k9s Kubernetes terminal UI

START HERE

k9s
Start in current context

s
Shell into container

:pod
Switch to a resource view

d
Describe

/ <text>
Filter the table

?
Show all key hints

l
Logs of selection

Esc
Back / leave mode

Command :ns, :deploy → pick **Table** live resource list → filter / act **Action** logs, shell, edit

CONCEPTS

k9s is a live table of one resource type at a time. Type : plus a name to change type, / to filter, a key to act on the highlighted row.

Start with --readonly on production clusters: it disables every modifying command (edit, delete, restart).

Shortcuts are context sensitive: u, r, t and w mean different things in the Node, Deployment, CronJob and Log views. Press ? to see the current ones.

LAUNCH

k9s
Start in the current kube context

k9s -n <namespace>
Start in a namespace

k9s -A
Start in all namespaces

k9s -c <resource>
Start in a given view, e.g. pod

k9s --context <name>
Use a non-default kube context

k9s --readonly
Disable all modifying commands

k9s info
Show config, log and dump paths

k9s help
List all CLI options

GENERAL

? Show key hints for this view

Ctrl + **A**
List all resource aliases

Esc
Go back / leave command or filter

:q / **Ctrl** + **C** Quit k9s

- Back to the last command
Works like cd - between views.

[/ **]**
Command history back / forward

Ctrl + **R** Refresh the view

Ctrl + **W**
Toggle wide columns

Ctrl + **Z** Toggle faults only

GENERAL (cont.)

Ctrl + **E** Toggle header

Ctrl + **G** Toggle breadcrumbs

SWITCH VIEWS

:<resource>
Open a resource view (:pod, :svc, :deploy)
Singular, plural, short name or alias all work.

:<resource> <namespace>
Open a view in a namespace

:pod /<text>
Pods filtered by text
Needs k9s 0.30.0 or newer.

:pod <key>=<val>,...
Pods matching labels
Needs k9s 0.30.0 or newer.

:pod @<context>
Pods in another context
Switches the current context too. 0.30.0+.

:ns
List and switch namespaces

:ctx
List and switch contexts

:ctx <name>
Switch straight to a context

:pulses / :pu
Cluster pulses overview

:xray <res> [ns]
Dependency tree (po, svc, dp, rs, sts, ds)

:popeye / :pop
Run the Popeye cluster sanitizer

:screendump / :sd
Browse saved screen dumps

FILTER

/<regex>
Filter rows by regex, e.g. web|api

/! <regex>
Inverse filter, hide matches

/-l <selector>
Filter by label selector

/-f <text>
Fuzzy find

INSPECT AND ACT

d Describe the resource

y View YAML

l View logs

INSPECT AND ACT (cont.)

p Previous logs

s Shell into a container (pods)

a Attach to a container (pods)

e Edit the resource
Not available with --readonly.

Ctrl + **D**
Delete, with confirmation
Press Tab to reach OK, then Enter.

Ctrl + **K**
Kill now, no confirmation
Same as kubectl delete --now.

c Copy resource name

n Copy namespace

Shift + **J** Jump to the owner

z
Show ReplicaSets (Deployment view)

SELECT AND SORT

Space Mark a row

Ctrl + **Space** Mark a range

Ctrl + **** Clear all marks

Ctrl + **S**
Save the table to a file

Shift + **N** Sort by name

Shift + **A** Sort by age

Shift + **S** Sort by status

Shift + **P**
Sort by namespace (all-namespaces view)

Shift + **O**
Sort by selected column

Shift + **Left** / **Shift** + **Right**
Move the selected column

LOGS AND WORKLOADS

w Toggle line wrap (log view)

t
Toggle timestamps (log view)

f
Toggle fullscreen (log, YAML, details)

Shift + **F**
Port-forward (pod, service, container)

LOGS AND WORKLOADS (cont.)

f List port-forwards

b
Benchmark a service or port-forward

r
Restart (Deployment, DaemonSet, StatefulSet)

Ctrl + **L**
Roll back a ReplicaSet

t Trigger a CronJob

u Cordon / uncordon a node

r Drain a node

CONFIG

Files in the config directory

File	Purpose
`config.yaml`	Main settings
`hotkeys.yaml`	Custom shortcuts
`plugins.yaml`	External commands
`aliases.yaml`	Extra command aliases
`views.yaml`	Custom columns
`jumps.yaml`	Custom jump targets
`skins/`	Colour themes

Config lives in ~/.config/k9s on Linux, ~/Library/Application Support/k9s on macOS. Logs and screen dumps go to the XDG state dir. Run k9s info for the real paths, or set K9S_CONFIG_DIR.

Skins go in the skins/ folder of the config directory (older versions: the data directory); pick one in config.yaml or set K9S_SKIN.

RECIPES

Safe look at production

```
k9s --readonly --context <prod> -n <app>
```

Find a crashing pod and read its logs

```
# :pod  then  Ctrl+Z (faults only)
# move to the pod, press l
# crashed already? press p for previous logs
```

Restart a deployment

```
:deploy
# select the row, press r
```

Pods of one app across contexts

```
:pod app=<name> @<ctx>
# or filter with  /-l app=<name>
```