Cheatography

Kubectl Cheat Sheet

by Sam15 via cheatography.com/127983/cs/24960/

Viewing and Finding Resources

List all namespaces in the cluster	kubectl get namespaces
List all services in the namespace	kubectl get services
List all pods in all namespaces	kubectl get pods --all-namespaces
List all pods in the namespace, with more details	kubectl get pods -o wide
List a particular deployment	kubectl get deployment my-dep
List all pods in the namespace	kubectl get pods
Get a pod's YAML	kubectl get pod my-pod -o yaml
List Services Sorted by Name	kubectl get services --sort-by=.metadata.name
Get all running pods in the namespace	kubectl get pods --field-selector=status.phase=-Running

Creating Objects

Create resource(s)	kubectl apply -f ./my-manifest.yaml
Create resource(s) in all manifest files in dir	kubectl apply -f ./dir
Create resource(s) from URL	kubectl apply -f https://git.io/vPieo
Start a single instance of nginx	kubectl create deployment nginx --image=nginx
Get the documentation for pod manifests	kubectl explain pods

Modifying and Deleting Resources

Add a label	kubectl label pods my-pod new-label=awesome
Add an annotation	kubectl annotate pods my-pod icon-url=http://goo.gl/XXBTWq

Interacting with Running Pods

Dump pod logs (stdout)	kubectl logs my-pod
Dump pod logs, with label name=myLabel (stdout)	kubectl logs -l name=myLabel
Dump pod logs (stdout) for a previous instantiation of a container	kubectl logs my-pod --previous
Dump pod container logs (stdout, multi--container case)	kubectl logs my-pod -c my-container
Stream pod logs (stdout)	kubectl logs -f my-pod
Stream pod container logs (stdout, multi--container case)	kubectl logs -f my-pod -c my-container
Run command in existing pod (1 container case)	kubectl exec my-pod -- ls /
Show metrics for a given pod and its containers	kubectl top pod POD_NAME --containers

Formatting Output

Print a table using a comma separated list of custom columns	-o=custom-columns=<spec>
Print a table using the custom columns template in the <filename> file	-o=custom-columns--file=<filename>
Print only the resource name and nothing else	-o=name
Print the fields defined in a jsonpath expression	-o=jsonpath=<template>
Print the fields defined by the jsonpath expression in the <filename> file	-o=jsonpath--file=<filename>
Output a YAML formatted API object	-o=yaml
Output in the plain-text format with any additional information, and for pods, the node name is included	-o=wide

By **Sam15**
cheatography.com/sam15/

Published 26th October, 2020.
Last updated 26th October, 2020.