

NAME

YunoHost – yunohost server administration command

SYNOPSIS

yunohost *CATEGORY COMMAND* [*SUBCOMMAND*] [*ARGUMENTS*]... [*OPTIONS*]...

DESCRIPTION

usage: yunohost {user, domain, app, backup, settings, service, firewall, dyndns, tools, hook, log, diagnosis} ...
[-h|--help] [--no-cache] [--output-as {json, plain, none}] [--debug] [--quiet] [--timeout ==SUP-PRESS==] [--admin-password PASSWORD] [-v|--version]

optional arguments:

-h, --help
show this help message and exit

categories:

{user, domain, app, backup, settings, service, firewall, dyndns, tools, hook, log, diagnosis}

user Manage users and groups

domain Manage domains

app Manage apps

backup Manage backups

settings Manage YunoHost global settings

service Manage services

firewall Manage firewall rules

dyndns Subscribe and Update DynDNS Hosts (deprecated, use 'yunohost domain dyndns' instead)

tools Specific tools

hook Manage hooks

log Manage debug logs

diagnosis
Look for possible issues on the server

global arguments:

- no-cache**
Don't use actions map cache
- output-as** {json,plain,none}
Output result in another format
- debug**
Log and print debug messages
- quiet**
Don't produce any output
- timeout** SECONDS
Number of seconds before this command will timeout because it can't acquire the lock (meaning that another command is currently running), by default there is no timeout and the command will wait until it can get the lock
- admin-password** PASSWORD
The admin password to use to authenticate
- v, --version**
Display YunoHost packages versions

YUNOHOST USER

usage: yunohost user {list,create,delete,update,info,export,import} ...

description:

Manage users and groups

yunohost user list [*--fields* fields]

List users

--fields fields

fields to fetch (username, fullname, mail, mail-alias, mail-forward, mailbox-quota, groups, shell, home-path)

yunohost user create *username* *-F|--fullname* fullname *-p|--password* password [*-d|--domain* domain] [*-q|--mailbox-quota* mailbox-quota] [*-s|--loginShell* loginShell]
Create user

username

The unique username to create

-F, --fullname *fullname*
The full name of the user. For example 'Camille Dupont'

-p, --password *password*
User password

-d, --domain *domain*
Domain for the email address

-q, --mailbox-quota *mailbox-quota* (default: 0)
Mailbox size quota

-s, --loginShell *loginShell* (default: /bin/bash)
The login shell used

yunohost user delete *username* [--purge] [--force]
Delete user

username
Username to delete

--purge
Purge user's home and mail directories

--force Force user deletion

yunohost user update *username* [-F|--fullname *fullname*] [-m|--mail *mail*] [-p|--change-password *change-password*] [--add-mailforward *add-mailforward*] [--remove-mailforward *remove-mailforward*] [--add-mailalias *add-mailalias*] [--remove-mailalias *remove-mailalias*] [-q|--mailbox-quota *mailbox-quota*] [-s|--loginShell *loginShell*]
Update user informations

username
Username to update

-F, --fullname *fullname*
The full name of the user. For example 'Camille Dupont'

-m, --mail *mail*

-p, --change-password *change-password*
New password to set

--add-mailforward *add-mailforward*
Mailforward addresses to add

--remove-mailforward *remove-mailforward*
Mailforward addresses to remove

--add-mailalias *add-mailalias*
Mail aliases to add

--remove-mailalias *remove-mailalias*
Mail aliases to remove

-q, --mailbox-quota *mailbox-quota*
Mailbox size quota

-s, --loginShell *loginShell* (default: /bin/bash)
The login shell used

yunohost user info *username*
Get user information

username
Username or email to get information

yunohost user export
Export users into CSV

yunohost user import *csvfile* [-u|--update] [-d|--delete]
Import several users from CSV

csvfile CSV file with columns username, firstname, lastname, password, mail, mailbox-quota, mail-alias, mail-forward, groups (separated by coma)

-u, --update

Update all existing users contained in the CSV file (by default existing users are ignored)

-d, --delete

Delete all existing users that are not contained in the CSV file (by default existing users are kept)

yunohost user group list [-f|--full] [-p|--include-primary-groups]

List existing groups

-f, --full

Display all informations known about each groups

-p, --include-primary-groups

Also display primary groups (each user has an eponym group that only contains itself)

yunohost user group create *groupname*

Create group

groupname

Name of the group to be created

yunohost user group delete *groupname*

Delete group

groupname

Name of the group to be deleted

yunohost user group info *groupname*

Get information about a specific group

groupname

Name of the group to fetch info about

yunohost user group add *groupname usernames*

Add users to group

groupname

Name of the group to add user(s) to

usernames

User(s) to add in the group

yunohost user group remove *groupname usernames*

Remove users from group

groupname

Name of the group to remove user(s) from

usernames

User(s) to remove from the group

yunohost user group add-mailalias *groupname aliases [--force]*

Add mail aliases to group

groupname

Name of the group to add user(s) to

aliases Mail aliases to add

--force Ignore warnings about special groups

yunohost user group remove-mailalias *groupname aliases [--force]*
Remove mail aliases to group

groupname

Name of the group to add user(s) to

aliases Mail aliases to remove

--force Ignore warnings about special groups

yunohost user permission list *apps [-s|--short] [-f|--full]*
List permissions and corresponding accesses

apps Apps to list permission for (all by default)

-s, --short

Only list permission names

-f, --full

Display all info known about each permission, including the full user list of each group it is granted to.

yunohost user permission info *permission*
Get information about a specific permission

permission

Name of the permission to fetch info about (use "yunohost user permission list" and "yunohost user permission -f" to see all the current permissions)

yunohost user permission update *permission* [-l|--label *label*] [-s|--show_tile *show_tile*]

Manage group or user permissions

permission

Permission to manage (e.g. mail or nextcloud or wordpress.editors) (use "yunohost user permission list" and "yunohost user permission -f" to see all the current permissions)

-l, --label *label*

Label for this permission. This label will be shown on the SSO and in the admin

-s, --show_tile *show_tile*

Define if a tile will be shown in the SSO

yunohost user permission add *permission names*

Grant permission to group or user

permission

Permission to manage (e.g. mail or nextcloud or wordpress.editors) (use "yunohost user permission list" and "yunohost user permission -f" to see all the current permissions)

names Group or usernames to grant this permission to

yunohost user permission remove *permission names*

Revoke permission to group or user

permission

Permission to manage (e.g. mail or nextcloud or wordpress.editors) (use "yunohost user permission list" and "yunohost user permission -f" to see all the current permissions)

names Group or usernames to revoke this permission to

yunohost user permission reset *permission*

Reset allowed groups to the default (all_users) for a given permission

permission

Permission to manage (e.g. mail or nextcloud or wordpress.editors) (use "yunohost user permission list" and "yunohost user permission -f" to see all the current permissions)

yunohost user ssh list-keys *username*

Show user's authorized ssh keys

username

Username of the user

yunohost user ssh add-key *username key [-c|--comment comment]*

Add a new authorized ssh key for this user

username

Username of the user

key The key to be added

-c, --comment *comment*

Optional comment about the key

yunohost user ssh remove-key *username key*
Remove an authorized ssh key for this user

username
Username of the user

key The key to be removed

YUNOHOST DOMAIN

usage: yunohost domain {list,info,add,remove,main-domain,url-available,action-run} ...

description:
Manage domains

yunohost domain list [*--exclude-subdomains*] [*--tree*] [*--features features*]
List domains

--exclude-subdomains
Filter out domains that are obviously subdomains of other declared domains

--tree Display domains as a tree

--features *features*
List only domains with features enabled (mail_in, mail_out)

yunohost domain info *domain*
Get domain aggregated data

domain
Domain to check

yunohost domain add *domain* [--ignore-dyndns] [--dyndns-recovery-password *dyndns-recovery-password*]
[--install-letsencrypt-cert]
Create a custom domain

domain

Domain name to add

--ignore-dyndns

If adding a DynDNS domain, only add the domain, without subscribing to the DynDNS service

--dyndns-recovery-password *dyndns-recovery-password*

If adding a DynDNS domain, subscribe to the DynDNS service with a password, used to later delete the domain

--install-letsencrypt-cert

If adding a subdomain of an already added domain, try to install a Let's Encrypt certificate

yunohost domain remove *domain* [-r|--remove-apps] [-f|--force] [--ignore-dyndns] [--dyndns-recovery-password *dyndns-recovery-password*]
Delete domains

domain

Domain to delete

-r, --remove-apps

Remove apps installed on the domain

-f, --force

Do not ask confirmation to remove apps

--ignore-dyndns

If removing a DynDNS domain, only remove the domain, without unsubscribing from the DynDNS service

--dyndns-recovery-password *dyndns-recovery-password*

If removing a DynDNS domain, unsubscribe from the DynDNS service with a password

yunohost domain main-domain [-n|--new-main-domain new-main-domain]
Check the current main domain, or change it

-n, --new-main-domain *new-main-domain*
Change the current main domain

yunohost domain url-available *domain path*
Check availability of a web path

domain
The domain for the web path (e.g. your.domain.tld)

path The path to check (e.g. /coffee)

yunohost domain action-run *domain action* [-a|--args args]
Run domain action

domain
Domain name

action action id

-a, --args *args*
Serialized arguments for action (i.e. "foo=bar&lorem=ipsum")

yunohost domain dyndns subscribe *domain* [-p|--recovery-password recovery-password]
Subscribe to a DynDNS service

domain

Domain to subscribe to the DynDNS service

-p, --recovery-password *recovery-password*

Password used to later recover the domain if needed

yunohost domain dyndns unsubscribe *domain* [-p|--recovery-password *recovery-password*]

Unsubscribe from a DynDNS service

domain

Domain to unsubscribe from the DynDNS service

-p, --recovery-password *recovery-password*

Recovery password used to delete the domain

yunohost domain dyndns set-recovery-password *domain* [-p|--recovery-password *recovery-password*]

Set recovery password

domain

Domain to set recovery password for

-p, --recovery-password *recovery-password*

The new recovery password

yunohost domain config get *domain key* [-f|--full] [-e|--export]

Display a domain configuration

domain

Domain name

key A specific panel, section or a question identifier

-f, --full
Display all details (meant to be used by the API)

-e, --export
Only export key/values, meant to be reimported using "config set --args-file"

yunohost domain config set *domain key* [-v|--value *value*] [-a|--args *args*]
Apply a new configuration

domain
Domain name

key The question or form key

-v, --value *value*
new value

-a, --args *args*
Serialized arguments for new configuration (i.e. "mail_in=0&mail_out=0")

yunohost domain dns suggest *domain*
Generate sample DNS configuration for a domain

domain
Target domain

yunohost domain dns push *domain* [-d|--dry-run] [--force] [--purge]
Push DNS records to registrar

domain

Domain name to push DNS conf for

-d, --dry-run

Only display what's to be pushed

--force Also update/remove records which were not originally set by Yunohost, or which have been manually modified

--purge

Delete all records

yunohost domain cert status *domain_list* [*--full*]

List status of current certificates (all by default).

domain_list

Domains to check

--full Show more details

yunohost domain cert install *domain_list* [*--force*] [*--no-checks*] [*--self-signed*]

Install Let's Encrypt certificates for given domains (all by default).

domain_list

Domains for which to install the certificates

--force Install even if current certificate is not self-signed

--no-checks

Does not perform any check that your domain seems correctly configured (DNS, reachability) before attempting to install. (Not recommended)

--self-signed

Install self-signed certificate instead of Let's Encrypt

yunohost domain cert renew *domain_list* [--force] [--email] [--no-checks]
Renew the Let's Encrypt certificates for given domains (all by default).

domain_list

Domains for which to renew the certificates

--force Ignore the validity threshold (15 days)

--email Send an email to root with logs if some renewing fails

--no-checks

Does not perform any check that your domain seems correctly configured (DNS, reachability) before attempting to renew. (Not recommended)

YUNOHOST APP

usage: yunohost app {catalog,search,manifest,list,info,map,install,remove,upgrade,change-url,setting,shell,register-url,makedefault,dismiss-notification,ssowatconf,change-label} ...

description:

Manage apps

yunohost app catalog [-f|--full] [-c|--with-categories] [-a|--with-antifeatures]
Show the catalog of installable application

-f, --full

Display all details, including the app manifest and various other infos

-c, --with-categories

Also return a list of app categories

-a, --with-antifeatures

Also return a list of antifeatures categories

yunohost app search *string*
Search installable apps

string Return matching app name or description with "string"

yunohost app manifest *app* [-s|--with-screenshot]
Return the manifest of a given app from the catalog, or from a remote git repo

app Name, local path or git URL of the app to fetch the manifest of

-s, --with-screenshot
Also return a base64 screenshot if any (API only)

yunohost app list [-f|--full] [-u|--upgradable]
List installed apps

-f, --full
Display all details, including the app manifest and various other infos

-u, --upgradable
List only apps that can upgrade to a newer version

yunohost app info *app* [-f|--full]
Show infos about a specific installed app

app Specific app ID

-f, --full

Display all details, including the app manifest and various other infos

yunohost app map [-a|--app *app*] [-r|--raw] [-u|--user *user*]

Show the mapping between urls and apps

-a, --app *app*

Specific app to map

-r, --raw

Return complete dict

-u, --user *user*

Allowed app map for a user

yunohost app install *app* [-l|--label *label*] [-a|--args *args*] [-n|--no-remove-on-failure] [-f|--force]

Install apps

app Name, local path or git URL of the app to install

-l, --label *label*

Custom name for the app

-a, --args *args*

Serialized arguments for app script (i.e. "domain=domain.tld&path=/path&init_main_permission=visitors")

-n, --no-remove-on-failure

Debug option to avoid removing the app on a failed installation

-f, --force

Do not ask confirmation if the app is not safe to use (low quality, experimental or 3rd party), or when the app displays a post-install notification

yunohost app remove *app* [-p|--purge]
Remove app

app App to remove

-p, --purge
Also remove all application data

yunohost app upgrade *app* [-u|--url *url*] [-f|--file *file*] [-F|--force] [-b|--no-safety-backup] [-c|--continue-on-failure]
Upgrade app

app App(s) to upgrade (default all)

-u, --url *url*
Git url to fetch for upgrade

-f, --file *file*
Folder or tarball for upgrade

-F, --force
Force the update, even though the app is up to date

-b, --no-safety-backup
Disable the safety backup during upgrade

-c, --continue-on-failure
Continue to upgrade apps even if one or more upgrade failed

yunohost app change-url *app* -d|--domain *domain* -p|--path *path*
Change app's URL

app Target app instance name

-d, --domain *domain*
New app domain on which the application will be moved

-p, --path *path*
New path at which the application will be moved

yunohost app setting *app key* [-v|--value *value*] [-d|--delete]
Set or get an app setting value

app App ID

key Key to get/set

-v, --value *value*
Value to set

-d, --delete
Delete the key

yunohost app shell *app*
Open an interactive shell with the app environment already loaded

app App ID

yunohost app register-url *app domain path*
Book/register a web path for a given app

app App which will use the web path

domain
The domain on which the app should be registered (e.g. your.domain.tld)

path The path to be registered (e.g. /coffee)

yunohost app makedefault *app* [-d|--domain *domain*] [-u|--undo]
Redirect domain root to an app

app App name to put on domain root

-d, --domain *domain*
Specific domain to put app on (the app domain by default)

-u, --undo
Undo redirection

yunohost app dismiss-notification *app name*
Dismiss post_install or post_upgrade notification

app App ID to dismiss notification for

name Notification name, either post_install or post_upgrade

yunohost app ssoatconf
Regenerate SSOwat configuration file

yunohost app change-label *app new_label*
Change app label

app App ID

new_label
New app label

yunohost app action list *app*
List app actions

app App name

yunohost app action run *app action* [-a|--args *args*]
Run app action

app App name

action action id

-a, --args *args*
Serialized arguments for app script (i.e. "domain=domain.tld&path=/path")

yunohost app config get *app key* [-f|--full] [-e|--export]
Display an app configuration

app App name

key A specific panel, section or a question identifier

-f, --full

Display all details (meant to be used by the API)

-e, --export

Only export key/values, meant to be reimported using "config set --args-file"

yunohost app config set *app key* [-v|--value *value*] [-a|--args *args*] [-f|--args-file *args-file*]

Apply a new configuration

app App name

key The question or panel key

-v, --value *value*
new value

-a, --args *args*
Serialized arguments for new configuration (i.e. "domain=domain.tld&path=/path")

-f, --args-file *args-file*
YAML or JSON file with key/value couples

YUNOHOST BACKUP

usage: yunohost backup { create,restore,list,info,download,delete } ...

description:

Manage backups

yunohost backup create [-n|--name *name*] [-d|--description *description*] [-o|--output-directory *output-directory*]
[-i|--methods *methods*] [--system *system*] [--apps *apps*] [--dry-run]

Create a backup local archive. If neither --apps or --system are given, this will backup all apps and all system parts. If only --apps if given, this will only backup apps and no system parts. Similarly, if only --system is given, this will only backup system parts and no apps.

- n, --name** *name*
Name of the backup archive
- d, --description** *description*
Short description of the backup
- o, --output-directory** *output-directory*
Output directory for the backup
- methods** *methods*
List of backup methods to apply (copy or tar by default)
- system** *system*
List of system parts to backup (or all if none given).
- apps** *apps*
List of application names to backup (or all if none given)
- dry-run**

yunohost backup restore *name* [--system *system*] [--apps *apps*] [--force]

Restore from a local backup archive. If neither --apps or --system are given, this will restore all apps and all system parts in the archive. If only --apps is given, this will only restore apps and no system parts. Similarly, if only --system is given, this will only restore system parts and no apps.

- name** Name or path of the backup archive
- system** *system*
List of system parts to restore (or all if none is given)
- apps** *apps*
List of application names to restore (or all if none is given)
- force** Force restauration on an already installed system

yunohost backup list [-i|--with-info] [-H|--human-readable]

List available local backup archives

-i, --with-info
Show backup information for each archive

-H, --human-readable
Print sizes in human readable format

yunohost backup info *name* [-d|--with-details] [-H|--human-readable]
Show info about a local backup archive

name Name or path of the backup archive

-d, --with-details
Show additional backup information

-H, --human-readable
Print sizes in human readable format

yunohost backup download *name*
(API only) Request to download the file

name Name of the local backup archive

yunohost backup delete *name*
Delete a backup archive

name Name of the archive to delete

YUNOHOST SETTINGS

usage: yunohost settings {list,get,set,reset-all,reset} ...

description:

Manage YunoHost global settings

yunohost settings list [-f|--full]

list all entries of the settings

-f, --full

Display all details (meant to be used by the API)

yunohost settings get *key* [-f|--full] [-e|--export]

get an entry value in the settings

key Settings key

-f, --full

Display all details (meant to be used by the API)

-e, --export

Only export key/values, meant to be reimported using "config set --args-file"

yunohost settings set *key* [-v|--value *value*] [-a|--args *args*]

set an entry value in the settings

key The question or form key

-v, --value *value*

new value

-a, --args *args*

Serialized arguments for new configuration (i.e. "mail_in=0&mail_out=0")

yunohost settings reset-all

reset all settings to their default value

yunohost settings reset *key*

set an entry value to its default one

key Settings key

YUNOHOST SERVICE

usage: yunohost service {add,remove,start,stop,reload,restart,reload_or_restart,enable,disable,status,log} ...

description:

Manage services

yunohost service add *name* [-d|--description *description*] [-l|--log *log*] [--test_status *test_status*] [--test_conf *test_conf*] [--needs_exposed_ports *needs_exposed_ports*] [-n|--need_lock]
Add a service

name Service name to add

-d, --description *description*
Description of the service

-l, --log *log*
Absolute path to log file to display

--test_status *test_status*
Specify a custom bash command to check the status of the service. Note that it only makes sense to specify this if the corresponding systemd service does not return the proper information already.

--test_conf *test_conf*

Specify a custom bash command to check if the configuration of the service is valid or broken, similar to `nginx -t`.

--needs_exposed_ports *needs_exposed_ports*

A list of ports that needs to be publicly exposed for the service to work as intended.

-n, --need_lock

Use this option to prevent deadlocks if the service does invoke yunohost commands.

yunohost service remove *name*

Remove a service

name Service name to remove

yunohost service start *names*

Start one or more services

names Service name to start

yunohost service stop *names*

Stop one or more services

names Service name to stop

yunohost service reload *names*

Reload one or more services

names Service name to reload

yunohost service restart *names*

Restart one or more services. If the services are not running yet, they will be started.

names Service name to restart

yunohost service reload_or_restart *names*

Reload one or more services if they support it. If not, restart them instead. If the services are not running yet, they will be started.

names Service name to reload or restart

yunohost service enable *names*

Enable one or more services

names Service name to enable

yunohost service disable *names*

Disable one or more services

names Service name to disable

yunohost service status *names*

Show status information about one or more services (all by default)

names Service name to show

yunohost service log *name* [-n|--number *number*]

Log every log files of a service

name Service name to log

-n, --number *number* (default: 50)
Number of lines to display

YUNOHOST FIREWALL

usage: yunohost firewall {list,allow,disallow,upnp,reload,stop} ...

description:

Manage firewall rules

yunohost firewall list [-r|--raw] [-i|--by-ip-version] [-f|--list-forwarded]

List all firewall rules

-r, --raw
Return the complete YAML dict

-i, --by-ip-version
List rules by IP version

-f, --list-forwarded

List forwarded ports with UPnP

yunohost firewall allow *protocol port* [-4|--ipv4-only] [-6|--ipv6-only] [--no-upnp] [--no-reload]
Allow connections on a port

protocol

Protocol type to allow (TCP/UDP/Both)

port Port or range of ports to open

-4, --ipv4-only

Only add a rule for IPv4 connections

-6, --ipv6-only

Only add a rule for IPv6 connections

--no-upnp

Do not add forwarding of this port with UPnP

--no-reload

Do not reload firewall rules

yunohost firewall disallow *protocol port* [-4|--ipv4-only] [-6|--ipv6-only] [--upnp-only] [--no-reload]
Disallow connections on a port

protocol

Protocol type to allow (TCP/UDP/Both)

port Port or range of ports to close

-4, --ipv4-only

Only remove the rule for IPv4 connections

-6, --ipv6-only

Only remove the rule for IPv6 connections

--upnp-only

Only remove forwarding of this port with UPnP

--no-reload

Do not reload firewall rules

yunohost firewall upnp *action* [*--no-refresh*]

Manage port forwarding using UPnP

action**--no-refresh**

Do not refresh port forwarding

yunohost firewall reload [*--skip-upnp*]

Reload all firewall rules

--skip-upnp

Do not refresh port forwarding using UPnP

yunohost firewall stop

Stop iptables and ip6tables

YUNOHOST DYNDNS

usage: yunohost dyndns {subscribe,update} ...

description:

Subscribe and Update DynDNS Hosts (deprecated, use 'yunohost domain dyndns' instead)

yunohost dyndns subscribe [-d|--domain *domain*] [-p|--recovery-password *recovery-password*]

Subscribe to a DynDNS service

-d, --domain *domain*

Full domain to subscribe with (deprecated, use 'yunohost domain dyndns subscribe' instead)

-p, --recovery-password *recovery-password*

Password used to later recover the domain if needed

yunohost dyndns update [-d|--domain *domain*] [-f|--force] [-D|--dry-run]

Update IP on DynDNS platform

-d, --domain *domain*

Full domain to update

-f, --force

Force the update (for debugging only)

-D, --dry-run

Only display the generated zone

YUNOHOST TOOLS

usage: yunohost tools {rootpw,maindomain,postinstall,update,upgrade,shell,basic-space-cleanup,shutdown,reboot,regen-conf,versions} ...

description:

Specific tools

yunohost tools rootpw *-n|--new-password new-password*
Change root password

-n, --new-password *new-password*

yunohost tools maindomain [*-n|--new-main-domain new-main-domain*]
Check the current main domain, or change it

-n, --new-main-domain *new-main-domain*
Change the current main domain

yunohost tools postinstall *-d|--domain domain -u|--username username -F|--fullname fullname -p|--password password [--ignore-dyndns] [--dyndns-recovery-password dyndns-recovery-password] [--force-diskspace] [--i-have-read-terms-of-services]*
YunoHost post-install

-d, --domain *domain*
YunoHost main domain

-u, --username *username*
Username for the first (admin) user. For example 'camille'

-F, --fullname *fullname*
The full name for the first (admin) user. For example 'Camille Dupont'

-p, --password *password*
YunoHost admin password

--ignore-dyndns
If adding a DynDNS domain, only add the domain, without subscribing to the DynDNS service

--dyndns-recovery-password *dyndns-recovery-password*
If adding a DynDNS domain, subscribe to the DynDNS service with a password, used to later recover the domain if needed

--force-diskspace

Use this if you really want to install YunoHost on a setup with less than 10 GB on the root filesystem

--i-have-read-terms-of-services

Automatically reply to the terms of services prompt, for example for non-interactive installations

yunohost tools update *target*

YunoHost update

target What to update, "apps" (application catalog) or "system" (fetch available package upgrades, equivalent to apt update), "all" for both

yunohost tools upgrade *target*

YunoHost upgrade

target What to upgrade, either "apps" (all apps) or "system" (all system packages)

yunohost tools shell [-c|--command *command*]

Launch a development shell

-c, --command *command*

python command to execute

yunohost tools basic-space-cleanup

Basic space cleanup (apt, journalctl, logs, ...)

yunohost tools shutdown [-f|--force]
Shutdown the server

-f, --force
skip the shutdown confirmation

yunohost tools reboot [-f|--force]
Reboot the server

-f, --force
skip the reboot confirmation

yunohost tools regen-conf *names* [-d|--with-diff] [-f|--force] [-n|--dry-run] [-p|--list-pending]
Regenerate the configuration file(s)

names Categories to regenerate configuration of (all by default)

-d, --with-diff
Show differences in case of configuration changes

-f, --force
Override all manual modifications in configuration files

-n, --dry-run
Show what would have been regenerated

-p, --list-pending
List pending configuration files and exit

yunohost tools versions
Display YunoHost's packages versions

yunohost tools migrations list [*--pending*] [*--done*]
List migrations

--pending
list only pending migrations

--done list only migrations already performed

yunohost tools migrations run *targets* [*--skip*] [*--force-rerun*] [*--auto*] [*--accept-disclaimer*]
Run migrations

targets Migrations to run (all pendings by default)

--skip Skip specified migrations (to be used only if you know what you are doing)

--force-rerun
Re-run already-ran specified migration (to be used only if you know what you are doing)

--auto Automatic mode, won't run manual migrations (to be used only if you know what you are doing)

--accept-disclaimer
Accept disclaimers of migrations (please read them before using this option)

yunohost tools migrations state
Show current migrations state

YUNOHOST HOOK

usage: yunohost hook {add,remove,info,list,callback,exec} ...

description:
Manage hooks

yunohost hook add *app file*
Store hook script to filesystem

app App to link with

file Script to add

yunohost hook remove *app*
Remove hook scripts from filesystem

app Scripts related to app will be removed

yunohost hook info *action name*
Get information about a given hook

action Action name

name Hook name

yunohost hook list *action* [-l|--list-by list-by] [-i|--show-info]
List available hooks for an action

action Action name

-l, --list-by *list-by* (default: name)
Property to list hook by

-i, --show-info
Show hook information

yunohost hook callback *action* [-n|--hooks *hooks*] [-a|--args *args*] [-d|--chdir *chdir*]
Execute all scripts binded to an action

action Action name

-n, --hooks *hooks*
List of hooks names to execute

-a, --args *args*
Ordered list of arguments to pass to the scripts

-d, --chdir *chdir*
The directory from where the scripts will be executed

yunohost hook exec *path* [-a|--args *args*] [--raise-on-error] [-d|--chdir *chdir*]
Execute hook from a file with arguments

path Path of the script to execute

-a, --args *args*
Ordered list of arguments to pass to the script

--raise-on-error
Raise if the script returns a non-zero exit code

-d, --chdir *chdir*
The directory from where the script will be executed

YUNOHOST LOG

usage: yunohost log {list,show,share} ...

description:

Manage debug logs

yunohost log list [-l|--limit *limit*] [-d|--with-details] [-s|--with-suboperations]

List logs

-l, --limit *limit* (default: 50)

Maximum number of operations to list (default to 50)

-d, --with-details

Show additional infos (e.g. operation success) but may significantly increase command time. Consider using --limit in combination with this.

-s, --with-suboperations

Include metadata about operations that are not the main operation but are sub-operations triggered by another ongoing operation... (e.g. initializing groups/permissions when installing an app)

yunohost log show *path* [-n|--number *number*] [--share] [-i|--filter-irrelevant] [-s|--with-suboperations]

Display a log content

path Log file which to display the content. 'last' or 'last-X' selects the last but X log file

-n, --number *number* (default: 50)

Number of lines to display

--share (Deprecated, see yunohost log share) Share the full log using yunopaste

-i, --filter-irrelevant

Do not show some lines deemed not relevant (like set +x or helper argument parsing)

-s, --with-suboperations

Include metadata about sub-operations of this operation... (e.g. initializing groups/permissions when installing an app)

yunohost log share *path*

Share the full log on yunopaste (alias to show --share)

path Log file to share

YUNOHOST DIAGNOSIS

usage: yunohost diagnosis {list,show,get,run,ignore,unignore} ...

description:

Look for possible issues on the server

yunohost diagnosis list

List diagnosis categories

yunohost diagnosis show *categories* [--full] [--issues] [--share] [--human-readable]

Show most recent diagnosis results

categories

Diagnosis categories to display (all by default)

--full Display additional information

--issues

Only display issues

--share Share the logs using yunopaste

--human-readable

Show a human-readable output

yunohost diagnosis get *category item*

Low-level command to fetch raw data and status about a specific diagnosis test

category

Diagnosis category to fetch results from

item List of criteria describing the test. Must correspond exactly to the 'meta' infos in 'yunohost diagnosis show'

yunohost diagnosis run *categories* [*--force*] [*--except-if-never-ran-yet*] [*--email*]

Run diagnosis

categories

Diagnosis categories to run (all by default)

--force Ignore the cached report even if it is still 'fresh'

--except-if-never-ran-yet

Don't run anything if diagnosis never ran yet ... (this is meant to be used by the webadmin)

--email Send an email to root with issues found (this is meant to be used by cron job)

yunohost diagnosis ignore [*--filter filter*] [*--list*]

Configure some diagnosis results to be ignored and therefore not considered as actual issues

--filter *filter*

Add a filter. The first element should be a diagnosis category, and other criterias can be provided using the infos from the 'meta' sections in 'yunohost diagnosis show'. For example: 'dnsrecords domain=yolo.test category=mail'

--list List active ignore filters

yunohost diagnosis unignore [*--filter filter*]

Configure some diagnosis results to be unignored and therefore considered as actual issues

--filter *filter*

Remove a filter (it should be an existing filter as listed with "ignore --list")