo il comando **apt install curl**

```
root@SSTreviso:/home/bitfede# apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 674 kB of archives.
After this operation, 1,428 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bookworm/main arm64 libcurl4 arm64 7.88.1-10+deb12u8 [365 kB]
Get:2 http://deb.debian.org/debian bookworm/main arm64 curl arm64 7.88.1-10+deb12u8 [309 kB]
Fetched 674 kB in 1s (748 kB/s)
Selecting previously unselected package libcurl4:arm64.
(Reading database ... 29586 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.88.1-10+deb12u8_arm64.deb ...
Unpacking libcurl4:arm64 (7.88.1-10+deb12u8) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.88.1-10+deb12u8_arm64.deb ...
Unpacking curl (7.88.1-10+deb12u8) ...
Setting up libcurl4:arm64 (7.88.1-10+deb12u8) ...
Setting up curl (7.88.1-10+deb12u8) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+deb12u9) ...
```

Installiamo Rust

A questo punto possiamo procedere con l'installazione di Rust eseguendo

```
curl --proto '=https' --tlsv1.2 -sSf  
https://sh.rustup.rs | sh
```

Durante il processo di installazione, ci verrà richiesto di scegliere un'opzione di installazione.

Digitiamo **1** e premiamo **Invio** per procedere con l'installazione predefinita.

```
root@SSTreviso:/home/bitfedede# curl --proto '=https' --tlsv1.2 -sSf  
https://sh.rustup.rs | sh  
info: downloading installer  
  
Welcome to Rust!  
  
This will download and install the official compiler for the Rust  
programming language, and its package manager, Cargo.  
  
Rustup metadata and toolchains will be installed into the Rustup  
home directory, located at:  
  
    /root/.rustup  
  
This can be modified with the RUSTUP_HOME environment variable.  
  
The Cargo home directory is located at:  
  
    /root/.cargo  
  
This can be modified with the CARGO_HOME environment variable.  
  
The cargo, rustc, rustup and other commands will be added to  
Cargo's bin directory, located at:  
  
    /root/.cargo/bin  
  
This path will then be added to your PATH environment variable by  
modifying the profile files located at:  
  
    /root/.profile  
    /root/.bashrc  
  
You can uninstall at any time with rustup self uninstall and  
these changes will be reverted.  
  
Current installation options:  
  
    default host triple: aarch64-unknown-linux-gnu  
    default toolchain:  stable (default)  
    profile:             default  
    modify PATH variable: yes  
  
1) Proceed with standard installation (default - just press enter)  
2) Customize installation  
3) Cancel installation  
>1
```

Installazione completata

Al termine dell'installazione ci viene consigliato di riavviare la shell.

Ci basterà fare **logout** e rientrare (o tornare superuser con su) per utilizzare i comandi installati con Rust.

```
Rust is installed now. Great!
```

```
To get started you may need to restart your current shell.  
This would reload your PATH environment variable to include  
Cargo's bin directory ($HOME/.cargo/bin).
```

```
To configure your current shell, you need to source  
the corresponding env file under $HOME/.cargo.
```

```
This is usually done by running one of the following (note the leading DOT):  
. "$HOME/.cargo/env" # For sh/bash/zsh/ash/dash/pdksh  
source "$HOME/.cargo/env.fish" # For fish
```

Installiamo clang e build-essential

Installiamo anche clang e il pacchetto build-essential eseguendo il seguente comando

```
apt install clang cmake build-essential -y
```



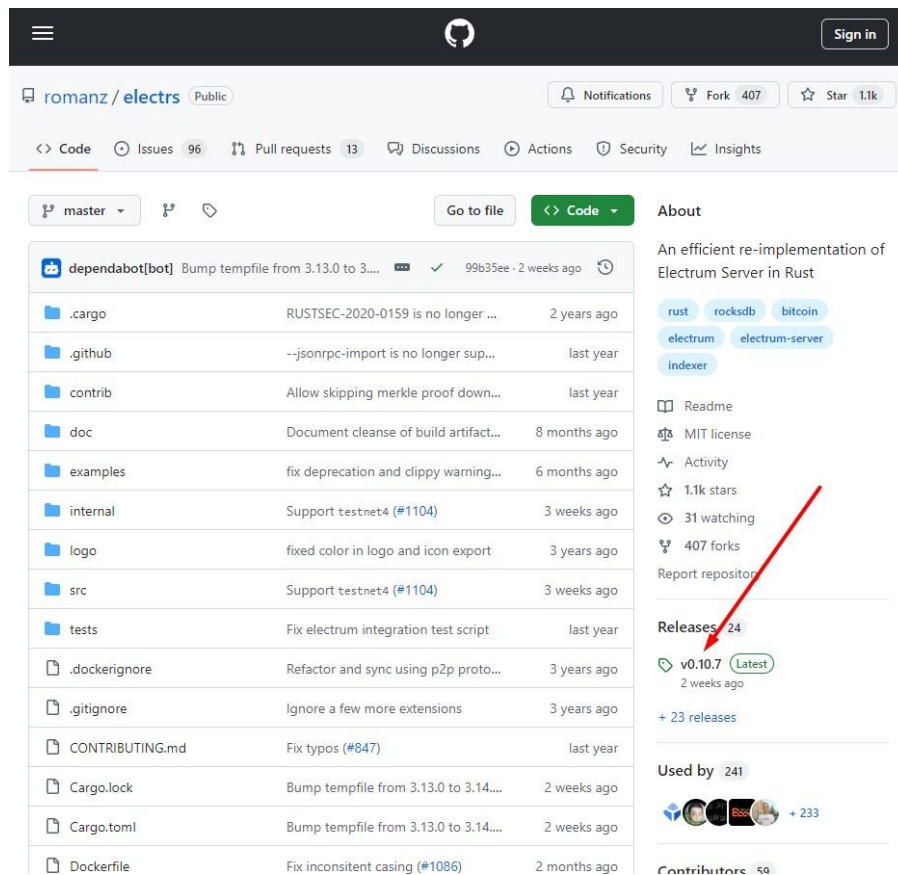
Download di Electrs

Versione di Electrs

Verifichiamo l'ultima versione di Electrs su

<https://github.com/romanz/electrs>

In questo caso è la 0.10.7



The screenshot shows the GitHub repository page for `romanz/electrs`. The repository is public and has 407 forks and 1.1k stars. The main content area displays a list of files and folders with their commit messages and dates. A red arrow points to the 'Releases' section on the right, which shows the latest release 'v0.10.7' from 2 weeks ago.

File/Folder	Commit Message	Time
<code>.cargo</code>	RUSTSEC-2020-0159 is no longer ...	2 years ago
<code>.github</code>	--jsonrpc-import is no longer sup...	last year
<code>contrib</code>	Allow skipping merkle proof down...	last year
<code>doc</code>	Document cleanse of build artifact...	8 months ago
<code>examples</code>	fix deprecation and clippy warning...	6 months ago
<code>internal</code>	Support testnet4 (#1104)	3 weeks ago
<code>logo</code>	fixed color in logo and icon export	3 years ago
<code>src</code>	Support testnet4 (#1104)	3 weeks ago
<code>tests</code>	Fix electrum integration test script	last year
<code>.dockerignore</code>	Refactor and sync using p2p proto...	3 years ago
<code>.gitignore</code>	Ignore a few more extensions	3 years ago
<code>CONTRIBUTING.md</code>	Fix typos (#847)	last year
<code>Cargo.lock</code>	Bump tempfile from 3.13.0 to 3.14...	2 weeks ago
<code>Cargo.toml</code>	Bump tempfile from 3.13.0 to 3.14...	2 weeks ago
<code>Dockerfile</code>	Fix inconsitent casing (#1086)	2 months ago

Releases 24

- v0.10.7 (Latest) 2 weeks ago
- + 23 releases

Used by 241

Contributors 59

Scarichiamo Electrs

Creiamo una variabile nel nostro sistema eseguendo `VERSION="0.10.7"` (ovviamente se l'ultima versione è 0.10.9, il valore tra virgolette è da cambiare).

Scarichiamo l'ultima versione di Electrs indicando proprio con la variabile: `git clone --branch v$VERSION https://github.com/romanz/electrs.git`

```
root@SSTreviso:/home/bitfede# VERSION="0.10.7"
root@SSTreviso:/home/bitfede# echo $VERSION
0.10.7
root@SSTreviso:/home/bitfede# git clone --branch v$VERSION https://github.com/romanz/electrs.git
fatal: destination path 'electrs' already exists and is not an empty directory.
root@SSTreviso:/home/bitfede# rmdir -r electrs/
rmdir: invalid option -- 'r'
Try 'rmdir --help' for more information.
root@SSTreviso:/home/bitfede# rmdir electrs/
rmdir: failed to remove 'electrs/': Directory not empty
root@SSTreviso:/home/bitfede# rm -r electrs/
root@SSTreviso:/home/bitfede# echo $VERSION
0.10.7
root@SSTreviso:/home/bitfede# git clone --branch v$VERSION https://github.com/romanz/electrs.git
Cloning into 'electrs'...
remote: Enumerating objects: 8299, done.
remote: Counting objects: 100% (1324/1324), done.
remote: Compressing objects: 100% (161/161), done.
remote: Total 8299 (delta 1250), reused 1190 (delta 1162), pack-reused 6975 (from 1)
Receiving objects: 100% (8299/8299), 3.51 MiB | 1.17 MiB/s, done.
Resolving deltas: 100% (5586/5586), done.
Note: switching to '7de627e439b825789d05c846637c8bbb7522ab94'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false
```

Importiamo la chiave dello sviluppatore

Ci spostiamo nella cartella `electrs` appena creata con il comando `cd electrs`.

Importiamo la chiave dello sviluppatore eseguendo `curl`

`https://romanzey.de/pgp.txt | gpg --import`

```
root@SSTreviso:/home/bitfede# cd electrs/
root@SSTreviso:/home/bitfede/electrs# curl https://romanzey.de/pgp.txt | gpg --import
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 1255  100 1255    0     0   2579      0  --:--:-- --:--:-- --:--:-- 2587
gpg: key 87CAE5FA46917CBB: public key "Roman Zeyde <me@romanzey.de>" imported
gpg: Total number processed: 1
gpg:                imported: 1
```



Verifichiamo la firma dello sviluppatore

Eseguiamo `git verify-tag v$VERSION` per verificare la firma dello sviluppatore

La firma è valida se compare la scritta **gpg: Good signature from “...”**

```
root@SSTreviso:/home/bitfede/electrs# git verify-tag v$VERSION
gpg: Signature made Tue 05 Nov 2024 22:19:10 CET
gpg:      using ECDSA key 15C8C3574AE4F1E25F3F35C587CAE5FA46917CBB
gpg:      issuer "me@romanzey.de"
gpg: Good signature from "Roman Zeyde <me@romanzey.de>" [unknown] ←
gpg:      aka "Roman Zeyde <roman.zeyde@gmail.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:      There is no indication that the signature belongs to the owner.
Primary key fingerprint: 15C8 C357 4AE4 F1E2 5F3F 35C5 87CA E5FA 4691 7CBB
root@SSTreviso:/home/bitfede/electrs#
```

Installazione di Electrs

Avviamo la compilazione

Sempre all'interno della cartella `electrs` eseguiamo `cargo build --locked --release` per avviare la compilazione.

La compilazione richiede parecchio tempo, perciò *armarsi* di pazienza :D

```
root@SSTreviso:/home/bitfede/electrs# cargo build --locked --release
```

Installiamo Electrs

Eseguiamo `install -m 0755 -o root -g root -t /usr/local/bin ./target/release/electrs` per installare Electrs.

```
root@SSTreviso:/home/bitfede/electrs# install -m 0755 -o root -g root -t /usr/local/bin ./target/release/electrs
```

Possiamo verificare l'avvenuta installazione controllando la versione eseguendo `electrs --version`

```
root@SSTreviso:/home/bitfede/electrs# electrs --version
v0.10.7
```

```
root@SSTreviso:/home/bitfede/electrs#
```

Impostazioni bitcoin.conf

Arrestiamo Bitcoin core

Eseguiamo `bitcoin-cli stop` per arrestare Bitcoin core.

```
root@SSTreviso:/home/bitfede/electrs# bitcoin-cli stop  
Bitcoin Core stopping
```



Modifichiamo il file `bitcoin.conf`

Eseguiamo `nano ~/.bitcoin/bitcoin.conf` per aprire il file `bitcoin.conf`

Aggiungiamo le seguenti righe:

```
server=1
```

```
rpcuser=USER
```

```
rpcpassword=PASSWORD
```

```
rpcallowip=127.0.0.1
```

```
whitelist=download@127.0.0.1
```

Premiamo **CTRL+x** per uscire, poi **y** per salvare e **invio** per sovrascrivere il file.

Spiegazione

server=1

Abilita Bitcoin Core per funzionare come un server RPC.

rpcuser=USER

Il nome utente (lo scegli tu) utilizzato per autenticare le connessioni RPC al nodo Bitcoin Core. Quando un'applicazione si connette al nodo tramite RPC, deve fornire questo nome utente.

rpcpassword=PASSWORD

La password (la scegli tu) associata al nome utente RPC. Serve per autenticare la connessione RPC.

rpcallowip=127.0.0.1

Specifica quali indirizzi IP possono accedere all'interfaccia RPC del nodo Bitcoin Core. In questo caso (127.0.0.1) solo i programmi eseguiti nella stessa macchina del nodo.

Spiegazione

`whitelist=download@127.0.0.1`

Questo specifica che i programmi provenienti dall'indirizzo IP 127.0.0.1 (locale, stessa macchina) possono scaricare blocchi e transazioni senza restrizioni, anche se altre regole del nodo potrebbero limitarlo (es: `maxuploadtarget=500`).

Sostanzialmente consente a Electrs di accedere liberamente ai dati del nodo.

Riavviamo Bitcoin core

Eseguiamo `bitcoind -daemon` per riavviare Bitcoin core.

```
root@SSTreviso:/home/bitfede/electrs# bitcoind -daemon  
Bitcoin Core starting
```



Impostazioni config.toml

Creiamo il file di configurazione

Assicuriamoci di essere dentro alla cartella dove è stato installato Electrs ed eseguiamo `nano config.toml` per creare il file di configurazione.

```
root@SSTreviso:/home/bitfedede/electrs# nano config.toml
```



Modifichiamo il file config.toml

Aggiungiamo le seguenti righe:

```
# bitcoin core configuration
auth = "USER:PASSWORD"
daemon_rpc_addr = "127.0.0.1:8332"
daemon_p2p_addr = "127.0.0.1:8333"

# electrs configuration
db_dir = ".electrum"
network = "bitcoin"
electrum_rpc_addr = "127.0.0.1:50001"
log_filters = "INFO"
```

Premiamo **CTRL+x** per uscire, poi **y** per salvare e **invio** per sovrascrivere il file.

Spiegazione configurazione di Bitcoin Core

auth = "USER:PASSWORD"

Sostituiamo USER e PASSWORD con le stesse credenziali impostate nel file bitcoin.conf del nodo per poterci autenticare tramite RPC.

daemon_rpc_addr = "127.0.0.1:8332"

Specifica l'indirizzo e la porta del servizio RPC del nodo. 127.0.0.1 indica che il nodo RPC è sulla stessa macchina, mentre 8332 è la porta predefinita per il servizio RPC di Bitcoin Core.

daemon_p2p_addr = "127.0.0.1:8333"

Specifica l'indirizzo e la porta per la connessione P2P al nodo. 127.0.0.1 indica che il servizio P2P è sulla stessa macchina, mentre 8333 è la porta predefinita per la comunicazione P2P di Bitcoin Core

Spiegazione configurazione di Electrs

db_dir = ".electrum"

Specifica la cartella in cui Electrs salva il suo database locale.

network = "bitcoin"

Specifica la rete su cui Electrs opera. In questo caso è la rete principale di Bitcoin.

electrum_rpc_addr = "127.0.0.1:50001"

Specifica l'indirizzo e la porta su cui Electrs offre il servizio RPC compatibile con i client Electrum. 127.0.0.1 indica che il servizio è accessibile solo dalla stessa macchina, mentre 50001 è la porta predefinita per connessioni non crittografate di Electrum.

log_filters = "INFO"

Imposta il livello di dettaglio dei log prodotti da Electrs.

Avvio indicizzazione Electrs

Avviamo l'indicizzazione

Assicuriamoci di essere dentro alla cartella dove è stato installato Electrs ed eseguiamo `electrs --conf config.toml` per avviare l'indicizzazione dei blocchi.

In questo esempio l'indicizzazione è stata avviata il 5 dicembre alle 6 e 21.

```
root@SSTreviso:/home/bitfede/electrs# electrs --conf config.toml
Starting electrs 0.10.7 on aarch64 linux with Config { network: Bitcoin, db_path: ".electrum/bitcoin",
db_log_dir: None, daemon_auth: UserPass("bitfede", "<sensitive>"), daemon_rpc_addr: 127.0.0.1:8332, d
aemon_p2p_addr: 127.0.0.1:8333, electrum_rpc_addr: 127.0.0.1:50001, monitoring_addr: 127.0.0.1:4224, w
ait_duration: 10s, jsonrpc_timeout: 15s, index_batch_size: 10, index_lookup_limit: None, reindex_last
blocks: 0, auto_reindex: true, ignore_mempool: false, sync_once: false, skip_block_download_wait: fals
e, disable_electrum_rpc: false, server_banner: "Welcome to electrs 0.10.7 (Electrum Rust Server)!", si
gnet_magic: f9beb4d9 }
[2024-12-05T06:21:03.810Z INFO electrs::metrics::metrics_impl] serving Prometheus metrics on 127.0.0.
1:4224
[2024-12-05T06:21:03.810Z INFO electrs::server] serving Electrum RPC on 127.0.0.1:50001
[2024-12-05T06:21:03.857Z INFO electrs::db] ".electrum/bitcoin": 23 SST files, 0.002067142 GB, 0.0000
59785 Grows
[2024-12-05T06:21:03.969Z INFO electrs::chain] loading 19480 headers, tip=00000000e24cbdf59f363cd520a
675f0a4a19f9abc3b812b0c13d31882b9c4e6
[2024-12-05T06:21:04.076Z INFO electrs::chain] chain updated: tip=00000000e24cbdf59f363cd520a675f0a4a
19f9abc3b812b0c13d31882b9c4e6, height=19480
[2024-12-05T06:21:04.094Z INFO electrs::index] indexing 2000 blocks: [19481..21480]
[2024-12-05T06:21:04.868Z INFO electrs::chain] chain updated: tip=0000000050b98a266aa0e0a5d2744430c4a
5d9096670f5c865614ee329a88987, height=21480
```

Attendiamo la completa indicizzazione

Lasciare il terminale aperto per permettere l'indicizzazione di tutta le rete.

Ci può impiegare molte ore.

```
[2024-12-05T18:44:53.826Z INFO electrs::chain] chain updated: tip=000000000000000000000000657a27fe539c038
b9bb135caa004fd025cff6e049334, height=873405
[2024-12-05T18:45:15.561Z INFO electrs::db] starting config compaction
[2024-12-05T18:45:15.573Z INFO electrs::db] starting headers compaction
[2024-12-05T18:45:21.627Z INFO electrs::db] starting txid compaction
[2024-12-05T19:55:20.763Z INFO electrs::db] starting funding compaction
[2024-12-05T22:38:33.396Z INFO electrs::db] starting spending compaction
[2024-12-06T01:24:40.580Z INFO electrs::db] finished full compaction
```

Processo completato il 6 dicembre all'1 e 24.

Dopo 19 ore e 3 minuti

Configurazione di Tor per Electrs

Modifichiamo il file di configurazione di Tor

Eseguiamo il comando `nano /etc/tor/torrc` per modificare il file di configurazione di Tor.

```
root@SSTreviso:/home/bitfedede# nano /etc/tor/torrc
```

Aggiungiamo le seguenti righe alla fine del file

```
HiddenServiceDir /var/lib/tor/electrs_hidden_service/  
HiddenServiceVersion 3  
HiddenServicePort 50001 127.0.0.1:50001
```

```
HiddenServiceDir /var/lib/tor/electrs_hidden_service/  
HiddenServiceVersion 3  
HiddenServicePort 50001 127.0.0.1:50001
```

Facciamo **CTRL+x** per uscire, **y** per salvare e **invio** per sovrascrivere.

Riavviamo tor

Riavviamo tor eseguendo il comando `systemctl restart tor`

```
root@SSTreviso:/home/bitfede# systemctl restart tor
```



Otteniamo indirizzo onion

Eseguiamo `cat /var/lib/tor/electrs_hidden_service/hostname` per vedere l'indirizzo onion associato a Electrs.

```
root@SSTreviso:/home/bitfede# cat /var/lib/tor/electrs_hidden_service/hostname
23h2k62rrt5f3ajiggsgp2smj3pp3qh3pw7nt2r7jpbjkiypxtdlmnid.onion
```

In questo caso l'indirizzo è:

23h2k62rrt5f3ajiggsgp2smj3pp3qh3pw7nt2r7jpbjkiypxtdlmnid.onion

Modifica dell'indirizzo onion per Electrs

Rimuoviamo la cartella bitcoin-service

Eseguiamo `rm -r /var/lib/tor/electrs_hidden_service/` per rimuovere la cartella `/electrs_hidden_service/` che contiene la chiave privata che genera l'indirizzo tor.

```
root@SSTreviso:/home/bitfede# rm -r /var/lib/tor/electrs_hidden_service/
```



Riavviamo Tor

Eseguiamo `systemctl restart tor` per riavviare Tor e rigenerare la cartella `/electrs_hidden_service/` così come la chiave pubblica e privata.

```
root@SSTreviso:/home/bitfede# systemctl restart tor
```



Verifichiamo il nuovo indirizzo onion

Eseguiamo `cat /var/lib/tor/electrs_hidden_service/hostname` per verificare il nuovo indirizzo onion.

```
root@SSTreviso:/home/bitfede# cat /var/lib/tor/electrs_hidden_service/hostname
po6venb23mmreemo7dzla3c7bxvevbtlg2kmkkrbwxyzwrt2ld7cv2yd.onion
```



Sitografia

-
- <https://linuxize.com/post/curl-command-examples/>
 - <https://rustup.rs/>
 - <https://github.com/romanz/electrs>
 - <https://github.com/romanz/electrs/blob/master/doc/config.md>
 - https://github.com/romanz/electrs/blob/master/doc/config_example.toml
 - <https://officinebitcoin.it/lezioni/fulhar/>
-