# kubectx & kubens Switch context and namespace

**START HERE**

**kubectx**
List or pick a context

`kubectx -c`
Show current context

`kubectx <name>`
Switch cluster

`kubectx <new>=<old>`
Rename a context

`kubectx -`
Back to previous context

**kubens**
List or pick a namespace

**kubens** *<name>*
Switch namespace

**kubens -**
Back to previous namespace

## CONCEPTS

Both tools only edit your kubeconfig: context picks the cluster and user, namespace is the default for kubectl in that context.

**kubeconfig** ~/.kube/config or $KUBECONFIG
**Context (kubectx)** cluster + user
**Namespace (kubens)** default for kubectl

What each tool changes

With fzf installed, running either tool with no argument opens a fuzzy menu instead of printing a list.

## CONTEXTS (KUBECTX)

**kubectx**
List contexts, or pick with fzf

`kubectx <name>`
Switch to a context

`kubectx -`
Switch to the previous context

`kubectx -c`
Show the current context

`kubectx <new>=<name>`
Rename a context

## CONTEXTS (KUBECTX) (cont.)

`kubectx <new>=.`
Rename the current context

`kubectx -d <name> [<name>...]`
Delete one or more contexts
*Only the context entry goes; the cluster and user entries it referenced stay in the kubeconfig.*

`kubectx -u`
Unset the current context

`kubectx --version`
Print the version

## NAMESPACES (KUBENS)

**kubens**
List namespaces, or pick with fzf

**kubens** *<name>*
Switch the active namespace

**kubens -**
Switch to the previous namespace

**kubens -c**
Show the current namespace

**kubens** *<name>* **-f**
Switch even if it does not exist yet

## NAMESPACES (KUBENS) (cont.)

**kubens -u**
Unset namespace, back to default

**kubens --version**
Print the version

## ISOLATED SHELLS

`kubectx -s <name>`
Shell scoped to one context
*Switching context inside this shell does not touch other terminals; exit returns you to where you were.*

`kubectx -s`
Pick the context for a scoped shell

`kubectx -r <name>`
Read-only shell for a context
*Write operations are blocked, which makes it a safer way to look around production.*

`kubectx -r`
Pick the context for a read-only shell

Plain kubectx changes the kubeconfig for every terminal at once. Use -s when you work on two clusters side by side.

## INTERACTIVE MODE

`kubectx | cat`
Plain list even with fzf installed

**export KUBECTX_IGNORE_FZF=1**
Turn off the fzf menu

**export NO_COLOR=1**
Turn off coloured output

**export KUBECTX_CURRENT_FGCOLOR=$(tput setaf 6)**
Text colour of the current item

**export KUBECTX_CURRENT_BGCOLOR=$(tput setab 7)**
Background colour of the current item

## INSTALL AND SETUP

**brew install kubectx**
Install with Homebrew

**sudo apt install kubectx**
Install on Debian or Ubuntu

**kubectl krew install ctx ns**
Install as kubectl plugins
*They then run as kubectl ctx and kubectl ns.*

## INSTALL AND SETUP (cont.)

**sudo pacman -S kubectx**
Install on Arch

**winget install --id ahmetb.kubectx**
Install on Windows (kubens: ahmetb.kubens)

**alias kx=kubectx kn=kubens**
Shorter names

Completion for bash, zsh and fish ships in the repo's completion/ directory: symlink the files into your completion path.

## MULTIPLE KUBECONFIGS

**export KUBECONFIG=~/.kube/a.yaml:~/.kube/b.yaml**
Merge several files into one view
*kubectx then lists the contexts of all files; a switch is written to the first file.*

**KUBECONFIG=***<file>* **kubectx**
Use one file for a single call

**kubectl config view --flatten**
Merge all files into one document

## RECIPES

**Hop between two clusters**

```
kubectx prod-eu
kubens payments
kubectx -   # back to the previous cluster
kubectx -   # and forth again
```

**Short context names**

```
kubectx -c
kubectx prod=gke_acme_europe-west1_prod-cluster
kubectx prod
```

**Safe look at production**

```
kubectx -r prod
kubectl get pods -A   # reads work
kubectl delete pod <name>   # blocked
```

**Two clusters in two terminals**

```
# terminal 1
kubectx -s staging
# terminal 2
kubectx -s prod
```