Kubectl Cheat Sheet

BY JESSICA CHERRY

kubectl is a powerful command-line tool to maintain your Kubernetes cluster. Here are commonly used commands to take you above and beyond average cluster administration.

Basic Commands

kubectl get

```
kubectl get <resource> --output wide
```

List all information about the select resource type. Common resources include:

- Pods (`kubectl get pods`)
- Namespaces (`kubectl get ns`)
- Nodes (`kubectl get node`)
- Deployments (`kubectl get deploy`)
- Service (`kubectl get svc`)
- ReplicaSets (`kubectl get rs`)

Call resources using singular (pod), plural (pods), or with shortcuts.

Get pod by namespace:

`-n, --namespace`

Wait for a resource to finish:

`-w, --watch`

Query multiple resources (comma-separated values):

```
kubectl get rs,services -o wide
```

kubectl edit

```
kubectl edit <resource-type>/<name>
```

Edit resources in a cluster. The default editor opens unless KUBE_EDITOR is specificed:

```
KUBE_EDITOR="nano" kubectl edit \
svc/container-registry
```

kubectl create

```
kubectl create --filename ./pod.yaml
```

Some resources require a name parameter. A small list of resources you can create include:

- Services (`svc`)
- Cronjobs (`cj`)
- Deployments (`deploy`)
- Quotas (`quota`)

See required parameters:

```
kubectl create cronjobs --help
```

kubectl delete

```
kubectl delete <resource>
```

Remove one more our resources by name, label, or by filename.

If you want to delete pods by label in mass you have to describe the pod and gather the app=”name” from the label section. This makes it easier to cycle multiple containers at once.

Add --grace-period=5 to give yourself a few seconds to cancel before deleting:

```
kubectl delete pod foo --grace-period=5
```


opensource.com: Kubectl Cheat Sheet

BY JESSICA CHERRY

Troubleshooting Commands

kubectl describe

```
kubectl describe <resource-type> <name>
```

Show details of a resource. Often used to describe a pod or node for errors in events, or whether resources are too limited to use. A few common examples:

```
kubectl describe pods/nginx

kubectl describe nodes container.proj

kubectl describe pods -l name=myLabel
```

kubectl logs

```
kubectl logs [-f] [-c] <resource-name>
[<pod-name>]
```

Helpful when an application is dead within a pod but the pod and containers are shown as active.

Follow a log as it is created:

```
-f, --follow
```

Get logs from a specific container:

```
-c, --container

kubectl logs -f -c ruby-app web-1
```

Advanced Commands

kubectl apply

```
kubectl apply --file ./<filename>
```

Apply configurations from files for resources within your cluster.

Use apply to add, update, or delete fields:

```
kubectl apply -f ./pod.json

cat pod.json | kubectl apply -f -
```

kubectl exec

`kubectl exec` is the ability to execute into a container that is having an issue but logging and debugging an app hasn't provided any answers.

kubectl cp

```
kubectl cp <source> <destination>
```

Copy files and directories to and from containers. The tar binary must be in the container. This can also be used to pull or restore backups in an emergency. File location must be specified.

Copy a file from a local machine to a container:

```
kubectl cp /tmp/cmd.txt \
charts/chart-884c-dmcfv:/tmp/cmd.txt
```

Copy file from container to local machine:

```
kubectl cp \
charts/chart-884c-dmcfv:/tmp/cmd.txt \
/tmp/cmd.txt
```