Superuser Daily Operations

Day-to-day operations for tacctl superusers · Checked against tacctl 0.2.25

Contents

1. Before you start 2
2. Daily health check 2
2.1 Investigating a login problem 3
3. User management 3
4. Device management 4
4.1 Find what you have 5
4.2 Register a new device 6
4.3 Find devices that authenticate but are not registered 6
4.4 Connect to a device 7
4.5 Keep the registry accurate 7
4.6 Notices 7
4.7 A device was replaced or reset (host key changed) 7
4.8 Check device AAA configuration for drift 7
4.9 Slow or distant devices 8
4.10 SNMP credentials: pools and defaults 8
5. Linux hosts 9
6. When something goes wrong 9
7. Undoing a mistake 10
8. Scope housekeeping 10
8.1 The Junos management filter mode 11
9. Logs 12
10. The login console 12
10.1 The system shell 12
10.2 Password cache, colours and list length 13
11. This server 13
11.1 The server's own login 14
11.2 The server address 14
11.3 The overview, the name, the listeners and the metrics exporter 15
12. Quick reference 15

This guide covers the work a tacctl superuser does every day: checking server health, managing users and devices, keeping scopes tidy, and knowing where to go when something goes wrong. Procedures that other roles share have their own pages, linked from here. Installation, upgrades and server internals are in the repository documentation.

> **Example names used on this page** Scopes: `lab`, `site-a`, `site-b` · Devices: `sw1` (Cisco), `rtr1` (Junos), `sw7` (legacy IOS), `pdu1` (WTI) · Host: `web1` · Users: `alice`, `bob`, `carol`

1. Before you start

- Work from the login console (or `tacctl shell` in your system shell): Tab completes commands and live names, `?` and `--help` show help, and the password cache saves retyping your password across many device logins. *Getting Started* §1 covers both.
- Every change takes a snapshot first and writes an audit line, so a mistake can be rolled back (*Backups and Undo*).
- Preview with `--dry-run` or `--check` wherever a command offers it. The conventions every command follows are in *Getting Started* §5.

---

2. Daily health check

Start each shift with these commands. All of them only read data.

```
tacctl status                       # services, stats, errors, password-age warnings
tacctl config validate              # store, settings and rendered configs: problems and
drift
tacctl emergency lockdown status    # any lockdowns still in place?
tacctl breakglass list              # any devices or hosts still held?
tacctl device notice list           # open notices, each with the command that fixes it
tacctl log failures                 # authentication failures, last 24 hours
```

What to look for:

- **`status`**: every enabled backend should be running. Note any users whose passwords are getting old, and the `Password change` line: users who still have to change theirs.
- **`config validate`**: warnings name the command that fixes them. A common one is a scope with no break-glass local users, which is a lockout risk.
- **Lockdowns and holds**: neither one expires on its own. If one is still in place from yesterday, find out why. While a lockdown stands, a red `EMERGENCY LOCKDOWN on scope ...` line also appears on related commands.
- **Notices**: for example, a device whose host key is not pinned. Fix the cause or acknowledge the notice (see §4.6).

2.1 Investigating a login problem

A user says "I can't log in to sw1". Work through these checks in order:

```
tacctl user show alice              # enabled? which group? which scopes? last login
tacctl device show sw1              # which scope answers this device's address?
tacctl scope lookup 10.10.0.1       # which scope owns this address (and any shadowed
overlaps)
tacctl log search alice             # recent log entries for that user
tacctl log tail -f                  # watch live while the user tries again (Ctrl-C to
stop)
tacctl user verify alice            # user details, and checks a password you type
```

Typical causes:

Symptom	Likely cause	Fix
User is `disabled`	Disabled earlier	`tacctl user enable alice`
`user show` has a `Password change` line	A password change is required: no device accepts the user until it is made	The user logs in to the tacctl server and changes it (*Getting Started* §2.1)
Device's scope is not in the user's scopes	Missing scope grant	`tacctl user scope alice add site-a`
`bad secret` in the logs / device `LAST SEEN` shows `rejected`	Device key does not match the scope's secret	Re-paste the AAA config: `tacctl device config show sw1`
No scope covers the device	Address outside all prefixes	`tacctl scope prefixes site-a add 10.10.0.0/16`
Login fails only on one device	Device held for break-glass	`tacctl breakglass list`
A whole group is refused	Emergency lockdown	`tacctl emergency lockdown status`

The full symptom list, including RADIUS, Junos and WTI cases, is in *Troubleshooting* §2.

---

3. User management

Joiners, movers, leavers, passwords and access reviews are in *Access Lifecycle*. The everyday commands:

```
tacctl user list                    # name, group, status, password age, scopes
tacctl user show alice              # full detail, incl. scope membership and last login
tacctl group list                   # groups: Cisco priv-lvl, Junos class, user count
tacctl group show operator          # every setting of a group and where it comes from
```

The optional `engineer` group comes from the preset. A custom group takes a Cisco privilege level and a Junos class:

```
tacctl group preset roles --dry-run     # preview the preset
tacctl group preset roles               # apply it; this creates the engineer group
tacctl group add ops-ro 1 RO-CLASS      # a custom group: priv-lvl, then Junos class
```

Task	Command	Details
New user	`tacctl user add alice operator --scopes site-a,site-b`, then `tacctl host sync --all`	Access Lifecycle §2
Grant or revoke a scope	`tacctl user scope alice add|remove site-b`	Access Lifecycle §3
Promote or demote	`tacctl user move alice engineer`	Access Lifecycle §3
Reset a password	`tacctl user passwd alice` (alice must change it)	Access Lifecycle §5
Require or lift a password change	`tacctl user require-password-change alice [on|off]`	Access Lifecycle §5.1
Suspend / restore	`tacctl user disable|enable carol`	Access Lifecycle §4
Remove	`tacctl user remove carol`	Access Lifecycle §4

> **After every user change, run `tacctl host sync --all`**, or the Linux hosts keep the old state. The group table (Cisco, Junos, WTI, tier) is in *Concepts* §2.

---

4. Device management

Devices are registered by name in the device registry. Registering a device does not restart any service. Commands like `ssh` and `device config` work by device name.

4.1 Find what you have

```
tacctl device list
# every registered device: scope, state, last seen, CONFIG
tacctl device list --scope site-a --vendor juniper
tacctl device list --ip 10.10.0.0/16
tacctl device list --stale             # not seen for stale-days (default 30)
tacctl device list --unconfigured      # registered, but no scope covers the address
tacctl device list --differs           # last pull differs from what is rendered now (no
login)
tacctl device list --probe             # adds REACH: TCP connect to the ssh port

tacctl device show sw1                 # one device in full
tacctl device check sw1                # checklist: scope, reachability, host key, SNMP,
vendor
tacctl device check --all
tacctl device check sw1 --sources      # also test the AAA source addresses (host key, no
login)
```

A `STATE` of `configured` means a scope's prefixes cover the device's address; `unconfigured` means none does, and only configured devices can authenticate users. `stale` marks a device not seen for stale-days, and `BREAK-GLASS` shows while the device is held back from the server (*Break Glass*). The `CONFIG` column is the state of the last pull: `ok`, `differs`, `never`, `failed`, or `pull` for a Cisco device whose configuration syntax is unknown, or whose vendor changed since its last pull (pull it again); `-` for a vendor `other` and a device no scope covers. The enrolled Linux hosts are not in this list: `tacctl host list` shows them (§5).

After a change of users, scopes or settings, `tacctl device list --differs` names the devices to pull or plan. It computes this from the pull records with no login, as `device config list --differs` does, and it counts as a filter: the heading reads `Registered devices (1), of 5` (the second number appears when the list is narrowed by an option or a word), and an empty result says `None match.`

> `device check` and `--probe` connect from the tacctl server. If the server has no route to the management ports, a timeout here may be a false alarm.

`--sources` (and `tacctl device scan --sources`, for every device) connects to each address the device sends its AAA requests from, other than its registered one, and compares its ssh host key with the pinned one. A source that does not answer, or presents another key, is a warning; nothing is pinned or changed (*Engineer Guide* §3.1).

4.2 Register a new device

```
# Cisco switch, register and read its configuration right away
tacctl device add sw1 10.10.0.1 --vendor cisco --pull

# Junos router with a description and location
tacctl device add rtr1 10.10.0.2 --vendor juniper \
    --description "Internet edge A" --snmp-location "Rack 4, DC1"

# Older IOS that only speaks SHA-1 / CBC ssh
tacctl device add sw7 10.20.0.7 --vendor cisco --legacy-ssh

# WTI console server / PDU
tacctl device add pdu1 10.10.0.50 --vendor wti
```

`device add` pins the ssh host keys the device offers and prints their fingerprints. **Compare them with the device's console** before you trust them:

Vendor	Console command
Cisco	`show ip ssh`
Junos	`file show /etc/ssh/ssh_host_ed25519_key.pub`
Linux	`ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub`

If you already know the fingerprint, pin exactly that key: `--host-key SHA256:<fp>`.

Factory-default names such as `switch` or `router` are refused. Name the device on the device itself as well.

Bulk registration from a CSV file (`name,address[,vendor[,port[,description]]]`):

```
tacctl device import --check devices.csv    # preview
tacctl device import devices.csv            # merge into the registry
```

4.3 Find devices that authenticate but are not registered

```
tacctl device discover
# each unregistered address, with a ready 'tacctl device add' line
tacctl device discover --snmp     # also read sysName and vendor by SNMP to fill in the
line
```

Copy the suggested lines, check the names, and run them.

4.4 Connect to a device

`tacctl ssh sw1` connects as you; `tacctl device client-config` gives plain `ssh` the same pinned keys. Refusals and host keys: *Connecting to Devices*.

4.5 Keep the registry accurate

```
tacctl device rename sw1 dc1-sw1
# registry name only; the device's hostname is unchanged
tacctl device rename sw1-new sw1 --replace  # a clone at a bench address takes sw1's place
tacctl device address rtr1 10.10.1.2        # device re-addressed
tacctl device description rtr1 "Internet edge A (ISP-1)"
tacctl device location rtr1 "Rack 5, DC1"
tacctl device location rtr1 --from-device   # take sysLocation from the device
tacctl device vendor sw7 cisco
tacctl device vendor sw7 --from-device      # take the vendor the device reports by SNMP
tacctl device legacy-ssh sw7 enable
tacctl device remove sw7                    # decommissioned (confirms)
```

`device rename <replacement> <name> --replace` removes the original with its files, after a snapshot and one confirmation, and moves the replacement (made with `device clone`) to the original's address. A hold on the original is cleared, so it needs a superuser (*Engineer Guide* §2.5).

4.6 Notices

Notices flag things that need attention, such as an unpinned host key or a generic name.

```
tacctl device notice list              # open notices, each with its fix
tacctl device notice list sw1 --all    # include acknowledged ones
tacctl device notice sw1 ack <kind>    # accept a known condition
tacctl device notice sw1 unack <kind>  # reopen it
```

4.7 A device was replaced or reset (host key changed)

`tacctl ssh` refuses and prints the pinned and offered fingerprints. **Check the new key on the console first**, then pin exactly that key with `tacctl device hostkey sw1 set SHA256:<fp>`. The full procedure is *Connecting to Devices* §5.

4.8 Check device AAA configuration for drift

```
tacctl device config pull --scope site-a      # log in as you and read each device
tacctl device config list --differs           # which devices differ
tacctl device config diff sw1                 # what differs, section by section
tacctl device config plan sw1                 # the exact lines to paste to fix it
tacctl device config apply rtr1 --check       # Junos: device-side commit check only
```

The full change workflow (plans, Cisco paste, Junos apply, WTI checklist, back-out) is in *Engineer Guide: Device Changes*. Superusers follow the same steps, for every scope. Only a superuser can list the shapes of a rollout (`tacctl device config shapes`) or file one that was applied and verified by hand (`tacctl device config shapes seed <name>`); what a shape is and when it counts as confirmed is in *Engineer Guide: Device Changes* §5.2 and §5.3.

4.9 Slow or distant devices

A device behind a satellite circuit (a round trip of about 600 ms) or with a slow CPU can miss the timings tacctl uses to reach it. The timing scale is one whole number from 1 to 10 that multiplies them all (1, the default, changes nothing). Only a superuser sets it:

```
tacctl device timeout-scale rtr1 3         # one device
tacctl scope timeout-scale site-a 3        # every device of a scope
tacctl config devices timeout-scale 2      # every device of the server
tacctl device timeout-scale rtr1           # the scale in force and where it comes from
tacctl device timeout-scale rtr1 clear     # remove the device's own
```

The scale in force is the device's own, else its scope's, else the server's, else 1. `clear` removes a device's or a scope's own. A scope's scale follows `scope rename` and goes with `scope remove`. `device show` and `scope show` have a `Timeout scale` row (and `device show --json` a `timeout_scale` object with `scale` and `from`) when the scale in force is not 1; engineers read it there. The scale multiplies the connect and handshake of the ssh and NETCONF sessions and of `tacctl ssh`, the host-key scans, the SNMP timeout (never above 10 s) and the default time of one device in a pull or an apply (kept between 10 s and 600 s); a typed `--timeout` wins, and the confirmed-commit windows are never scaled. *Connecting to Devices* §7 has the table. `config validate` checks all three stored values, and `rollback` removes them (*Backups and Undo* §5). To find the slow device of a run, use `--timing` (*Engineer Guide* §3).

4.10 SNMP credentials: pools and defaults

tacctl reads SNMP itself (v2c, or v3 at authPriv) for the name, location and vendor lookups of `device add`, `device check` and the like. The settings exist at three levels: the server (`config snmp`), a scope (`scope snmp <scope>`) and a device (`device snmp <name>`). Only a superuser changes them.

At each level the version, port, timeout and protocols are **one setting**. The v3 users are a **pool** of names, each with its passphrases, and the level has one **default** user or none:

```
tacctl config snmp version v3                    # a v3 user does not set the version
tacctl config snmp v3-user add monitor --default # asks for the passphrases twice
tacctl config snmp v3-user list                  # names only, never a passphrase
tacctl config snmp v3-user default monitor       # choose the default (--none clears it)
tacctl scope snmp site-a v3-user add monitor-a --default
tacctl device snmp rtr1 v3-user default monitor-a
# its own, its scope's or the server's user
tacctl config snmp test sw1                      # read one device and say which settings
it used
```

- A device reads with its own default user, else its scope's, else the server's. A device may be told to use a user of its own pool, its scope's or the server's.
- Adding a user changes nothing but the pool: set the version and the default yourself. The same name again replaces the passphrases.
- A name stands for one user wherever a device can see it, so a name that another pool holds is refused, naming the level. A user cannot be removed while a device's default names it. With no default the v3 lookups fail until one is chosen.
- `config snmp version [v2c|v3]` shows or sets the version of the lookups; `community` sets v2c.
- The credential files are version 1 until a pool needs more than one user per level; going back to an earlier release drops the extra users (*Backups and Undo* §4.3 puts them back).

---

5. Linux hosts

```
tacctl host list                    # target, address, scope, state, last seen, by,
notices
tacctl host list --scope site-a --stale     # filters: --name, --ip, --scope, --stale
tacctl host check --all             # read-only: does every host match what tacctl would
make it?
tacctl host sync --all              # push user changes to every enrolled host
```

Enrolling, moving, retiring and recovering hosts are in *Linux Host Operations*.

---

6. When something goes wrong

Situation	First command	Page
A group must lose access to a scope now	`tacctl emergency lockdown <scope> enable --dry-run`	*Emergency Lockdown*
A device or host is locked out after a bad change	`tacctl breakglass hold <name> --reason "<ticket>"`	*Break Glass*
A tacctl change made things worse	`tacctl backup list`	*Backups and Undo*

Situation	First command	Page
Compromised account, leaked secret, mass failures	—	*Incident Response Runbooks*

Lockdowns and holds **never expire**. Check both at every shift start (§2).

---

7. Undoing a mistake

Every change takes a snapshot first: `tacctl backup list`, then `tacctl backup diff`, then `tacctl backup restore <timestamp>`. Read *Backups and Undo* before you restore.

---

8. Scope housekeeping

Scopes bundle client address prefixes with a shared secret. These are the scope tasks that come up during normal operations:

```
tacctl scope list                                  # one row per scope
tacctl scope show site-a                           # prefixes, users, secret length,
settings
tacctl scope routing                               # first-match order across all scopes
tacctl scope lookup 10.10.4.20                     # which scope answers this address?

tacctl scope prefixes site-a add 10.12.0.0/16      # a new site subnet
tacctl scope prefixes site-a remove 10.12.0.0/16

tacctl scope secret site-a show                    # print the shared secret (logged)
tacctl scope secret site-a set                     # set a new one (prompted, not echoed)
tacctl scope secret site-a generate                # generate a new one
```

After a change that affects device configuration, the command tells you which devices now need one, for example `tacctl device config apply --scope site-a --check`.

A new scope, a renamed one and a removed one:

```
tacctl scope add site-c --prefixes 10.13.0.0/16 --secret generate    # prefixes are
required
tacctl scope default site-c                        # the scope `user add` uses without --
scopes
tacctl scope rename site-c site-d                  # users, hosts and settings follow the
name
tacctl scope remove site-d                         # refused while a user has it; --force
strips it
```

A prefix belongs to one scope, so `scope add` refuses a prefix another scope already has. `scope remove` also removes every setting stored under the scope's name, so a new scope of that name starts from the defaults.

> Changing a scope's secret takes effect on the server immediately. Every device in the scope fails to authenticate until its key is updated. Plan the re-paste before you rotate.

8.1 The Junos management filter mode

A scope's Junos management filter is `ssh-only` (the default: ssh, NETCONF and SNMP restricted, everything else accepted) or `strict` (deny by default). *Vendor Notes* §3.4 explains both and the traffic a strict filter stops. Only Junos devices read the mode.

```
tacctl scope mgmt-acl site-a list                # the permit list the filter is built from
tacctl scope mgmt-acl site-a add 198.51.100.1/32  # a routing neighbour: every protocol from it
tacctl scope mgmt-acl site-a mode                # show the mode and where it comes from
tacctl scope mgmt-acl site-a mode strict         # deny by default
tacctl scope mgmt-acl site-a mode ssh-only       # back to the default
tacctl config juniper --scope site-a             # review the filter before it reaches a device
```

> **Check which list the scope renders from before you add to it.** When `list` says `Source: global (mgmt_acl.permits)`, the scope has no list of its own, and the first entry you add with `tacctl scope mgmt-acl <scope> add` replaces the global list for this scope. Add the global entries too, or the global subnets lose access once a strict filter is applied.

Setting `strict` prints what stops and which permit list the scope renders from, and says so when the permit list is empty or has no IPv4 entry that permits ssh (no filter is rendered until it has one). Setting the mode the scope already has changes nothing. The change reaches a device only through a plan or an apply (*Engineer Guide* §4).

> **Keep a console session open** when a strict filter goes onto a device, and use `commit confirmed`.

Rolling tacctl back to a release without the mode removes it from every scope, with a warning; see Rolling back.

---

9. Logs

```
tacctl log tail                    # last 20 entries
tacctl log tail -f                 # follow live (Ctrl-C to stop)
tacctl log tail 100 --backend radius
tacctl log search alice            # by username or keyword (TACACS+: last 7 days)
tacctl log failures                # auth failures, last 24 hours
tacctl log accounting 50           # last 50 accounting records (who ran what)
tacctl log failures --since -1h --user alice   # filters combine
tacctl log accounting --device sw1 --json | jq .
```

`tail`, `search`, `failures` and `accounting` take `--since`, `--until`, `--user` and `--device` (and `--type login|change|command` for accounting); `--json` prints one object per line. With two backends enabled each shows a section per backend, and `--backend <id>` shows one. The RADIUS daemon log names clients, not addresses, so with `--device` its section says so and shows nothing. The filters and their time forms are in *Day-to-Day Device Operations* §1.4.

---

10. The login console

The console is the login shell of every tacctl user on the tacctl server, of every tier; there is no switch for it, and the user's scopes play no part: every user can log in to the server and its console unless disabled (`user disable`) or in a group of tier none. The accounts follow at the next `host sync` of the server; a user stored with a period in the name (0.2.20 and 0.2.21 allowed it) has no Linux account on any host, so no console login. Local accounts that are not tacctl users keep their own shells.

```
tacctl console show       # the settings, each user's shell and system-shell access, and why
tacctl console check      # does sshd apply the console's settings? (operator tier and up)
```

10.1 The system shell

The console word `system-shell` starts a user's own system shell. Who has it is set per tier and per user:

```
tacctl console system-shell                    # each tier, then the users with their own setting
tacctl console system-shell operator enable    # open it to a tier
tacctl console system-shell user bob enable    # open it to one user, over the tier's setting
tacctl console system-shell user bob disable   # take it from one user, over the tier's setting
tacctl console system-shell user bob clear     # the tier decides again
```

- The superuser tier always has it and cannot be disabled. Read-only and operator are disabled until you enable them. The engineer tier never has it: enabling it, or one engineer, is refused.
- A user's `enable` counts only while the user stays in the tier they had when it was set. After a move to another tier it is listed as `stale: set for operator, now readonly` until you set it again. A `disable` holds in any tier.
- `user rename` moves a user's setting to the new name, and `user remove` drops it.
- The same people may open an editor from the console (`scope block … edit`, `device block … edit`), since an editor can start a shell.
- Operators and engineers can read every form; only superusers set them.

10.2 Password cache, colours and list length

```
tacctl console password-cache tiers operator,engineer   # the tiers allowed the cache (default none)
tacctl console password-cache user                      # who has turned it on
tacctl console password-cache user alice clear          # switch alice's choice off
tacctl console color off                                # colour in the console's lists: auto|on|off
tacctl console list-max 100                             # the longest list shown without asking
```

- `password-cache tiers` is the permission only. Within an allowed tier, each user turns the cache on for themselves with `console password-cache on`, and it is off until they do (*Getting Started* §1.3).
- `console color` decides whether Tab's lists, `?`, `-h` and `--help` are in colour in the console: command names, sub-commands, fixed words and options in teal, live names (users, groups, devices, hosts) in bold. `auto`, the default, lets the terminal decide.
- `console list-max` is the longest list the console shows before it asks `Show all N?` (default 40, at most 1000; a negative number never asks).
- Both are server-wide, and each console session reads them when it starts.

---

11. This server

`tacctl server` holds this server's own settings and is the superuser's alone: `server show` (an overview), `server name`, `server addresses`, `server listen` and `server metrics`.

11.1 The server's own login

Every user logs in to the tacctl server and its console through a login that tacctl manages: the scope `tacctl`, this server's entry in the host registry, the PAM client and sshd's drop-in, and the users' accounts. The scope is reserved: no listing or Tab shows it and no command creates, edits or looks it up. Device logins keep the user's own scopes.

- `tacctl install` and `tacctl upgrade` set the login up. `install` asks once (`Set up this machine's own login now, with <address> as the server address? [y/N]`; `--yes` answers it; `--server <ip>` sets the address); `upgrade` asks on a terminal only. A healthy setup prints nothing, and a step that cannot be done safely is one line that names `tacctl host enroll --local`, with nothing left half applied.
- `tacctl host enroll --local` rebuilds, moves or repairs the login. It takes no `--scope`.
- `tacctl host sync <server>` repairs a login changed by hand (`Repaired the managed server login.`), and `host forget` or `host unenroll` of the server takes the login with them.
- A scope called `tacctl` made by an earlier release blocks all of it. `host enroll --local`, the sync of the server and `config validate` stop with `Scope 'tacctl' is reserved. Rename the scope of that name first: tacctl scope rename tacctl <new-name>`, then `tacctl host enroll --local`. Do it before upgrading.
- `emergency lockdown --all-scopes` includes the login (its lines say `this server's own login`); `emergency lockdown tacctl ...` is refused.

11.2 The server address

The address devices and hosts reach this server at is the setting `server.addresses`. It is one IP address (behind NAT or a VPN if need be; not a host name, `0.0.0.0`, `::`, a loopback or a link-local address).

```
tacctl server addresses                      # the recorded address
tacctl server addresses replace 192.0.2.10   # change it
```

`add` sets the address when none is set, `remove` of the only address is refused, and `replace` changes it. A change takes a snapshot and then runs in one apply: the setting, the registry entry, the managed scope's prefixes and this server's PAM client and accounts. If the setup on this machine fails, nothing is changed. Then it says what is left:

```
[INFO] Re-point the devices and hosts at the new address: tacctl host sync --all; the
device
walkthroughs name the new address
```

The recorded address is the default of `--server` for the device walkthroughs (`config cisco|juniper|wti`, and so `device config show`, `pull`, `diff`, `plan` and `apply`), `config linux script` and `host enroll`; `--server` wins for that run. Without a record the address is detected as it was. A host that is already enrolled keeps the address it has, and `host enroll --server` changes it.

11.3 The overview, the name, the listeners and the metrics exporter

```
tacctl server show              # this server at a glance; --json for a script
tacctl server name tacacs-a     # the name the console shows for this server
tacctl server listen            # where the TACACS+ listener binds
tacctl server metrics           # the Prometheus exporter
```

`server show` changes nothing and writes no audit line. It prints the server address (the setting, else the copy in this server's registry entry), a `Name` line (the name in effect, with `the host name` beside it when none is set), the default method of `host enroll`, the metrics exporter (its address, `enabled` or `disabled`, `loopback only` or `externally reachable`), each backend with whether it is installed and enabled and the state of its service, and each listener of the installed or enabled backends with its network, address, port and unit state. `--json` has the same data: `addresses`, `address_source`, `name` (`value` and `source`, `setting` or `host name`), `default_method`, `metrics` and `backends` with their `listeners`.

`server name [<label>|unset]` sets the name the console shows in its prompt (`<name>>`) and its notices (the banner of a session, the header of `device client-config`) in place of the host name. With no word it shows the name in effect and whether it is the setting or the host name; `unset` removes the setting. A name is 1 to 63 lower-case letters, digits and hyphens, a letter first and no hyphen last. Nothing else is renamed: the host name, this server's registry entry and the accounts stay. A console session reads the name once, when it starts.

`server listen [--backend <id>] [--listener <name>] [show|reset|<network> <address>]` shows, changes or resets where a backend listens. It asks for confirmation before `tcp6` or `udp6`, restarts the unit, and puts the old setting back if the unit does not stay up. `server metrics [show|enable|disable|reset|address <host:port>]` controls the Prometheus exporter of the TACACS+ backend (loopback only, 127.0.0.1:8080, by default; `disable` moves it to a port no scraper can reach; an address that is not loopback prints a warning). The device walkthroughs follow the listeners' ports (*Engineer Guide* §4.1).

---

12. Quick reference

Task	Command
Server health	`tacctl status`
Find problems	`tacctl config validate`
Add user	`tacctl user add <name> <group> --scopes <s,...>` then `host sync --all`
Reset password	`tacctl user passwd <name>`
Grant / revoke scope	`tacctl user scope <name> add|remove <s>`
Change role	`tacctl user move <name> <group>`

Task	Command
Suspend / restore user	`tacctl user disable|enable <name>`
Delete user	`tacctl user remove <name>`
Push users to Linux hosts	`tacctl host sync --all`
Register device	`tacctl device add <name> <ip> --vendor cisco|juniper|wti`
Unregistered devices	`tacctl device discover`
Device health	`tacctl device check <name>`
Connect	`tacctl ssh <name>`
Re-pin host key	`tacctl device hostkey <name> set SHA256:<fp>`
AAA config drift	`tacctl device config pull --scope <s>` then `device config list --differs`
Fix plan	`tacctl device config plan <name>`
Which scope owns an IP	`tacctl scope lookup <ip>`
Junos filter mode	`tacctl scope mgmt-acl <s> mode ssh-only|strict`
Lockdown (preview)	`tacctl emergency lockdown <scope> enable --dry-run`
Lockdown	`tacctl emergency lockdown <scope> enable --reason "<text>"`
Lift lockdown	`tacctl emergency lockdown <scope> disable`
Break-glass a device	`tacctl breakglass hold <name> --reason "<text>"`
Release break-glass	`tacctl breakglass release <name>`
Undo a change	`tacctl backup list` then `backup restore <ts>`
Watch logins live	`tacctl log tail -f`
Console settings and each user's shell	`tacctl console show`
System shell for a tier / a user	`tacctl console system-shell <tier> enable` · `console system-shell user <name> enable|disable|clear`
This server at a glance	`tacctl server show`
Server address: show / change	`tacctl server addresses` · `tacctl server addresses replace <ip>`
Name of the server in the console	`tacctl server name [<label>|unset]`
Timing scale for a slow or distant device / scope / server	`tacctl device timeout-scale <name> [<n>|clear]` · `scope timeout-scale <scope> [<n>|clear]` · `config devices timeout-scale [<n>]`
Rebuild, move or repair the server's own login	`tacctl host enroll --local`

See *Exit Codes* for exit statuses.

See also: Access Lifecycle · Engineer Guide: Device Changes · Incident Response Runbooks · CLI reference (repo) · `man tacctl`