Hetzner Cloud — API y modo Rescue de VPS

Hetzner Cloud — API y modo Rescue de VPS

Guía para arrancar un VPS de Hetzner Cloud en **modo Rescue** (sistema mínimo de mantenimiento) tanto desde la consola web como desde la API, con la lección aprendida en producción sobre cómo inyectar la llave SSH correctamente.

Cuándo se usa el modo Rescue

- Para **editar el sistema o la configuración de arranque** de un VPS que no puede arrancar por su cuenta (por ejemplo, retirar servicios del boot, reparar `fstab`, recuperar contraseñas).
- Para **encender un VPS sin que levante los servicios de producción** (mantenimiento, failover, failback).
- El sistema Rescue es una **imagen mínima independiente** que no monta el disco del sistema por defecto: tú lo montas a mano (`mount /dev/sda1 /mnt`) para trabajar sobre la raíz real.

Método de acceso: contraseña o llave registrada

Al activar el Rescue hay que elegir **cómo entrar después**:

- **Contraseña**: Hetzner genera una password que se muestra una sola vez.

- **Llave SSH registrada en el proyecto**: aquí conviene elegir la **más moderna (ED25519)** de las registradas. El arranque Rescue escucha SSH en el **puerto 22** (el puerto de gestión habitual del VPS puede ser otro).

> ⚠ La **clave de host cambia** en modo Rescue (es otro sistema). En la conexión de rescate hay que saltarse la verificación del host key (`StrictHostKeyChecking=no` + `UserKnownHostsFile=/dev/null`) **solo** para esa sesión, sin contaminar el `known_hosts` normal.

Por consola web (panel de Hetzner)

1. Servidor → pestaña **Rescue**.
2. Sistema **linux64** → método de acceso (llave ED25519 recomendada).
3. **Power on** si está apagado.
4. `ssh -p 22 root@<ip-del-vps>` con la llave elegida.
5. Al terminar: **desactivar Rescue** en el panel y **Reiniciar**.

Por API — lección importante sobre la llave

La API permite activar el Rescue completo (activar, encender, apagar, desactivar). **Punto aprendido en producción:** la llave se inyecta por el parámetro `ssh_keys` con el **ID o fingerprint de una llave registrada en el proyecto** — no vale pasar una `public_key` suelta: la API acepta la acción, pero luego el SSH del Rescue **no autentica** con esa llave (`Permission denied`), aunque sea la misma que sí entra vía `ssh_keys`.

Secuencia validada (token de la API de Hetzner Cloud; servidor y fingerprint de ejemplo):

```
TOKEN="<token>"; SVR="<server-id>"
FP="<fingerprint-del-id-de-la-llave-numerico>"

# Estado actual
curl -sS -H "Authorization: Bearer $TOKEN" "https://api.hetzner.cloud/v1/servers/$SVR"
```

```
# Activar Rescue CON la llave registrada (id o fingerprint). Esperar status success.
curl -sS -X POST -H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json" \
  -d "{\"type\":\"linux64\",\"ssh_keys\":[\"$FP\"]}" \
  "https://api.hetzner.cloud/v1/servers/$SVR/actions/enable_rescue"

# Power on si está apagado, y esperar status running
curl -sS -X POST -H "Authorization: Bearer $TOKEN" \
  "https://api.hetzner.cloud/v1/servers/$SVR/actions/poweron"

# SSH al Rescue (host key distinto al normal)
ssh -p 22 -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no root@<ip-del-vps>
#    mount /dev/sda1 /mnt   → trabajar sobre la raíz (ej. retirar enlaces de systemd)
#    umount /mnt            → antes de apagar

# Desactivar Rescue y reinicio normal
curl -sS -X POST -H "Authorization: Bearer $TOKEN" \
  "https://api.hetzner.cloud/v1/servers/$SVR/actions/disable_rescue"
curl -sS -X POST -H "Authorization: Bearer $TOKEN" \
  "https://api.hetzner.cloud/v1/servers/$SVR/actions/poweroff"
# esperar status off, luego:
curl -sS -X POST -H "Authorization: Bearer $TOKEN" \
  "https://api.hetzner.cloud/v1/servers/$SVR/actions/poweron"
```

Verificación tras el arranque normal

- El disco raíz se monta en `/dev/sda1` (comprobarlo con `lsblk` en el Rescue).
- Al volver al arranque normal, confirmar que el sistema **no** levantó los servicios de producción que se retiraron del boot.
- Si se retiraron enlaces de arranque de systemd, comprobar que no quedan: `ls /etc/systemd/system/multi-user.target.wants/` y `/etc/systemd/system/timers.target.wants/`.

Notas

- El fingerprint se obtiene desde el panel o desde la propia API (`/api/ssh_keys`), y corresponde al **ID numérico** para el campo `ssh_keys`.

- Un `schedule:run` aislado con el runtime parado no es un servicio levantado: no asumir que el VPS volvió a producción solo por ver un proceso PHP suelto.

Revision #1
Created 2026-11-22 07:04:09 UTC by Abkrim
Updated 2026-11-22 07:04:09 UTC by Abkrim