How to Set Up SSH on Linux

Turning a Linux machine into one you can reach, in about four commands

This is the server side of SSH: turning a Linux machine into one you can connect to. If you only need to reach a machine someone else already set up, you do not need any of this; start with "What SSH Is and Why You'd Use It."

Setting up SSH on Linux takes about three minutes, because it was built to work this way. If you have seen the Windows version of this and come away rattled, relax; Linux does not make you fight it. This covers Ubuntu and Debian-based systems, which is most of what you will run into.

Quick orientation, in case you have not read part 1 of the SSH series: the machine you are setting up here is the **server**, the thing that accepts connections. Your laptop, the thing you type into, is the **client**. Everything below happens on the server.

Step 1: install the SSH server

Open a terminal on the Linux machine and run:

```
sudo apt update && sudo apt install openssh-server -y
```

That is the whole installation, and it takes about thirty seconds. The `-y` flag answers "yes" to the confirmation prompt so you do not have to.

Step 2: confirm it is running

```
sudo systemctl status ssh
```

You want to see `active (running)` in green. If it is not running, start it and set it to come back on its own after a reboot:

```
sudo systemctl start ssh
sudo systemctl enable ssh
```

Running both is harmless, so run both.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Step 3: open the firewall

If you are running UFW, Ubuntu's firewall, and you should be, tell it to allow SSH and confirm:

```
sudo ufw allow ssh
sudo ufw status
```

SSH should now appear in the list of what is allowed through.

Step 4: find the machine's IP address

Other computers reach this server by its address. Find it with:

```
ip a
```

Look for your network adapter, usually `eth0` for wired or `wlan0` for wireless, and read the `inet` line. That is the local IP, something like `192.168.1.112`.

Step 5: connect from another machine

From any other computer on the same network:

```
ssh your-username@192.168.1.112
```

Swap in your actual Linux username and the IP you just found. It asks for your password; type it and press Enter, and you are in.

One thing that trips people up: Linux does not show anything as you type a password. No dots, no moving cursor. That is deliberate, not a frozen screen. Type it and press Enter.

Stop typing passwords

Password login works, and it is the weak link. The better setup is an SSH key, which is more secure and means no password prompt at all. That is its own tip in this series: see "How and Why to Use an SSH Key" (part 2), which walks through `ssh-keygen` and `ssh-copy-id` step by step.

Quick reference

Task	Command
Install the SSH server	`sudo apt install openssh-server -y`
Check status	`sudo systemctl status ssh`
Start it	`sudo systemctl start ssh`

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Task	Command
Start on boot	`sudo systemctl enable ssh`
Open the firewall	`sudo ufw allow ssh`
Find the IP	`ip a`
Connect from another machine	`ssh username@ip-address`
Switch to key login	See part 2, How and Why to Use an SSH Key

That is a Linux box you can reach from anywhere on your network, set up in the time it takes to read this.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.