NACARSOFT · CONNECTION GATEWAY

One single entry point to your whole environment.

NSGateway concentrates terminals, SSH servers, internal applications and public portals: a reverse proxy with login, RBAC and auditing in a single binary.

SEE FEATURES →

HTTPS · RBAC · Auditing · No clients

NacarSoft



one entry point for your whole environment

sessions · apps · groups · audit · proxy

EL PROBLEMA

Access to servers and consoles does not scale

Every operator carries their own client, keys and tunnels: sessions cannot be audited and internal consoles stay outside the access policy.

01 Keys per person

A keyring nobody maintains. Rotation becomes a quarter-end project.

02 No central identity

Everyone gets in as they can. No user, no role, no expiry.

03 No audit

The SSH login leaves no useful trace: nobody knows what ran or from where.

04 Loose consoles

The intranet is published on its own, bypassing the access policy.

- **Scattered clients and keys**
- **No central identity**
- **Untraceable consoles**

QUÉ ES

The connection gateway for your environment

NSGateway brings the web terminal, SSH access, reverse proxy and public portals together in a single binary. The administrator publishes applications and targets; the user enters through one URL with their identity, RBAC and audit.

- ✓ xterm terminals with tabs, reconnection and Natural PF keys
- ✓ SSH applications with no client to install: host, port and authentication
- ✓ Reverse proxy for internal applications, with allowlist and SSRF protection
- ✓ Public/private portals and Host-based sites: limits, logs and audit

0
clients to install

1
binary, no dependencies

HTTPS
the only network need

4
access modes

ARQUITECTURA

How a keystroke travels, step by step

1 — Browser

xterm.js: key, mouse and scroll travel over WebSocket.

2 — Web session

Cookie + JWT: the user is already authenticated and role-bound.

3 — Check

Group quota, active app and allowed target before opening.

4 — PTY on the host

The command starts with the assigned user and its environment.

5 — Live output

The output returns to xterm and is recorded as a session.

⚠ The session lives on the server: if the browser closes, the process keeps running and can be picked up later.

HTTPS and WSS only

No shell on the client

Auditable session

SIN SSH

No clients: HTTPS is enough

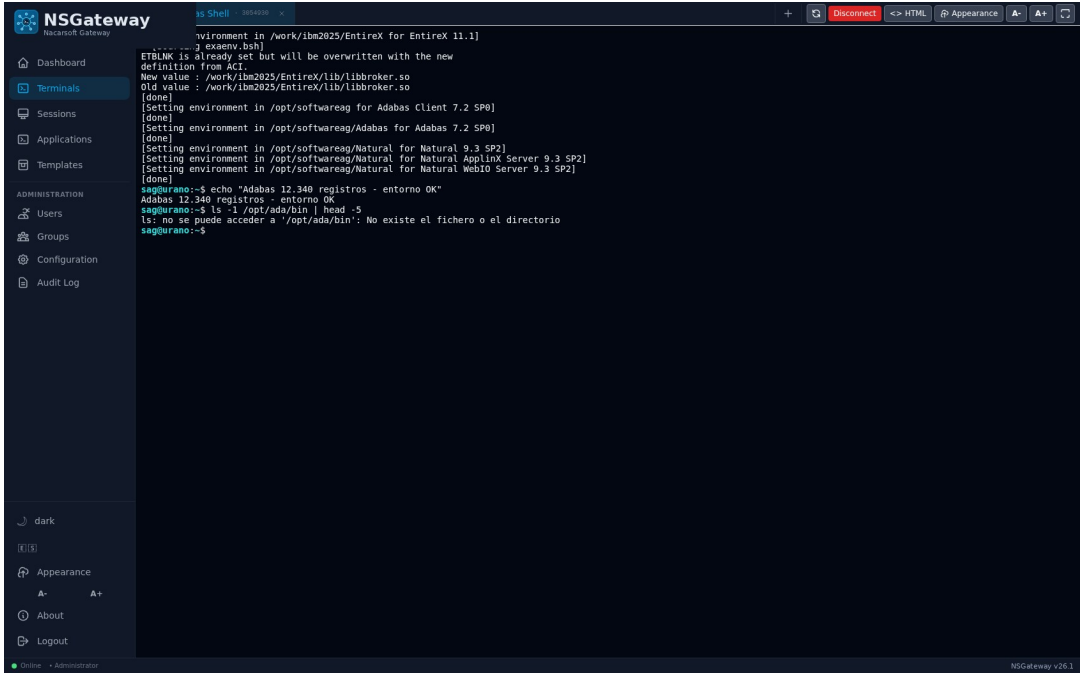
No sshd, no keys, no bastion, no tunnels. NSGateway concentrates access and speaks WebSocket over TLS; the user just opens a URL.

Access	SSH client + private key	Browser with a session cookie
Identity	The user's public key	Panel user, group and role
Audit	At best, auth.log	Log per session, user and IP
Quotas	None	Max sessions per user and group
Internal consoles	Tunnels and port forwarding	Reverse proxy with RBAC
For the user	Install and configure a client	Open a URL

The only network requirement is a reverse proxy with HTTPS on port 11030.

CONEXIONES

From the desktop browser



A real terminal in the browser, with the session PID on the tab.

- xterm.js with the server character set and colours
- Tabs: several terminals at once, with name and PID
- Reconnection: reload the page and the session is picked up
- Full screen, font sizing and HTML mirror
- Shortcuts and Natural function keys (PF) when the app asks for them

No plugin

No PuTTY Java

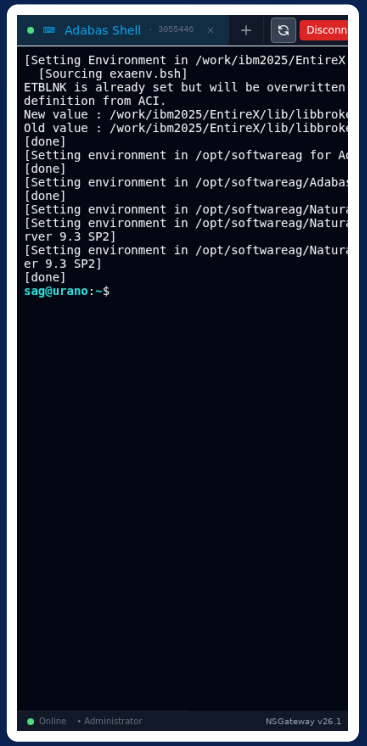
Bookmarks and favourites

MÓVIL Y TABLETA

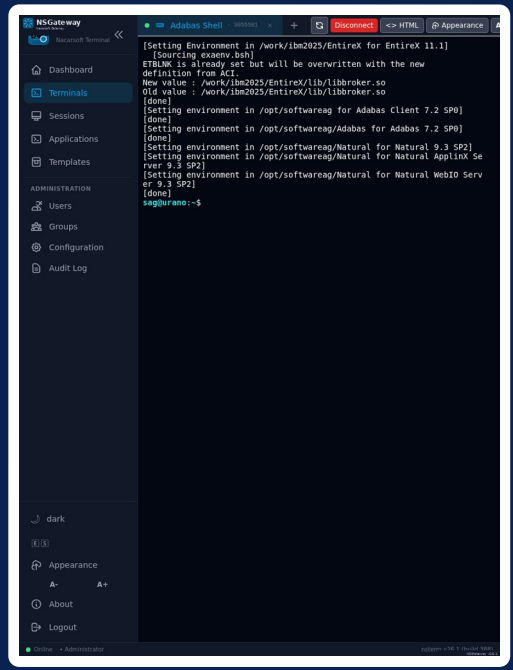
Also from the tablet and the phone

- The same session, with the layout adapted to the screen width
- On-screen keyboard with the keys the app needs
- Long press and text selection over the command output
- Ideal for one-off queries and on-call duty away from the desk

The very same link works on the PC, the tablet and the phone.



Phone: 390 px, the terminal adapts without losing commands.

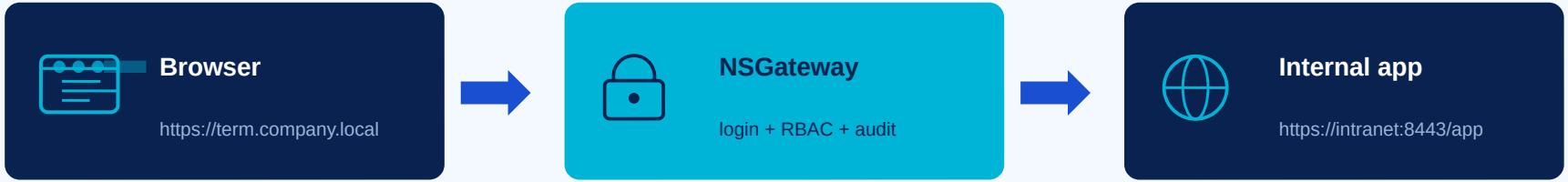


Tablet: 834 px, a comfortable working width.

PROXY INVERSO

Reverse proxy: publish your internal consoles

Beyond terminals, NSGateway publishes internal web applications: it exposes an http/https URL through NSGateway itself, inheriting the login, the per-group RBAC and the audit.



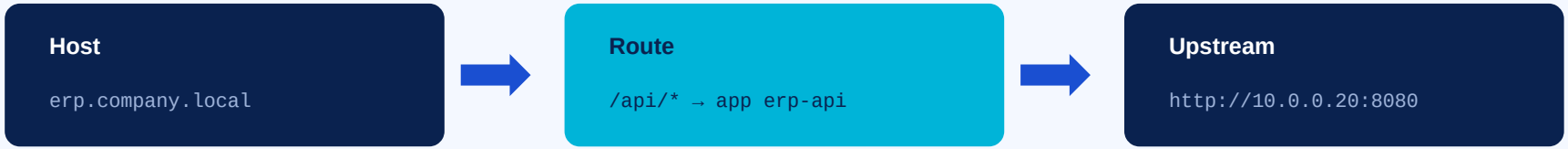
- Only admin-defined URLs: the user never types one
- Host allowlist, private-IP and cloud-metadata blocking
- Link rewriting and WebSocket support on the backend
- Per-application body limit and response timeout

The internal console leaves the network and goes through the very same login as the terminal.

PUBLIC SITES

One Host, its routes: Caddy-style inside NSGateway

Publish internal applications as independent sites by Host, without writing a Caddyfile: NSGateway routes by Host and path, compresses, redirects and logs every request.



- Exact or wildcard Host (*.company.local); routes /api/*, /health or /*
- Proxy to an existing app or redirect 301/302/307/308
- strip_prefix, preserve_host and request/response headers
- gzip/zstd and JSONL log per site or route
- Read-only Caddyfile tab with the equivalent block

```
erp.company.local {
  encode gzip zstd
  /api/*  →  erp-api (strip)
  /old    →  /new   301
}
```

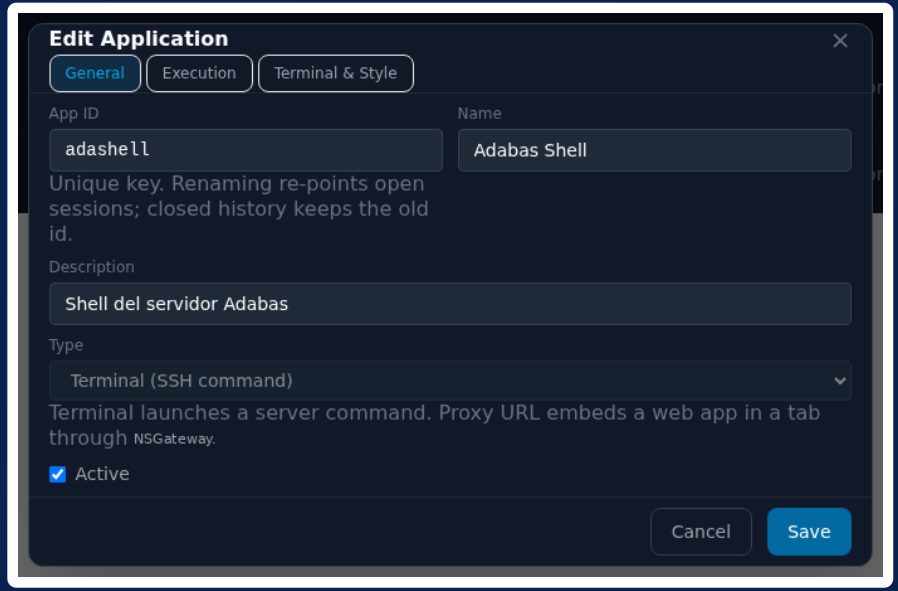
No own TLS/ACME, static files, basic_auth or load balancing in this version.

Public sites: no login or RBAC; WAF, rate limit, blocklist and SSRF still apply. TLS terminates in front.

ADMINISTRACIÓN

Applications: the admin decides what is published

- Command, arguments, working directory and environment
- Terminal size (cols/rows) or fit to the container
- Colour scheme, icon, logo and application colour
- System user it runs as: fixed or the connected user's account
- Session timeout, HTML design and Natural keys



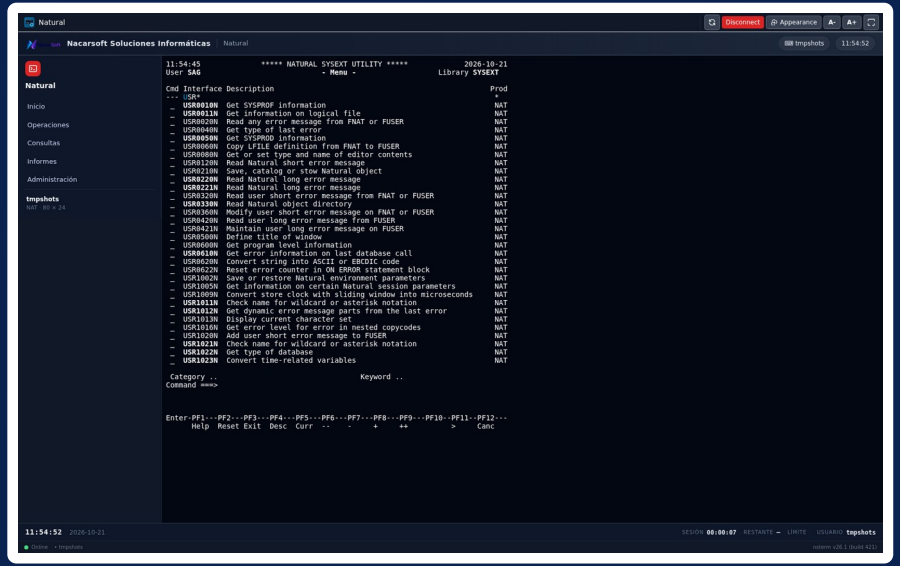
The application editor: id, name, description, type and state.

A user only sees the applications assigned to their groups.

PLANTILLAS

Templates: the terminal with a web application look

- Header, side containers and status bar with your own HTML and CSS
- Corporate logo and name, taken from the server configuration
- PF1..PF12 buttons that send the key to the session
- Live values: clock, date, session time and remaining time
- The CSS runs isolated: it cannot break the NSGateway interface



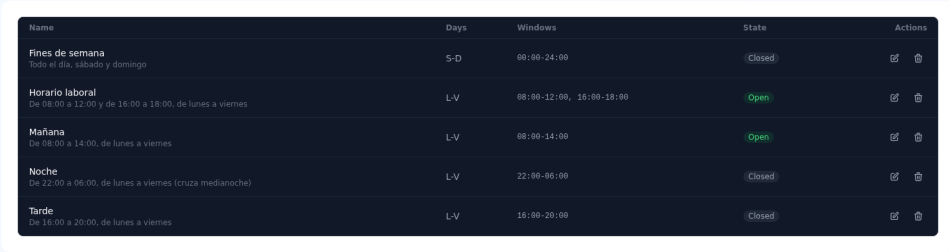
The same application, with a corporate header, side menu and a status bar with the clock.

Eleven example templates are created on install: assign the one you want to each application.

HORARIOS

Schedules: the application is only available in its window

- Schedule templates: morning, afternoon, night, weekend
- Outside the window the app shows as Closed on the dashboard
- The server blocks it: it cannot be opened bypassing the panel
- Window per day, several windows and night shifts
- Administrators are subject to the schedule as well



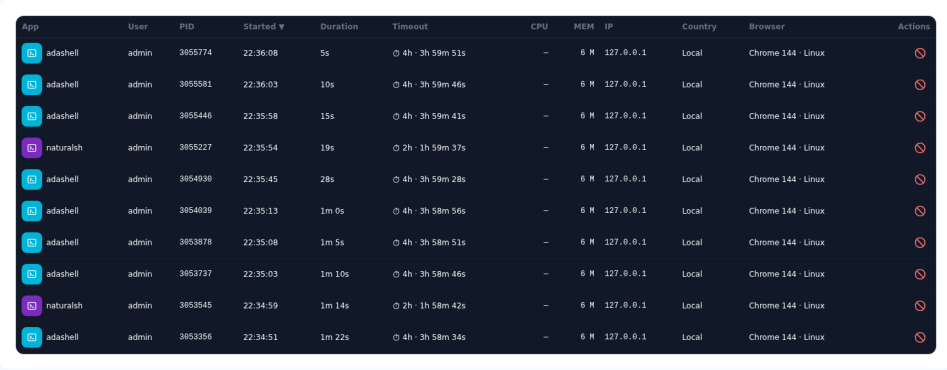
The available schedule templates and their current state.

Console access becomes office hours, instead of always being open.

SESIONES

Every session is a process, and it is recorded

- Application, user, PID, start and duration of every session
- IP, country and browser of whoever opened it
- Child processes: what the command has spawned
- CPU, memory and the configured session timeout
- Kill the session asking for a reason, which goes to the log



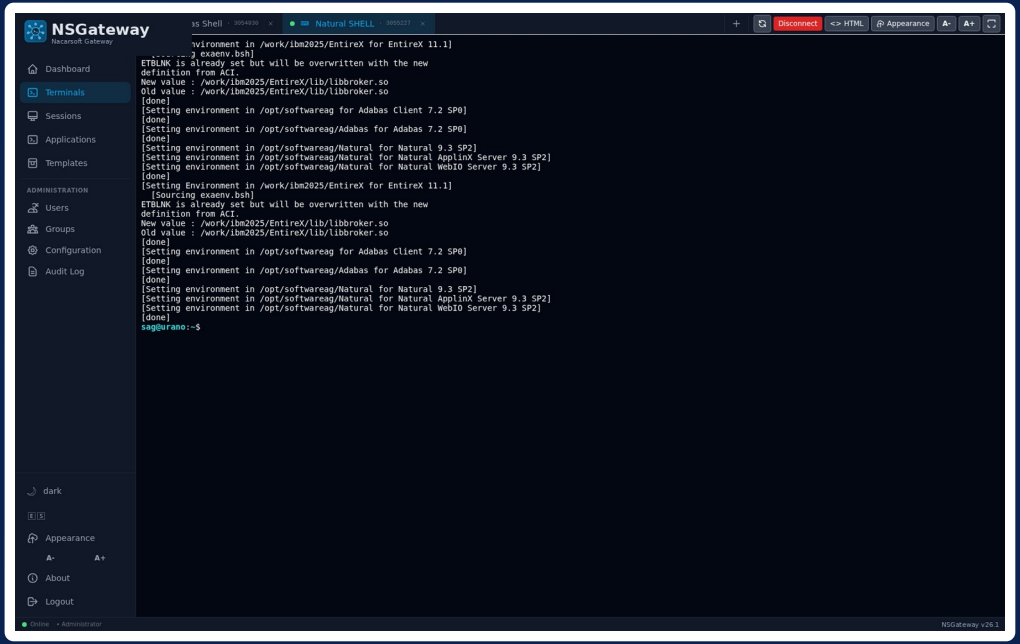
Sessions: who runs what, from where and since when.

The table updates live over WebSocket: no need to reload.

MULTITERMINAL

Several terminals, one window

- A new tab without losing the previous session
- Each tab with its own PID and application name
- Automatic reconnection after a network drop
- The session state, visible in the tab bar



Tabs: one terminal per session, all in the same window.

USUARIOS Y GRUPOS

Users, groups and session quotas

- A user belongs to groups; the group defines the role
- Each group has its applications assigned
- Per-user, per-app quota: the group cap wins; 0 = unlimited
- Past the quota, the terminal warns and opens no session
- User administration: alias, email and state



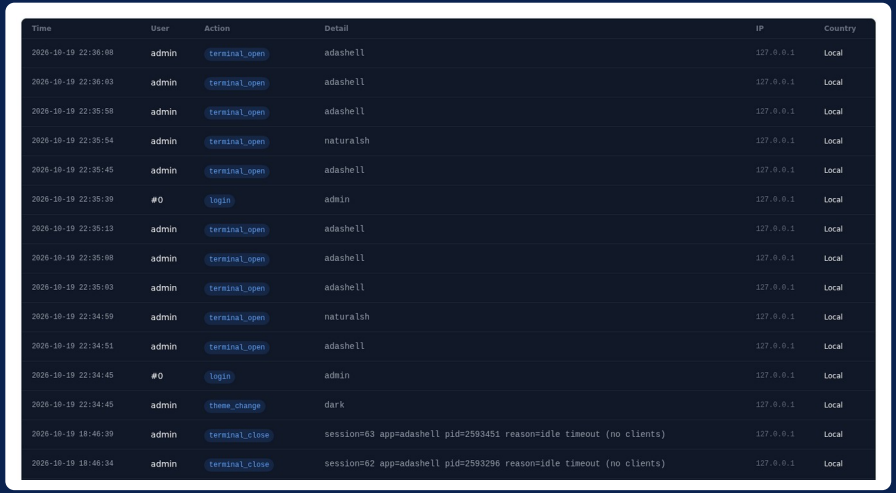
Groups: role, assigned apps and the maximum sessions per user.

The same application can live in several groups, with different quotas.

SEGURIDAD

Who opened what, and it is recorded

- Logins, app creations, group changes and session endings
- Mandatory reason when closing somebody else's session
- Optional LDAP or Active Directory, alongside local users
- Password policies: expiry, early warning and history
- Session token with configurable expiry and HttpOnly cookie



Time	User	Action	Detail	IP	Country
2026-10-19 22:36:08	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:36:03	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:35:58	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:35:54	admin	terminal_open	naturalsh	127.0.0.1	Local
2026-10-19 22:35:45	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:35:39	#0	login	admin	127.0.0.1	Local
2026-10-19 22:35:13	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:35:08	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:35:03	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:34:59	admin	terminal_open	naturalsh	127.0.0.1	Local
2026-10-19 22:34:51	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:34:45	#0	login	admin	127.0.0.1	Local
2026-10-19 22:34:45	admin	theme_change	dark	127.0.0.1	Local
2026-10-19 18:46:39	admin	terminal_close	session=63 app=adashell pid=2593451 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:46:34	admin	terminal_close	session=62 app=adashell pid=2593296 reason=idle timeout (no clients)	127.0.0.1	Local

Audit log: logins, applications, groups and session endings.

Machine-signed license and a 30-day trial.

GETTING STARTED

From zero to a gateway, in three steps

1 Create the environment

nsgateway init /opt/nsgateway: it creates config.yaml, the SQLite database, the logs and the admin user.

2 Publish the reverse proxy

Terminate TLS at the proxy and forward to https://server:11030. The server itself is not touched.

3 Publish the shell

Create the application (command, args, directory and environment) and assign it to the groups that need it.

```
nsgateway init /opt/nsgateway   →   nsgateway start --config /opt/nsgateway
→   https://term.company.local
```

REQUIREMENTS

- Linux (RHEL 8 / 9)
- A single binary, no dependencies
- SQLite for state and audit
- Reverse proxy with HTTPS in production
- Port 11030 (configurable)

No SSH: no need to open port 22, deploy keys or touch the sshd configuration.

FREQUENTLY ASKED

The questions we hear most

Is it really true that no SSH is needed?

Yes. The browser talks HTTPS and WebSocket with NSGateway and the command runs on the server with its own PTY. sshd is not involved.

Who runs the command?

The admin decides: a fixed system user or the Linux account mapped to the connected user. NSGateway verifies the sudoers permissions.

Can I publish an internal website?

Yes: a Proxy URL app for a single URL, or a site by Host (Caddy-style) with /api/* routes, redirects, compression and logs. Sites are public: WAF and limits still apply.

What if the browser closes?

The session stays alive on the server. When you come back, the tab is restored and re-attached to the same process, with its PID.

How is usage limited?

Each group has a maximum of simultaneous sessions per user and application. Past it, the terminal warns and opens no new session.

Can it integrate with Active Directory?

Yes: LDAP/AD is optional and lives alongside local users. The user is mapped to their Linux account for the PTY.

Your whole environment, through one entry point

Try NSGateway with your environment: we set it up with your applications and watch it run.

Guided demo

Try it in your environment

Let's talk about your case

nacarsoft.com · support@nacarsoft.com

