Command quick reference

For NEX 1.0 beta 3. The details of each command with `man <command>` and in appendix A of the book. Upper-case words are values to put in their place, square brackets mark optional parts.

For users

Jobs on the nodes · `nex(1)`

Command	What it does
`nex run --node NODE -- COMMAND [ARG...]`	runs it on the node, in the same directory (inside `/nex`)
`nex run --node any -a -- COMMAND`	runs it on the least loaded compute node and prints which one
`nex run --cpu PCT --mem SIZE --env K=V -- COMMAND`	CPU limit (`200` = two CPUs), memory limit, extra variables
`nex run -d [-q] -- COMMAND`	a detached job, which keeps running after you log out; prints `NODE:JOB`
`nex jobs`	the detached jobs of the whole cluster
`nex attach [NODE:]JOB`	reattaches a job; Ctrl-] or Ctrl-C detach without stopping it
`nex stop [NODE:]JOB`	stops a detached job
`nex kill [-SIGNAL] [NODE:]PID...`	signals a process on any node (SIGTERM by default) and waits

Files and snapshots · `nex(1)`

Command	What it does
`nex snapshot [list] [PATH]`	the snapshots of a directory; for a file, which ones contain it
`nex snapshot restore PATH [--from NAME] [--overwrite]`	copies a file or directory back from a snapshot
`nex snapshot create [--name NAME] [PATH]`	a manual snapshot of the directory

Queue and scheduled tasks · `nex(1)`

Command	What it does
`nex submit [-p 0-9] [--cpu PCT] [--mem SIZE] -- COMMAND`	puts a job in the queue: it starts when a node has free CPUs
`nex queue`	the queue: position, `node:job`, outcome
`nex cancel ENTRY...`	removes entries still waiting from the queue
`nex cron add NAME 'SCHEDULE' -- COMMAND`	a scheduled task, with the five `crontab` fields or `@daily`
`nex cron list`	the tasks, with the next and the last run
`nex cron run NAME`	runs a task right away, outside its schedule

Services · `nex(1)`

Command	What it does
`nex service apply FILE`	creates or updates services from a JSON definition, one instance at a time
`nex service list [NAME]`	instances, `node:job`, state, restarts, nodes
`nex service endpoints NAME`	the `node:port` addresses of the healthy instances
`nex service restart NAME`	restarts one instance at a time

Getting programs to talk · `nex(1)`

Command	What it does
`nex kv set KEY VALUE [--ttl D] [--if-absent , --expect V]`	a value shared by all nodes, optionally expiring or conditional
`nex kv get KEY , del , list [PREFIX]`	reads, removes, lists; `--group NAME` or `--cluster` for the shared spaces
`nex secret set NAME`	an encrypted personal secret, read from the terminal, never from the arguments
`nex secret get NAME`	writes it to standard output: `PW=$(nex secret get db)`
`nex lock [--nowait , --timeout S] NAME -- COMMAND`	runs the command while holding a cluster-wide lock
`nex pub TOPIC MESSAGE`	delivers a message to everyone subscribed to the topic
`nex sub TOPIC [--history N] [--count N]`	listens and prints one line per message
`nex mq push QUEUE MESSAGE`	puts a message in a queue
`nex mq pop QUEUE [--wait S]`	takes a message: each one goes to a single consumer

History, logs and alarms · `nex(1)`

Command	What it does
`nex history [--node NODE] [--since 6h] [--what cpu,mem,net]`	the metrics history of each node, up to 30 days
`nex logs [-n LINES] [--node NODE] [-u UNIT] [-g TEXT] [-f]`	the journal of all the nodes, in time order
`nex alarm [list]`	the active alarms and the latest events; exit 0 none, 3 warnings only, 4 critical

Usage and audit · `nex(1)`

Command	What it does
`nex acct [--from YYYY-MM-DD] [--to YYYY-MM-DD]`	your own usage: CPU hours, jobs, peak memory, space
`nex audit [--since 2h] [--type PREFIX] [-n N]`	the audit log: jobs, services, tokens, snapshots

Account and space · `nex(1)`

Command	What it does
`nex passwd`	changes your password on all the nodes; Samba needs it too
`nex quota`	space and quota of your home, CPU and memory limits
`nex token create NAME [--read-only] [--days N]`	an API token; the secret is shown only once
Windows: `\\ADDRESS\NAME`	in File Explorer: your home or a shared folder, with the cluster user and password (`smb.md`)
Mac: Finder → Go → Connect to Server, `smb://ADDRESS/NAME`	the same files from a Mac, with the cluster user and password (`smb.md`)

nexsh, the cluster shell · `nex(1)`

Command	What it does
`on NODE COMMAND`	runs a command on another node, in the current directory (`nexsh(1)`)
`on any COMMAND`	runs it on the least loaded node (`nexsh(1)`)
`cn NODE`	moves the whole shell to another node; `cn` back, `cn -` to the previous one (`nexsh(1)`)
`nodes`	the nodes: state, role (`term` for terminals), load, CPU, memory, jobs (`nexsh(1)`)
`nxtop [-b] [--node NODE]`	like `top` for the whole cluster; `k` stops a process, `q` quits
`nxps [--local]`	the processes of the whole cluster, in the form `node:pid`
`nxfree`	the memory of each node and of the cluster
`nxdf`	the space of `/nex` and of the nodes' disks

For administrators

The commands in this part are given as root, with `sudo`, except where the row says «as a user».

nex-setup and NEX Setup · `nex-setup(8)`

Command	What it does
`nex-setup-gui`	NEX Setup, as a user: creates a cluster or joins one (`nex-setup-gui(1)`)
`nex-setup`	the guided procedure on a terminal; on a node already set up it shows the status
`nex-setup status`	the node configuration, the services and the cluster status
`nex-setup create [--node NAME] [--addr IP] [--roles ROLES] [--disk DIR]`	makes this machine the first node, the admin node
`nex-setup add --from ADMIN_NODE [--roles ROLES]`	adds this machine to the cluster
`nex-setup add --from ADMIN_NODE --password-stdin`	ssh password from stdin: the key is made and copied by itself
`--roles compute`, `storage`, `compute,storage`	runs programs; keeps part of the data; both
`--roles terminal`	light computer or laptop: sees `/nex` and launches on the others, runs nothing for the others
`nex-setup remove --from ADMIN_NODE`	unhooks the node and removes it from the cluster

NEX Live, the USB stick · `nex-live`

Command	What it does
`nex-live status`	as a user: where the data lives (in RAM, on the stick, on a disk), in JSON
`nex-live disks`	as a user: the computer's disks and what is on them, in JSON
`nex-live persist --stick`	keeps node and data in the free space of the stick; then reboot
`nex-live persist --disk sdX --confirm sdX`	keeps node and data on a whole disk, which is erased; then reboot
`nex-live clusters`	as a user: the clusters announced on the local network, in JSON
`nex-live password`	the password of nex on this computer, from standard input
`nex-live cluster-password`	the cluster password for nex, from standard input; turns on ssh
`nex-live secure-sudo`	from now on sudo asks for the password of nex
`nex-live announce`	on the admin node: announces the cluster on the local network

nexctl: nodes and storage · `nexctl(8)`

Command	What it does
`nexctl status`	the node status: whether `/nex` and the binds are hooked up
`nexctl leave [--terminate [--kill]]`	unhooks the node, after moving its services away
`nexctl storage status`	the node among the NEXFS data and meta nodes, and the health
`nexctl storage maintenance-enter`	short stop: copies are not rebuilt; `maintenance-exit` ends it
`nexctl storage drain-check`	says whether the storage node can be removed
`nexctl storage drain`	the final exit: empties the node
`nexctl storage disk-list`	disks: the cluster's, empty, others
`nexctl storage disk-add DISK`	adds an empty disk to this node
`nex service drain NODE`	moves the services off a node with no interruption; `undrain` reopens it (`nex(1)`)

nexctl: users, quotas and Samba · `nexctl(8)`

Command	What it does
`nexctl users list`	the cluster users
`nexctl users add NAME [--groups G,...]`	a cluster user, with the home in `/nex/home`
`nexctl users passwd NAME`	a user's password, asked twice
`nexctl users lock NAME`	locks a user; `unlock` unlocks
`nexctl users group-add GROUP`	creates a cluster group
`nexctl users groups NAME --groups G,...`	the user's groups become these
`nexctl users quota NAME [--space 10G] [--cpu 200] [--mem 8G]`	home quota and per-node CPU and memory limits
`nexctl smb enable`	shares `/nex` with Samba from this node; then each user runs `nex passwd` once
`nexctl smb share-add NAME --path DIR --group GROUP`	a project folder shared with a group

nexfsctl: the NEXFS storage · `nexfsctl(8)`

Command	What it does
`nexfsctl health [--no-traffic]`	the whole group in one answer: `ok`, `warning` or `error`, with the reasons
`nexfsctl space`	total and free space, files, number of copies
`nexfsctl node list`	the data nodes: address, space, state
`nexfsctl node drain ID [--cancel , --dry-run]`	empties a data node, or first says if it can and how much moves
`nexfsctl copies N [PATH]`	the number of copies of the group or of a directory
`nexfsctl disk add DIR`	adds a mounted disk

Updates, virtual addresses, backups · `nex(1)`

Command	What it does
`nex apt status`	pending updates and required reboots, node by node
`nex apt upgrade [--reboot] [--dry-run]`	upgrades the whole system, one node at a time
`nex apt install PACKAGE...`	the same packages on all the compute nodes
`nexctl upgrade --apt [--dry-run]`	upgrades NEX and NEXFS one node at a time (`nexctl(8)`)
`nex vip add NAME IP/PREFIX [--service SVC] [--port P]`	a virtual address that follows a service or a port
`nex vip list`, `remove NAME`	the virtual addresses and who holds them; removes one
`nex backup set backup_target USER@HOST:/PATH`	where to make the backup, with `rsync` over SSH
`nex backup [status]`, `run`	the status of the last backup; a backup right away
`nex report [--since 24h]`	also as a user: the archive to attach to a problem report

The desktop · `nex-desktop(1)`

Command	What it does
`nex-desktop terminal [NODE]`	a graphical terminal with nexsh, or with a shell on the node
`nex-monitor [--node NODE]`	NEX Monitor: nodes, processes, network, disks, history (`nex-monitor(1)`)
The NEX icon in the notification area	starts by itself in XFCE and KDE Plasma: status, nodes, NEX Monitor, terminal (`desktop.md`)

`nex` exit codes: 0 success, 1 error, 2 some node did not answer; with `run` and `attach` the command's own code, 128 + N if killed by signal N, 255 NEX error or *LOST* job.

The guides: in `/usr/share/doc/nex-cluster/` on every node, and on the site nex.aratoingegneria.it.