

Kubernetes interaction for dog lovers – k9s



Arka Bhattacharya



Principal Solutions Architect @ solo.io




@findarka



[linkedin.com/in/findarka](https://www.linkedin.com/in/findarka)

It all started with *kubectl get pods*

 **kubernetes**

DocumentationKubernetes BlogTrainingPartnersCommunityCase StudiesVersionsEnglish

Search

Home

Getting started

Concepts

Tasks

Tutorials

Reference

- Glossary
 - API Overview
 - API Access Control
 - Well-Known Labels, Annotations and Taints
 - Kubernetes API
 - Instrumentation
 - Kubernetes Issues and Security
 - Node Reference Information
 - Networking Reference
 - Setup tools
 - Command line tool (kubectl)

Viewing and finding resources

Get commands with basic output

kubectl get services

kubectl get pods --all-namespaces

kubectl get pods -o wide

kubectl get deployment my-dep

kubectl get pods

kubectl get pod my-pod -o yaml

List all services in the namespace

List all pods in all namespaces

List all pods in the current namespace, with more details

List a particular deployment

List all pods in the namespace

Get a pod's YAML

Describe commands with verbose output

kubectl describe nodes my-node

kubectl describe pods my-pod

List Services Sorted by Name

kubectl get services --sort-by=.metadata.name

List pods Sorted by Restart Count

kubectl get pods --sort-by='.status.containerStatuses[0].restartCount'

List PersistentVolumes sorted by capacity

kubectl get pv --sort-by=.spec.capacity.storage

Get the version label of all pods with label app=cassandra

kubectl get pods --selector=app=cassandra -o \

jsonpath='{.items[*].metadata.labels.version}'

Retrieve the value of a key with dots, e.g. 'ca.crt'

kubectl get configmap myconfig \

-o jsonpath='{.data.ca\.crt}'

Retrieve a base64 encoded value with dashes instead of underscores.

kubectl get secret my-secret --template='{{index .data "key-name-with-dashes"}}'

Edit this page

Create child page

Create an issue

Print entire section

Kubectl autocomplete

BASH

ZSH

A note on --all-namespaces

Kubectl context and configuration

Kubectl apply

Creating objects

Viewing and finding resources

Updating resources

Patching resources

Editing resources

Scaling resources


Deleting resources

Interacting with running Pods

Copying files and directories to and from containers

Interacting with Deployments and Services

Interacting with Nodes and



#osummit

Thank you, aliases and plugins!

github.com/ohmyzsh/ohmyzsh/tree/master/plugins/kubectl#aliases

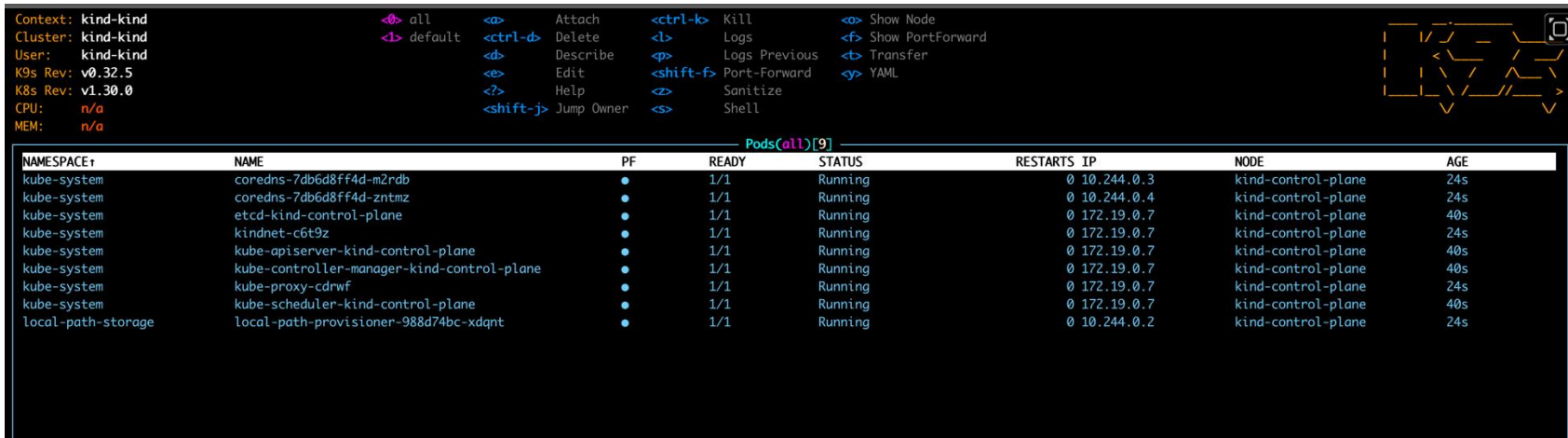
README.md

Aliases

Alias	Command	Description
k	<code>kubectl</code>	The kubectl command
kca	<code>kubectl --all-namespaces</code>	The kubectl command targeting all namespaces
kaf	<code>kubectl apply -f</code>	Apply a YML file
keti	<code>kubectl exec -ti</code>	Drop into an interactive terminal on a container
		Manage configuration quickly to switch contexts between local, dev and staging
kcuc	<code>kubectl config use-context</code>	Set the current-context in a kubeconfig file
kcsc	<code>kubectl config set-context</code>	Set a context entry in kubeconfig
kcdc	<code>kubectl config delete-context</code>	Delete the specified context from the kubeconfig
kccc	<code>kubectl config current-context</code>	Display the current-context
kcgC	<code>kubectl config get-contexts</code>	List of contexts available
		General aliases
kdel	<code>kubectl delete</code>	Delete resources by filenames, stdin, resources and names, or by resources and label selector
kdelf	<code>kubectl delete -f</code>	Delete a pod using the type and name specified in -f argument

[kubectl zsh plugin](#)

Oh.. My.. k9s!



The screenshot displays the k9s terminal interface. At the top, it shows context information: Context: kind-kind, Cluster: kind-kind, User: kind-kind, K9s Rev: v0.32.5, K8s Rev: v1.30.0, CPU: n/a, and MEM: n/a. A menu of keyboard shortcuts is visible, including actions like Attach, Delete, Describe, Edit, Help, Jump Owner, Kill, Logs, Logs Previous, Port-Forward, Sanitize, Shell, Show Node, Show PortForward, Transfer, and YAML. A diagram of a Kubernetes cluster topology is shown in the top right corner.

Below the menu, a table titled "Pods(all)[9]" lists the running pods. The table has columns for Namespace, Name, Phase (PF), Ready status, Status, Restarts, IP, Node, and Age.

Namespace	Name	PF	Ready	Status	Restarts	IP	Node	Age
kube-system	coredns-7db6d8ff4d-m2rdb	●	1/1	Running	0	10.244.0.3	kind-control-plane	24s
kube-system	coredns-7db6d8ff4d-zntmz	●	1/1	Running	0	10.244.0.4	kind-control-plane	24s
kube-system	etcd-kind-control-plane	●	1/1	Running	0	172.19.0.7	kind-control-plane	40s
kube-system	kindnet-c6t9z	●	1/1	Running	0	172.19.0.7	kind-control-plane	24s
kube-system	kube-apiserver-kind-control-plane	●	1/1	Running	0	172.19.0.7	kind-control-plane	40s
kube-system	kube-controller-manager-kind-control-plane	●	1/1	Running	0	172.19.0.7	kind-control-plane	40s
kube-system	kube-proxy-cdrwf	●	1/1	Running	0	172.19.0.7	kind-control-plane	24s
kube-system	kube-scheduler-kind-control-plane	●	1/1	Running	0	172.19.0.7	kind-control-plane	40s
local-path-storage	local-path-provisioner-988d74bc-xdant	●	1/1	Running	0	10.244.0.2	kind-control-plane	24s

"K9s is a terminal based UI to interact with your Kubernetes clusters"

k9scli.io

github.com/derailed/k9s



Let's see it in action!

k9s cheat sheet

- `esc : resource-name = kubectl get resource-name`
- `esc : po = kubectl get pods`
 - `0` = all namespaces
 - `return` or `enter` = see the container/s within a pod
 - `s` = exec into a container
 - `l` = **log**
 - `f` = **full** screen
 - `w` = **w**rap
- `shift + f` = port-forward
- `y` = see the object **y**aml
- `ctrl + d` = **d**delete an object
- `esc : sec = kubectl get secrets`
 - `x` = decode a secret
- `esc : ns = kubectl get namespace`
- `esc : deploy = kubectl get deployment`
- `esc /` = To filter rows and search for an item in the table
- `up` or `down` arrow + `Spacebar` = Select/de-select multiple objects
- `s` = **s**cale Deployment, ReplicaSet etc
- `esc : ctx = kubectl config get-contexts` - select a context and hit return to start using that context

Thank You!



@findarka



linkedin.com/in/findarka