



# Kubernetes cluster interaction simplified with k9s

**Arka Bhattacharya**

#ossummit @findarka





Arka Bhattacharya



Senior Consultant @ Solo.io



@findarka



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

# It all started with *kubectl get pods*

The screenshot shows the 'kubectl Cheat Sheet' page on the Kubernetes documentation website. The page title is 'kubectl Cheat Sheet' and it includes a sub-header 'Kubernetes Documentation / Reference / Command line tool (kubectl) / kubectl Cheat Sheet'. The main content area is titled 'kubectl Cheat Sheet' and states 'This page contains a list of commonly used kubectl commands and flags.' Below this, there is a section for 'Kubectl autocomplete' and 'BASH'. The 'BASH' section includes a code block with the following content: 

```
source <(kubectl completion bash) # set up autocomplete in bash into the current shell, bash-completion package
echo "source <(kubectl completion bash)" >> ~/.bashrc # add autocomplete permanently to your bash shell.
```

 Below the code block, there is a note: 'You can also use a shorthand alias for kubectl that also works with completion:'. Below this, there is a code block with the following content: 

```
alias k=kubectl
complete -o default -F __start_kubectl k
```

 The 'ZSH' section includes a code block with the following content: 

```
source <(kubectl completion zsh) # set up autocomplete in zsh into the current shell
echo '[[ $commands[kubectl] ]] && source <(kubectl completion zsh)' >> ~/.zshrc # add autocomplete permanent
```

 The right sidebar contains links to 'Edit this page', 'Create child page', 'Create an issue', and 'Print entire section'. The left sidebar contains a search bar and a list of navigation links: 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, and Command line tool (kubectl).

The screenshot shows the 'Viewing and finding resources' page on the Kubernetes documentation website. The page title is 'Viewing and finding resources' and it includes a sub-header 'Documentation / Kubernetes Blog / Training / Partners / Community / Case Studies / Versions / English'. The main content area is titled 'Viewing and finding resources' and contains two columns of code blocks. The left column contains the following content: 

```
# 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
```

 The right column contains the following content: 

```
# 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
```

 Below the code blocks, there is a section for 'Describe commands with verbose output' and 'List Services Sorted by Name'. The 'List Services Sorted by Name' section includes a code block with the following content: 

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

 The 'List Pods Sorted by Restart Count' section includes a code block with the following content: 

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

 The 'List PersistentVolumes sorted by capacity' section includes a code block with the following content: 

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

 The 'Get the version label of all pods with label app=cassandra' section includes a code block with the following content: 

```
kubectl get pods --selector=app=cassandra -o \
jsonpath='{.items[*].metadata.labels.version}'
```

 The 'Retrieve the value of a key with dots, e.g. 'ca.crt'' section includes a code block with the following content: 

```
kubectl get configmap myconfig \
-o jsonpath='{.data.ca.crt}'
```

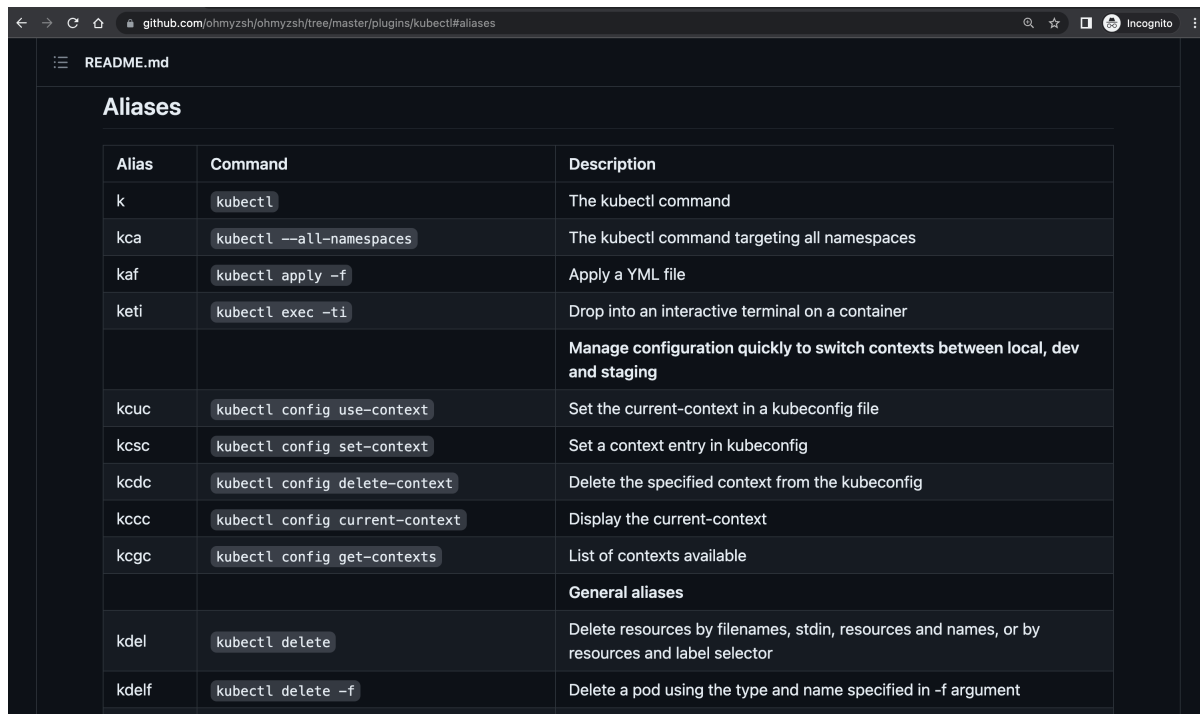
 The 'Retrieve a base64 encoded value with dashes instead of underscores' section includes a code block with the following content: 

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

 The right sidebar contains links to 'Edit this page', 'Create child page', 'Create an issue', and 'Print entire section'. The left sidebar contains a search bar and a list of navigation links: 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, and Command line tool (kubectl).

[kubectl Cheat Sheet link](#)

# Thank you, aliases and plugins!



The screenshot shows a web browser displaying a GitHub repository page for 'ohmyzsh/ohmyzsh/tree/master/plugins/kubectl#aliases'. The page title is 'Aliases'. Below the title is a table with three columns: 'Alias', 'Command', and 'Description'. The table lists various kubectl aliases for common tasks like switching contexts, deleting resources, and applying configurations. A section header 'General aliases' is also present.

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 YAML file
keti	<code>kubectl exec -ti</code>	Drop into an interactive terminal on a container
		<b>Manage configuration quickly to switch contexts between local, dev and staging</b>
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
kcg	<code>kubectl config get-contexts</code>	List of contexts available
		<b>General aliases</b>
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!

```
Context: minikube
Cluster: minikube
User: minikube
K9s Rev: v0.27.3
K8s Rev: v1.26.3
CPU: n/a
MEM: n/a
```

<0>	all	<a>	Attach	<l>	Logs
<1>	default	<ctrl-d>	Delete	<p>	Logs Previous
		<d>	Describe	<shift-f>	Port-Forward
		<e>	Edit	<s>	Shell
		<?>	Help	<n>	Show Node
		<ctrl-k>	Kill	<f>	Show PortForward

YAML

Pods(all)[7]

NAMESPACE↑	NAME	PF	READY	RESTARTS	STATUS	IP	NODE	AGE
kube-system	coredns-787d4945fb-tg2kz	●	1/1	0	Running	10.244.0.2	minikube	77s
kube-system	etcd-minikube	●	1/1	0	Running	192.168.49.2	minikube	90s
kube-system	kube-apiserver-minikube	●	1/1	0	Running	192.168.49.2	minikube	92s
kube-system	kube-controller-manager-minikube	●	1/1	0	Running	192.168.49.2	minikube	90s
kube-system	kube-proxy-pjwck	●	1/1	0	Running	192.168.49.2	minikube	77s
kube-system	kube-scheduler-minikube	●	1/1	0	Running	192.168.49.2	minikube	90s
kube-system	storage-provisioner	●	1/1	0	Running	192.168.49.2	minikube	90s

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

[k9scli.io](https://k9scli.io)

[github.com/derailed/k9s](https://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` = `wrap`
- `shift + f` = `port-forward`
- `y` = see the object `yaml`
- `ctrl + 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` = `scale` 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



THE LINUX FOUNDATION

# OPEN SOURCE SUMMIT NORTH AMERICA