**Kubernetes Cheat Sheet**

A cheat sheet for Kubernetes commands.

**Kubectl Alias**

Linux

alias k=kubectl

Windows

Set-Alias -Name k -Value kubectl

**Cluster Info**

* Get clusters

kubectl config get-clusters

NAME

docker-for-desktop-cluster

foo

* Get cluster info.

kubectl cluster-info

Kubernetes master is running at https://172.17.0.58:8443

**Contexts**

A context is a cluster, namespace and user.

* Get a list of contexts.

kubectl config get-contexts

CURRENT NAME CLUSTER AUTHINFO NAMESPACE

docker-desktop docker-desktop docker-desktop

\* foo foo foo bar

* Get the current context.

kubectl config current-context

foo

* Switch current context.

kubectl config use-context docker-desktop

* Set default namesapce

kubectl config set-context $(kubectl config current-context) --namespace=my-namespace

To switch between contexts, you can also install and use [kubectx](https://github.com/ahmetb/kubectx).

**Get Commands**

kubectl get all

kubectl get namespaces

kubectl get configmaps

kubectl get nodes

kubectl get pods

kubectl get rs

kubectl get svc kuard

kubectl get endpoints kuard

Additional switches that can be added to the above commands:

* -o wide - Show more information.
* --watch or -w - watch for changes.

**Namespaces**

* --namespace - Get a resource for a specific namespace.

You can set the default namespace for the current context like so:

kubectl config set-context $(kubectl config current-context) --namespace=my-namespace

To switch namespaces, you can also install and use [kubens](https://github.com/ahmetb/kubectx/blob/master/kubens).

**Labels**

* Get pods showing labels.

kubectl get pods --show-labels

* Get pods by label.

kubectl get pods -l environment=production,tier!=frontend

kubectl get pods -l 'environment in (production,test),tier notin (frontend,backend)'

**Describe Command**

kubectl describe nodes [id]

kubectl describe pods [id]

kubectl describe rs [id]

kubectl describe svc kuard [id]

kubectl describe endpoints kuard [id]

**Delete Command**

kubectl delete nodes [id]

kubectl delete pods [id]

kubectl delete rs [id]

kubectl delete svc kuard [id]

kubectl delete endpoints kuard [id]

Force a deletion of a pod without waiting for it to gracefully shut down

kubectl delete pod-name --grace-period=0 --force

**Create vs Apply**

kubectl create can be used to create new resources while kubectl apply inserts or updates resources while maintaining any manual changes made like scaling pods.

* --record - Add the current command as an annotation to the resource.
* --recursive - Recursively look for yaml in the specified directory.

**Create Pod**

kubectl run kuard --generator=run-pod/v1 --image=gcr.io/kuar-demo/kuard-amd64:1 --output yaml --export --dry-run > kuard-pod.yml

kubectl apply -f kuard-pod.yml

**Create Deployment**

kubectl run kuard --image=gcr.io/kuar-demo/kuard-amd64:1 --output yaml --export --dry-run > kuard-deployment.yml

kubectl apply -f kuard-deployment.yml

**Create Service**

kubectl expose deployment kuard --port 8080 --target-port=8080 --output yaml --export --dry-run > kuard-service.yml

kubectl apply -f kuard-service.yml

**Export YAML for New Pod**

kubectl run my-cool-app —-image=me/my-cool-app:v1 --output yaml --export --dry-run > my-cool-app.yaml

**Export YAML for Existing Object**

kubectl get deployment my-cool-app --output yaml --export > my-cool-app.yaml

**Logs**

* Get logs.

kubectl logs -l app=kuard

* Get logs for previously terminated container.

kubectl logs POD\_NAME --previous

* Watch logs in real time.

kubectl attach POD\_NAME

* Copy files out of pod (Requires tar binary in container).

kubectl cp POD\_NAME:/var/log .

You can also install and use [kail](https://github.com/boz/kail).

**Port Forward**

kubectl port-forward deployment/kuard 8080:8080

**Scaling**

* Update replicas.

kubectl scale deployment nginx-deployment --replicas=10

**Autoscaling**

* Set autoscaling config.

kubectl autoscale deployment nginx-deployment --min=10 --max=15 --cpu-percent=80

**Rollout**

* Get rollout status.

kubectl rollout status deployment/nginx-deployment

Waiting for rollout to finish: 2 out of 3 new replicas have been updated...

deployment "nginx-deployment" successfully rolled out

* Get rollout history.

kubectl rollout history deployment/nginx-deployment

kubectl rollout history deployment/nginx-deployment --revision=2

* Undo a rollout.

kubectl rollout undo deployment/nginx-deployment

kubectl rollout undo deployment/nginx-deployment --to-revision=2

* Pause/resume a rollout

kubectl rollout pause deployment/nginx-deployment

kubectl rollout resume deploy/nginx-deployment

**Pod Example**

apiVersion: v1

kind: Pod

metadata:

name: cuda-test

spec:

containers:

- name: cuda-test

image: "k8s.gcr.io/cuda-vector-add:v0.1"

resources:

limits:

nvidia.com/gpu: 1

nodeSelector:

accelerator: nvidia-tesla-p100

**Deployment Example**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

namespace: my-namespace

labels:

- environment: production,

- teir: frontend

annotations:

- key1: value1,

- key2: value2

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.7.9

ports:

- containerPort: 80

**Dashboard**

* Enable proxy

kubectl proxy

**Azure Kubernetes Service**

[List of az aks commands](https://docs.microsoft.com/en-us/cli/azure/aks?view=azure-cli-latest)

**Get Credentials**

az aks get-credentials --resource-group <Resource Group Name> --name <AKS Name>

**Show Dashboard**

Secure the dashboard like [this](http://blog.cowger.us/2018/07/03/a-read-only-kubernetes-dashboard.html). Then run:

az aks browse --resource-group <Resource Group Name> --name <AKS Name>

**Upgrade**

Get updates

az aks get-upgrades --resource-group <Resource Group Name> --name <AKS Name>