Set up SSH keys without locking yourself out

Prepare a recovery route

Use a separate SSH key for each administrator or device. The private key stays on your computer; only its public half belongs on the VPS. Before changing authentication, confirm that the browser console works and keep your current SSH session open. Know which Linux account you intend to use and whether it has the required sudo access.

Record the server address, port and trusted host-key fingerprint. A new fingerprint after an unexpected change needs investigation; do not disable host verification to make a warning disappear. Back up the current SSH configuration, including included snippets, before editing it.

Create a key on your own computer

On a computer with OpenSSH, create an Ed25519 key if your organisation and target server support it:

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

Choose an unused filename so you do not overwrite an existing key. Protect it with a passphrase and use your operating system's SSH agent when convenient. A passphrase protects the local key; it is different from the remote account password.

For an existing account, `ssh-copy-id` can append your chosen public key when that tool is available. This example requires replacing the username and hostname:

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub username@server.example
```

If copying manually, append the complete public-key line to the intended account's `~/.ssh/authorized_keys`. Preserve existing authorised keys. The directory normally uses mode `700` and the file `600`, owned by that account. Never paste the private-key file into the server, a ticket or a public chat.

Test a completely new connection

Open a second terminal and authenticate with the intended key. Confirm that you reached the correct machine and account. Test `sudo -v` if administrative access is required. Do this while the original session and recovery console remain available.

If key authentication fails, inspect the server authentication logs, username, file ownership and effective SSH configuration. Do not disable permission checks or grant a key to every user. A successful connection through a different cached key is not proof that the new key works; explicitly select the intended identity when testing.

Restrict passwords only after verification

On Debian or Ubuntu, inspect `/etc/ssh/sshd_config` and its included snippets. OpenSSH generally uses the first obtained value, and `Match` blocks can change a specific connection's result. Do not assume a later filename overrides an earlier cloud-init setting.

If switching to key-only authentication, review both `PasswordAuthentication` and `KbdInteractiveAuthentication`. Keyboard-interactive access may be part of a deliberate MFA design, so do not turn it off blindly. Use `sudo /usr/sbin/sshd -t` to check syntax, and inspect effective settings with `sshd -T`; use connection criteria for relevant Match blocks.

Reload the confirmed SSH service only after validation, then test another new session. Keep your old session until this succeeds. If it fails, restore the saved settings from that session or console and validate again. Removing a person's access requires reviewing their authorised keys as well as their password.

For service access, see VPS access and console recovery. When contacting us, follow the support-ticket checklist.

Further reading

- Ubuntu OpenSSH setup
- OpenSSH configuration semantics
- Account and key access

Revision #2
Created 2026-11-08 21:45:13 UTC by CloudExa Hosting Admin
Updated 2026-11-08 21:45:13 UTC by CloudExa Hosting Admin