interweb

Generated on 24/11/2026 11:02

KNOWLEDGE BASE · PUBLIC DOCUMENT

Scripts for your servers: restart or reinstall a VPS and keep nameservers and DNS up to date

What You Can Automate

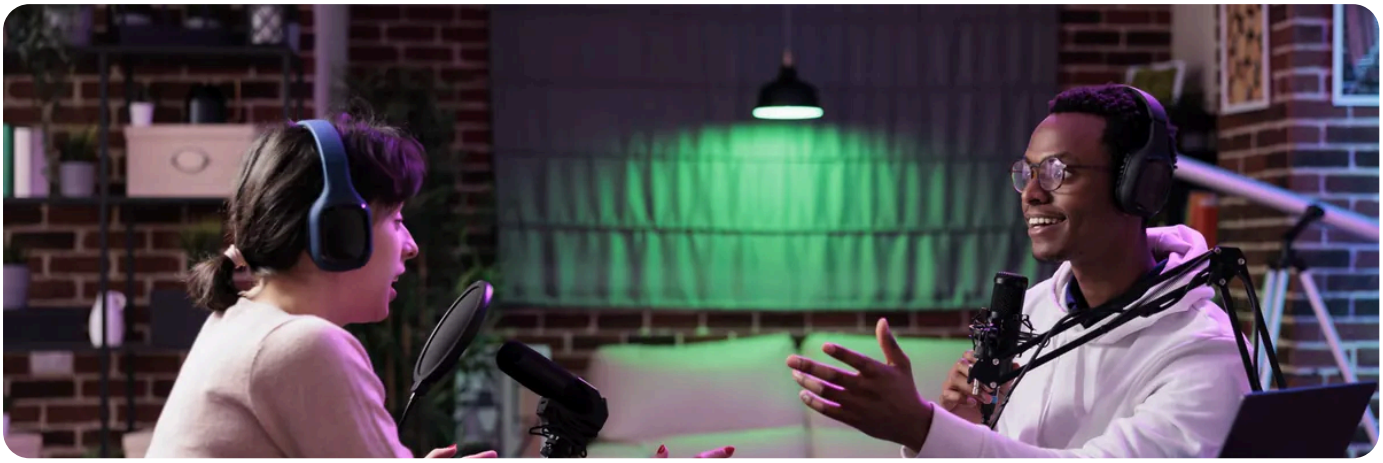
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-2796	v2026.11.24	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/2796

Scripts for your servers: restart or reinstall a VPS and keep nameservers and DNS up to date

What You Can Automate Created on **November 24, 2026** 6 min read



The short answer: a script on your computer, on a monitoring server or in your deployment routine can **start, stop, restart and reinstall** your VPS servers and keep your domains' **nameservers** and **DNS records** the way they should be. This guide brings the two areas together and shows the habits that make a script safe: check the state before acting, follow the action to the end, retry only when it is worth it and leave alone what is already right.

The script's key

Task	Permission
See the servers and the state of each one	`vps:read`
Start, stop, restart and change reverse DNS	`vps:write`
Reinstall	`vps:reinstall` (sensitive, off by default)
Read and change nameservers	`domains:read` , `domains:write`
Read and change DNS records	`dns:read` , `dns:write`

Give the script only what it does. A monitoring script that restarts does not need `vps:reinstall` : that one lives in a separate key, used by hand. Fill in the key's IP list with the IP the script runs from, and send the `X-API-Mode: production` header on every request: if the key is in the other mode, the API refuses with `invalid_request` and `mode_mismatch` , instead of the script quietly running in Sandbox. Sensitive permissions are explained in **sensitive API permissions**.

Restarting and following it to the end

1. **Check the state.** `GET /v1/servers/{id}` returns `power_state` (`running`, `stopped`, `restarting` and others), `state_checked_at` and, if it could not be read right now, `state_stale` set to `true`. The `id` is the same as the service's, and `GET /v1/servers` lists them.

2. **Ask for the action** with `POST /v1/servers/{id}/restart` (or `start`, or `stop`). The answer is `202` with an `action_id`.

3. **Follow it** with `GET /v1/actions/{id}`: `running` while it runs, then `done` or `failed`. The state comes from a copy up to 20 seconds old, so asking more often gains nothing.

4. **Use the same key for both requests:** only the key that asked for the action can look it up.

An example in bash, using `jq` to read the JSON (any language will do; `700` is the server's `id`):

```
ACTION=$(curl -s -X POST https://api.interweb.ao/v1/servers/700/restart \
-H "Authorization: Bearer $INTERWEB_API_KEY" -H "X-API-Mode: production" | jq -r
'.data.action_id')
while true; do
STATUS=$(curl -s https://api.interweb.ao/v1/actions/$ACTION \
-H "Authorization: Bearer $INTERWEB_API_KEY" | jq -r '.data.status')
[ "$STATUS" != "running" ] && break
sleep 20
done
echo "Result: $STATUS"
```

In a real script, first make sure the restart answered `202`. If the action ends `failed` with `timed_out` in `result.reason`, check the server's state; with `server_gone`, the server is no longer available. More on the `action_id` in **long-running API operations**.

The answers your script should expect

Answer	When	What to do
`409 already_running` / `already_stopped`	You asked to start a server that is already running, or stop one already stopped.	Nothing: it is already as you wanted.
`409 server_stopped`	You asked to restart a server that is stopped.	Use `start`.

Answer	When	What to do
`409 action_in_progress`	The server is already doing another action. Its `action_id` comes in `details`.	Follow that action instead of asking for another.
`409 service_not_active`	The service is suspended: data can be read, actions cannot.	Check your invoices in Meu Interweb.
`429 rate_limited`	15 seconds between actions on the same server, 10 actions a minute across the account.	Wait the number of seconds in `Retry-After`.
`500`, `502`, `503`	A failure on our side or in the service behind the API, or the API is turned off.	Try again later and, if it persists, give support the `X-Correlation-ID`.

Reinstalling without surprises

1. **Choose the system** from `GET /v1/servers/{id}/operating-systems` and keep its `id`.

2. **Ask for the reinstall** with `POST /v1/servers/{id}/reinstall`: `confirm` with the server's name (the `name` from `GET /v1/servers/{id}`), `operating_system` and `backup_first`, plus the `Idempotency-Key` header.

3. **Save the root password at once:** it only appears in that answer. Repeating with the same `Idempotency-Key` does not create another and returns `password` as `null`.

4. **Follow it by the** `action_id`. With `backup_first` set to `true`, the reinstall starts by itself once the backup appears, which can take up to an hour; if the server already has 4 backups, the answer is `409 backup_limit_reached` and nothing is erased.

```
curl https://api.interweb.ao/v1/servers/700/reinstall -X POST \
-H "Authorization: Bearer $REINSTALL_KEY" \
-H "Content-Type: application/json" \
-H "Idempotency-Key: 0f8fad5b-d9cb-469f-a165-70867728950e" \
-d '{"confirm": "web1.asuaempresa.ao", "operating_system": "ubuntu-24.04", "backup_first": true}'
```

> **A reinstall erases everything on the server.**
> There is a limit of 3 reinstalls or restores an hour, across the account. In Sandbox nothing is touched and the password is not real. After reinstalling, log in with an SSH key, as we explain in **how to create an SSH key**. Every VPS action is covered in **controlling a VPS through the API**.

Keeping nameservers the way they should be

1. **Read them** with `GET /v1/domains/{domain}/nameservers`: it returns `nameservers` and `uses_our_nameservers`.
2. **Compare** with the list you want. If it matches, the script does nothing else.
3. **Only if it differs,** change them with `PUT /v1/domains/{domain}/nameservers`, with 2 to 5 nameservers. What you change through the API shows straight away.

```
curl https://api.interweb.ao/v1/domains/asuaempresa.ao/nameservers -X PUT \
-H "Authorization: Bearer $INTERWEB_API_KEY" \
-H "Content-Type: application/json" \
-d '{"nameservers": ["ns1.asuaempresa.ao", "ns2.asuaempresa.ao"]}'
```

The read may come from a copy up to 60 seconds old, which does not count towards the limit of 30 reads a minute. If the nameservers are your own VPS's, see **pointing a domain at your own VPS**; to go back to ours, **the nameservers to use with us**.

> **The domain of your account's e-mail is protected.**
> If the e-mail you log in with, or that of one of the account's users, is on this domain, changing the nameservers answers `409 account_email_domain`, in Sandbox too. Make that change in Meu Interweb.

Updating a DNS record when the IP changes

The typical case: the site moved to another VPS, or a recovery script sends traffic to a standby server.

1. **Read the zone** with `GET /v1/domains/{domain}/dns` and find the record by `name` and `type`.
2. **If the `value` is already the right IP, do nothing.** It saves requests and does not create zone backups for no reason.

3 **If it differs,** change it with `PUT /v1/domains/{domain}/dns/{record_id}`, with `name`, `type`, `value` and `ttl`.

4 **Keep the `id` from the answer.** A record's `id` changes when its value changes; changing only the `ttl` keeps it. Always use the one from the latest list.

```
curl https://api.interweb.ao/v1/domains/asuaempresa.ao/dns/3f2a91c07d11e4b6 -X PUT \
-H "Authorization: Bearer $INTERWEB_API_KEY" \
-H "Content-Type: application/json" \
-d '{"name": "www", "type": "A", "value": "192.0.2.20", "ttl": 14400}'
```

There is always a way back:
Before each change a copy of the zone is kept, the last 20 per domain stay, and `POST /v1/domains/{domain}/dns/backups/{id}/restore` restores one. DNS and nameserver changes have a limit of 20 a minute, per account. For a planned move, **lower the TTL first**; if the VPS sends e-mail, also set its reverse DNS with `PUT /v1/servers/{id}/reverse-dns`, as we explain in **PTR and reverse DNS**. The plans are on the **VPS servers** page.

RECOMMENDED

Want your scripts to talk to your account? See the API page and start with a Sandbox key, which touches no server.

See the API page →

SEE ALSO

- **Controlling a VPS through the API: start, stop, restart, root password, reinstall and backups** →
- **DNS through the API: adding, changing and deleting records, and restoring a zone backup** →
- **Long-running API operations: the 202 answer, the action_id and following the result** →
- **Domains through the API: checking a name, registering, renewing and transferring with credit** →
- **Pointing a domain at your own VPS: nameservers and glue records** →
- **Getting an alert when your site goes down: a simple uptime check** →

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-2796 · v2026.11.24 · Generated on 24/11/2026 11:02