Helm Kubernetes package manager

CONCEPTS

A chart is the package, a release is one installation of it. Each upgrade creates a new numbered revision.

Repository chart index
↓ `helm pull / install`
Chart + values templates
↓ `helm template`
Manifests rendered YAML
↓ `helm install`
Release revisions in cluster

Helm uses your current kubectl context. Add -n <namespace> to every command, or releases land in the default namespace.

Docs now show Helm 4: --atomic became --rollback-on-failure. Helm 3 still uses --atomic.

REPOSITORIES

`helm repo add <name> <url>`
Add a chart repository

`helm repo update`
Refresh local chart indexes

`helm repo list`
List configured repositories

`helm search repo <keyword>`
Search added repositories

`helm search repo <chart> --versions`
List all versions of a chart

`helm search hub <keyword>`
Search Artifact Hub

`helm repo remove <name>`
Remove a repository

`helm pull <chart> --untar`
Download and unpack a chart

INSPECT CHARTS

`helm show values <chart>`
Print the default values

`helm show readme <chart>`
Show the chart README

INSTALL

`helm install <release> <chart>`
Install a chart as a release

INSTALL (cont.)

`helm install <release> <chart> -n <ns> --create-namespace`
Install into a new namespace

`helm install <chart> --generate-name`
Let Helm pick the release name

`helm install <release> <chart> -f <values.yaml>`
Install with a values file

`helm install <release> <chart> --set <key>=<val>`
Override a single value

`helm install <release> <chart> --version <ver>`
Pin the chart version

`helm install <release> <chart> --wait --timeout 5m`
Wait until resources are ready

`helm install <release> <chart> --dry-run=client`
Simulate without a cluster

`helm install <release> ./<dir>`
Install a local chart directory

RELEASES

`helm list`
List releases in this namespace

`helm list -A`
List releases in all namespaces

`helm status <release>`
Show release status and notes

`helm get values <release>`
Values you supplied

`helm get values <release> --all`
Values including chart defaults

`helm get manifest <release>`
The manifests as deployed

`helm history <release>`
List revisions of a release

UPGRADE & ROLLBACK

`helm upgrade <release> <chart> -f <values.yaml>`
Upgrade with a values file

`helm upgrade --install <release> <chart>`
Install or upgrade in one command

UPGRADE & ROLLBACK (cont.)

`helm upgrade <release> <chart> --set <key>=<val>`
Change one value

`helm upgrade <release> <chart> --reuse-values`
Keep previous values, merge new

`helm upgrade <release> <chart> --dry-run=server`
Check an upgrade against the cluster

`helm rollback <release>`
Roll back to the previous revision

`helm rollback <release> <revision>`
Roll back to a given revision

`helm uninstall <release>`
Remove a release

TEMPLATING & DEBUG

`helm template <release> <chart> -f <values.yaml>`
Render manifests locally

`helm template <release> <chart> -s templates/<file>`
Render a single template

`helm template <release> <chart> --debug`
Verbose output while rendering

`helm lint <chart-dir>`
Check a chart for problems

`helm create <name>`
Scaffold a new chart

`helm package <chart-dir>`
Build a chart archive (.tgz)

`helm dependency update <chart-dir>`
Fetch dependencies into charts/

Values precedence (low to high)

Source	Wins over
`chart values.yaml`	nothing
`-f file`	chart defaults
`later -f file`	earlier -f
`--set`	all files

HELP

`helm <command> --help`
Help for a command

`helm version`
Print the client version