Il tuo server di casa

Guida per il server di casa · ottobre 2026 · vale per Agenda di Casa e per tutte le app che metterai sul server

Riquadri: verde = messaggio per l'assistente IA · rosso = ATTENZIONE · giallo = NON PROVATO (come leggerli)

Prima ti serve: un computer da tenere sempre acceso (anche vecchio), una chiavetta USB da almeno 4 GB, e il tuo computer personale (Ubuntu, Windows o Mac).

Dopo puoi: installare le app di casa (per esempio con la guida Agenda di Casa — con il tuo server) e la VPN (guida La VPN di casa con WireGuard).

Un **server di casa** è un computer che resta acceso giorno e notte e tiene i dati delle tue app: calendario, spese, documenti, foto... Invece di stare sui computer di un'azienda, i tuoi dati stanno in casa tua.

In questa guida prepari il server una volta per tutte: lo installi, impari a entrarci dal tuo computer, lo metti in sicurezza e lo prepari ad accogliere le app. Ogni app poi ha la sua guida, che parte da qui.

Se usi un assistente IA (per esempio un chatbot)

Copia e incolla questo messaggio insieme alla guida:

"Guidami a preparare il mio server di casa seguendo questa guida. Prima chiedimi che computer userò come server (marca e modello, o Raspberry Pi) e che sistema ho sul mio computer personale (Ubuntu, Windows o Mac). Poi guidami un passo alla volta: dimmi su quale computer devo fare ogni passo, aspetta che ti incolli quello che compare e confrontalo con 'Cosa deve succedere'. Non saltare i riquadri ATTENZIONE. Non chiedermi mai di scriverti password o chiavi."

Prima di cominciare

Che computer usare

Va bene quasi tutto, purché possa restare acceso:

- un **vecchio PC o portatile** (anche di 8–10 anni), oppure un **mini PC** ricondizionato: piccoli, silenziosi, consumano poco;
- un **Raspberry Pi 4 o 5** (con almeno 4 GB di memoria e un buon alimentatore).

Meglio collegarlo al router **con il cavo di rete**: è più stabile del Wi-Fi.

ATTENZIONE — tutto quello che c'è sul disco del server verrà cancellato

Installare Ubuntu Server cancella il disco del computer scelto. Se ci sono file che ti interessano, copiali altrove **prima**.

Due modi di fare il server

	Server dedicato (consigliato)	Un solo computer
Come funziona	il server ha **Ubuntu Server**, senza schermo né mouse; lo gestisci dal tuo computer personale	il tuo computer con **Ubuntu Desktop** fa anche da server
Pro	resta acceso solo per quello, consuma poco, niente programmi personali che lo rallentano o lo mettono a rischio	non serve un secondo computer
Contro	serve un secondo computer	deve restare sempre acceso, e lo usi anche per altro
Cosa fai in questa guida	tutte le parti	salta le parti 1 e 2, poi segui il riquadro Un solo computer

Questa guida segue il **server dedicato**. Per l'altro modo, guarda il riquadro Un solo computer in fondo.

Su quale computer si fa ogni passo

In questa guida lavori su **due computer**. Ogni passo te lo dice così:

- **[PC]** = il tuo computer personale (Ubuntu, Windows o Mac);
- **[SERVER]** = il server: all'inizio con tastiera e schermo collegati, poi dal tuo PC con SSH (parte 2).

Due cose da sapere sul terminale

Il **terminale** è la finestra dove si scrivono i comandi.

- **Ubuntu:** si apre con **Ctrl+Alt+T**. Si incolla con **Ctrl+Shift+V** o con **tasto destro → Incolla**.
- **Windows:** usa **PowerShell** (tasto Windows, scrivi `powershell`, Invio). Si incolla con il **tasto destro**.
- **Mac:** l'app si chiama **Terminale**. Si incolla con **Cmd+V**.
- Quando un comando chiede una **password**, mentre la scrivi **non compare niente**. È normale.
- Se un comando incollato compare con caratteri in più (per esempio `[200~` o `00~`), **non premere Invio**: cancella la riga con **Ctrl+U** e incolla con **tasto destro → Incolla**.
- **Copia i comandi dalla versione online** della guida, non dal PDF: nel PDF i comandi lunghi vanno a capo e arrivano spezzati.

Percorso rapido (per chi sa già come si fa)

1. Ubuntu Server LTS su chiavetta (Rufus, balenaEtcher o Creatore di dischi di avvio), installazione su tutto il disco, **OpenSSH server sì**, nessuno snap.
2. `ssh-keygen -t ed25519` sul PC, chiave pubblica in `~/.ssh/authorized_keys` del server, poi `/etc/ssh/sshd_config.d/01-sicurezza.conf` con `PasswordAuthentication no` e `KbdInteractiveAuthentication no` (prova da un secondo terminale prima di chiudere).
3. `sudo apt update && sudo apt full-upgrade -y`; `unattended-upgrades` è già attivo.
4. Prenotazione DHCP dell'indirizzo del server sul router.
5. `ufw`: SSH solo dalla rete di casa, poi `ufw enable`.
6. Docker: `sudo apt install -y docker.io docker-compose-v2`. Ricorda che le porte pubblicate da Docker scavalcano `ufw`.
7. BIOS: riaccensione automatica dopo un'interruzione di corrente.

Parte 1 — Installare Ubuntu Server

Se il tuo server è un Raspberry Pi, salta alla parte 1-bis.

1.1 — Scarica Ubuntu Server [PC]

Dal sito ufficiale **ubuntu.com/download/server** scarica l'ultima versione **LTS** (sta per Long Term Support: aggiornamenti di sicurezza per anni; è quella giusta per un server). È un file `.iso` di qualche GB.

1.2 — Prepara la chiavetta [PC]

La chiavetta (almeno 4 GB) **verrà cancellata**: controlla che non ci sia niente di importante.

- **Da Ubuntu:** apri **Creatore di dischi di avvio** (cercalo nelle applicazioni), scegli il file `.iso` e la chiavetta, poi **Crea disco di avvio**.
- **Da Windows:** scarica **Rufus** da rufus.ie (gratuito e libero). In Dispositivo scegli la chiavetta, in Selezione boot il file `.iso`, il resto lascialo com'è, poi **AVVIA**. Se chiede la modalità, lascia **Scrivi in modalità immagine ISO**.
- **Da Mac (o da qualsiasi sistema): balenaEtcher** da etcher.balena.io: scegli il file, la chiavetta, **Flash**.

Cosa deve succedere: il programma finisce senza errori (Rufus mostra PRONTO in verde).

1.3 — Avvia il server dalla chiavetta [SERVER]

1. Collega al server **tastiera, schermo, cavo di rete** e la chiavetta.
2. Accendilo e premi subito, più volte, il tasto del **menu di avvio**. Cambia da marca a marca: di solito **F12**, **F11**, **F10**, **F9** o **Esc** (spesso è scritto per un attimo sullo schermo all'accensione; altrimenti chiedi al tuo assistente IA con marca e modello).

3. Nel menu scegli la chiavetta (a volte compare come UEFI: nome della chiavetta).
4. Scegli **Try or Install Ubuntu Server**.

Cosa deve succedere: dopo un minuto compare la prima schermata dell'installazione (scelta della lingua). Si usa solo la **tastiera**: frecce per muoversi, **Invio** per scegliere, **Tab** per passare al pulsante successivo, **barra spaziatrice** per mettere o togliere una spunta.

1.4 — L'installazione, schermata per schermata [SERVER]

Schermata	Cosa scegliere
Lingua	Italiano va bene. (L'inglese rende più facile cercare i messaggi d'errore, ma non è obbligatorio.)
Tastiera (Keyboard)	**Italian** (Italiano), se la tua tastiera è italiana.
Tipo di installazione	**Ubuntu Server** (non la versione minimized).
Rete (Network)	Deve comparire un indirizzo tipo `192.168.1.x` accanto alla scheda di rete (preso **DHCP**). Lascia così: l'indirizzo fisso lo prenoterai dal router (parte 4). **Annota questo indirizzo.**
Proxy	Lascia vuoto.
Mirror	Lascia quello proposto.
Disco (Storage)	**Usa un disco intero** (Use an entire disk), con **LVM** spuntato. Per la cifratura del disco vedi il riquadro sotto.
Profilo (Profile)	Il tuo nome; il **nome del server** (per esempio `casaserver`, solo minuscole e trattini); il tuo **nome utente** (per esempio `mario`); una **password** lunga, da salvare nel gestore di password.
Ubuntu Pro	Salta (Skip). Non serve.
SSH	Metti la spunta su **Install OpenSSH server**. È fondamentale: serve per entrare dal tuo PC.
Snap in evidenza (Featured snaps)	**Non scegliere niente**, nemmeno docker: Docker lo installerai nel modo giusto nella parte 6.

Poi l'installazione parte. Alla fine scegli **Riavvia ora** (Reboot Now) e, quando te lo chiede, **togli la chiavetta** e premi Invio.

NON PROVATO passo per passo da chi ha scritto la guida

L'installazione è stata fatta dall'autore sul suo server, ma questa tabella non è stata ricontrollata schermata per schermata, e l'installazione di Ubuntu cambia un po' a ogni versione. Se una schermata non corrisponde, scegli l'opzione con il significato più vicino oppure segui il tutorial ufficiale Install Ubuntu Server (in inglese; un assistente IA può tradurtelo). Le due scelte da non sbagliare restano: **Install OpenSSH server** sì, **nessuno snap**. Poi segnala cosa non ha funzionato (e cosa ha funzionato) scrivendo al progetto (proactiveproject.eu/progetto): così la guida migliora per tutti.

Cifrare il disco: sì o no?

La **cifratura** protegge i dati se qualcuno **ruba** il server: senza la frase segreta il disco è illeggibile. Il prezzo: a ogni accensione il server si ferma e aspetta che qualcuno scriva la frase **con la tastiera collegata**. Per un server senza schermo, che dopo un'interruzione di corrente deve ripartire da solo, è scomodo.

Esistono soluzioni (lo sblocco automatico con il chip TPM del computer, o lo sblocco a distanza), ma sono passi avanzati. **Per iniziare: senza cifratura.** La protezione principale dei tuoi dati è che il server non è raggiungibile da internet.

Cosa deve succedere: dopo il riavvio compare una schermata nera con `casaserver login:` (con il nome che hai scelto). Entra con nome utente e password: compare una riga come `mario@casaserver:~$`. Il server è installato.

Scrivi:

```
hostname -I
```

e **annota l'indirizzo** (per esempio `192.168.1.10`): è quello con cui lo raggiungerai dal PC.

1.5 — Riaccensione automatica dopo un'interruzione di corrente [SERVER]

Se va via la corrente, il server deve riaccendersi da solo quando torna. È un'impostazione del **BIOS** (il programma che parte prima del sistema):

1. Riavvia il server (`sudo reboot`) e premi subito il tasto del **BIOS/Setup** (di solito **F10**, **F2**, **F1** o **Canc**).
2. Cerca una voce come **After Power Loss**, **AC Power Recovery** o **Restore on AC Power Loss** e mettila su **Power On** (oppure Last State).
3. Salva ed esci (di solito **F10**).

Le voci cambiano da computer a computer: chiedi al tuo assistente IA "dove trovo l'impostazione per riaccendere dopo un'interruzione di corrente nel BIOS del (marca e modello)". Nel BIOS non toccare altro.

Da qui in poi tastiera e schermo non servono più: puoi staccarli.

1-bis — Se il server è un Raspberry Pi [PC]

Sul Raspberry Pi non si usa la chiavetta: si prepara direttamente la sua schedina (microSD) dal PC.

1. Installa **Raspberry Pi Imager** dal sito ufficiale raspberrypi.com/software.
2. Scegli il modello di Raspberry Pi, poi come sistema **Other general-purpose OS → Ubuntu → Ubuntu Server LTS (64-bit)**, poi la microSD. (Le voci del programma possono cambiare un po' tra una versione e l'altra.)
3. Quando ti propone di **personalizzare le impostazioni**, scegli **Sì** e imposta: nome del server, nome utente e password, fuso orario Europe/Rome, tastiera it, e nella scheda dei servizi **attiva SSH** (con password, per ora).
4. Scrivi la microSD, mettila nel Raspberry Pi, collega il cavo di rete e l'alimentazione.
5. Dopo qualche minuto trova il suo indirizzo nell'elenco dei dispositivi del router (guida Il router di casa), oppure prova con il nome: `ssh mario@casaserver.local` (parte 2).

Il Raspberry Pi si riaccende da solo quando torna la corrente: la 1.5 non serve.

> **NON PROVATO da chi ha scritto la guida**
>
> Questa parte (Raspberry Pi) non è stata provata. Se qualcosa non torna, segui la guida ufficiale di Ubuntu: How to install Ubuntu Server on your Raspberry Pi (in inglese; un assistente IA può tradurtela). Poi segnala cosa non ha funzionato (e cosa ha funzionato) scrivendo al progetto (proactiveproject.eu/progetto): così la guida migliora per tutti.

Parte 2 — Entrare nel server dal tuo PC (SSH)

SSH è il modo sicuro per usare il terminale del server dal tuo computer personale, come se fossi seduto davanti al server.

2.1 — La prima volta, con la password [PC]

Apri il terminale del PC (Ubuntu, Mac) o PowerShell (Windows) e scrivi, con il tuo nome utente e l'indirizzo del server:

```
ssh mario@192.168.1.10
```

La prima volta chiede se ti fidi del server (Are you sure you want to continue connecting): scrivi `yes` e Invio. Poi la password del server.

Cosa deve succedere: compare `mario@casaserver:~$`. Ora stai scrivendo **sul server**. Per tornare al PC: `exit`.

ATTENZIONE — guarda sempre su quale computer stai scrivendo

La riga prima del cursore ti dice dove sei: `mario@casaserver` = server; il nome del tuo PC = PC. Molti errori nascono da un comando giusto lanciato sul computer sbagliato.

2.2 — Crea la tua chiave SSH [PC]

Una **chiave SSH** è come una chiave di casa fatta di due pezzi: la parte **privata** resta sul tuo PC (non va mai data a nessuno), la parte **pubblica** la copi sul server. È molto più sicura di una password, che si può indovinare.

Sul PC (Ubuntu, Mac o PowerShell di Windows):

```
ssh-keygen -t ed25519
```

- Alla domanda su dove salvarla premi **Invio** (va bene il posto proposto).
- Poi chiede una **passphrase**: una frase che protegge la chiave se qualcuno copia il tuo PC. Mettila (lunga, nel gestore di password) e ripetila.

Cosa deve succedere: compare Your identification has been saved e un disegno fatto di caratteri (la "randomart"). Nella cartella `.ssh` della tua cartella personale ci sono ora `id_ed25519` (privata) e `id_ed25519.pub` (pubblica).

2.3 — Copia la chiave pubblica sul server [PC]

Da Ubuntu o Mac:

```
ssh-copy-id mario@192.168.1.10
```

Da Windows (PowerShell), che non ha `ssh-copy-id`, tutto su una riga:

```
type $env:USERPROFILE\.ssh\id_ed25519.pub | ssh mario@192.168.1.10 "mkdir -p ~/.ssh &&
chmod 700 ~/.ssh && cat >> ~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys"
```

Ti chiede la **password del server** (un'ultima volta).

Poi prova:

```
ssh mario@192.168.1.10
```

Cosa deve succedere: ora chiede la **passphrase della chiave** (non la password del server) ed entri. Se chiede ancora la password del server, la chiave non è stata copiata: ripeti il comando di copia.

NON PROVATO da chi ha scritto la guida: la copia da Windows e da Mac

I comandi per Ubuntu sono provati; quelli per **Windows** e **Mac** no. Se non funzionano, c'è un **metodo a mano** che va bene ovunque:

1. **[PC]** Apri il file della chiave **pubblica** con un editor di testo: su Windows `C:\Users\tuo-nome\.ssh\id_ed25519.pub` con il **Blocco note**; su Mac, nel Terminale, `cat ~/.ssh/id_ed25519.pub`. È **una sola riga** che inizia con `ssh-ed25519`. Copiala tutta. (Il file **senza** `.pub` è la chiave privata: non aprirlo e non copiarlo mai.)
2. **[SERVER]** Entra con la password (`ssh mario@192.168.1.10`) e scrivi `mkdir -p ~/.ssh && chmod 700 ~/.ssh && nano ~/.ssh/authorized_keys`.
3. Incolla la riga (tasto destro), salva con **Ctrl+O**, **Invio**, esci con **Ctrl+X**, poi scrivi `chmod 600 ~/.ssh/authorized_keys`.
4. `exit` e riprova `ssh mario@192.168.1.10`: deve chiedere la passphrase della chiave.

Poi segnala cosa non ha funzionato (e cosa ha funzionato) scrivendo al progetto (proactiveproject.eu/progetto): così la guida migliora per tutti.

2.4 — Spegni l'accesso con password [SERVER]

Ora che la chiave funziona, chiudiamo la porta alle password: da quel momento si entra **solo** con una chiave.

ATTENZIONE — fallo solo dopo che la 2.3 funziona

Se spegni le password senza una chiave funzionante, non entri più da remoto (dovrai rimettere tastiera e schermo sul server). Per sicurezza, durante questo passo **tieni aperta** la sessione SSH già collegata.

Collegato al server, copia e incolla:

```
printf 'PasswordAuthentication no\nKbdInteractiveAuthentication no\n' | sudo tee /etc/ssh/sshd_config.d/01-sicurezza.conf && sudo sshd -t && sudo systemctl reload ssh
```

Cosa fa: scrive un piccolo file di regole per SSH (il nome inizia con `01-` così vale prima degli altri), controlla che non ci siano errori (`sshd -t`) e ricarica SSH.

Ora **apri un secondo terminale sul PC** (senza chiudere il primo) e prova a entrare di nuovo:

```
ssh mario@192.168.1.10
```

Cosa deve succedere: entri con la **passphrase della chiave**, come prima. Se funziona, puoi chiudere il primo terminale.

Prova del nove (facoltativa): da un computer che non ha la tua chiave, `ssh mario@192.168.1.10` deve rispondere `Permission denied (publickey)`, senza chiedere nessuna password.

Più computer? Ogni computer da cui vuoi entrare ha la **sua** chiave (2.2) da copiare sul server. Per farlo, riaccendi le password per un momento oppure aggiungi la chiave pubblica nuova in `~/.ssh/authorized_keys` da un computer che entra già.

Parte 3 — Aggiornamenti [SERVER]

Aggiorna tutto subito:

```
sudo apt update && sudo apt full-upgrade -y
```

Cosa deve succedere: alla fine torna la riga `mario@casaserver:~$` senza errori in rosso. Se compare System restart required, riavvia con `sudo reboot` e ricollegati dopo un minuto.

Ubuntu Server installa da solo gli **aggiornamenti di sicurezza** ogni giorno (si chiama `unattended-upgrades`). Controlla che sia attivo:

```
systemctl is-enabled unattended-upgrades
```

Cosa deve succedere: `enabled`.

Gli aggiornamenti più grandi (o quelli che richiedono un riavvio) falli tu, ogni tanto, con il comando di sopra.

Parte 4 — Un indirizzo che non cambia

Il server deve avere **sempre lo stesso indirizzo**, altrimenti un giorno le app non lo trovano più. Si fa sul router: guida Il router di casa, sezione **5.2** (prenotare l'indirizzo di un dispositivo).

Cosa deve succedere: dopo un riavvio del server (`sudo reboot`), `hostname -I` mostra sempre lo stesso indirizzo.

Parte 5 — Il firewall [SERVER]

Il **firewall** decide chi può parlare con il server. Ubuntu ne ha uno semplice, **ufw**. Lo impostiamo così: tutto chiuso, tranne SSH dalla rete di casa.

Prima controlla com'è fatta la tua rete di casa: l'indirizzo del server inizia con `192.168.`? Allora usa il comando così com'è. Se inizia con `10.` o `172.`, chiedi al tuo assistente IA come cambiare `192.168.0.0/16` nel comando.

```
sudo ufw default deny incoming && sudo ufw default allow outgoing && sudo ufw allow from 192.168.0.0/16 to any port 22 proto tcp && sudo ufw enable
```

Alla domanda Command may disrupt existing ssh connections. Proceed with operation? rispondi `y`: SSH dalla rete di casa è già permesso, quindi la tua connessione resta in piedi.

Controlla:

```
sudo ufw status
```

Cosa deve succedere: `Status: active` e una riga con `22/tcp ALLOW 192.168.0.0/16`.

Ogni app ti dirà se serve aprire una sua porta nel firewall, e solo verso la rete di casa.

Parte 6 — Docker [SERVER]

Docker fa girare le app dentro "scatole" separate (container). Molte guide lo usano (per esempio la VPN), quindi installalo adesso:

```
sudo apt install -y docker.io docker-compose-v2 && sudo docker compose version
```

Cosa deve succedere: alla fine `Docker Compose version v2…` (o più alto). Docker parte da solo a ogni accensione.

> **ATTENZIONE — Docker scavalca il firewall**
>
> Le porte che Docker apre per i suoi container sono raggiungibili **anche se ufw dice il contrario**. Non è un problema finché il **router** non le inoltra verso internet: **non farlo mai** (guida Il router di casa, sezione 5.3). Da fuori casa si passa solo dalla VPN.

Parte 7 — Le buone abitudini

Per…	Comando o azione
spegnere il server	`sudo shutdown now` — **mai** staccare la corrente
riavviarlo	`sudo reboot`
vedere quanto spazio resta sul disco	`df -h /`
vedere quanta memoria usa	`free -h`
vedere i container Docker accesi	`sudo docker ps`
fare le **copie** dei dati	ogni app spiega le sue copie nella sua guida; in più, tieni le copie **fuori dal server** (un disco esterno, un altro computer)

Le copie sono la tua responsabilità. Un server di casa senza copie è una scommessa: prima o poi un disco si rompe. Più avanti potrai automatizzare le copie notturne su un disco esterno (per esempio con il programma libero restic).

Un solo computer (Ubuntu Desktop che fa anche da server)

Se hai deciso di non avere un server dedicato:

- **salta le parti 1 e 2**: hai già Ubuntu Desktop (guida Installare Ubuntu Desktop) e lavori direttamente nel suo terminale;
- **fai le parti 3, 4, 5 e 6** uguali (nella 5 il firewall va bene così; se SSH non c'è, il comando funziona lo stesso);
- **impedisci che vada in sospensione**: Impostazioni → **Energia** → Sospensione automatica **disattivata** (anche "con alimentazione"). Un computer sospeso non risponde alle app;
- ricordati che **è anche il tuo computer**: se lo spegni la sera, le app della famiglia smettono di funzionare.

Windows, Mac e Ubuntu: le differenze che contano

Il server ha sempre Ubuntu. Cambia solo il **tuo** computer, quello da cui entri:

	Ubuntu	Windows	Mac
Terminale	Terminale (Ctrl+Alt+T)	PowerShell	Terminale
Incollare	Ctrl+Shift+V o tasto destro	tasto destro	Cmd+V
Creare la chiave	`ssh-keygen -t ed25519`	uguale	uguale
Copiare la chiave sul server	`ssh-copy-id`	comando lungo della 2.3	`ssh-copy-id`
Dove sta la chiave	`~/.ssh/`	`C:\Users\tuo-nome\.ssh\`	`~/.ssh/`
Chiavetta di installazione	Creatore di dischi di avvio	Rufus	balenaEtcher

Tutti i comandi che si scrivono **sul server** sono identici, qualunque sia il tuo computer.

Se qualcosa non va

Cosa succede	Cosa fare
Il computer non parte dalla chiavetta	Prova un altro tasto del menu di avvio; nel BIOS controlla che l'avvio da USB sia permesso. Rifai la chiavetta se il programma aveva dato errori.
Durante l'installazione la rete non prende un indirizzo	Controlla il cavo di rete e che il router sia acceso; prova un'altra presa del router.
`ssh: connect to host … port 22: No route to host` o Connection timed out	Sei sulla rete di casa? L'indirizzo è giusto (`hostname -I` sul server)? Il server è acceso?
`Permission denied (publickey)` dal tuo PC	La chiave del PC non è sul server: con la tastiera sul server riaccendi le password (cancella `/etc/ssh/sshd_config.d/01-sicurezza.conf` e `sudo systemctl reload ssh`), copia la chiave (2.3) e rispegnile (2.4).
`WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED`	Succede se hai reinstallato il server. Se sei sicuro che sia il tuo: sul PC `ssh-keygen -R 192.168.1.10` , poi riprova. Se non hai reinstallato niente, fermati e chiedi aiuto.
`sshd -t` mostra un errore	Il file della 2.4 è scritto male: ricopia il comando intero. Non ricaricare SSH finché `sshd -t` non risponde senza niente.
Dopo `ufw enable` non entri più	Il range nel comando non corrisponde alla tua rete. Con tastiera sul server: `sudo ufw disable` , correggi il comando, riattivalo.
Il server non si riaccende dopo un'interruzione di corrente	Impostazione del BIOS (1.5).

Questa guida descrive un server vero di casa (un mini PC con Ubuntu Server). Se trovi un errore o un passo poco chiaro, segnalalo scrivendo al progetto (proactiveproject.eu/progetto).

Testo di questa guida: licenza CC BY-SA 4.0. Puoi copiarlo, tradurlo e modificarlo, anche per pubblicarlo altrove, citando **proactiveproject** e mantenendo la stessa licenza. **Una richiesta, non un obbligo:** se migliori la guida, proponi la modifica scrivendo al progetto (proactiveproject.eu/progetto), così il miglioramento arriva a tutti.