1##Create a namespace called 'mynamespace' and a pod with image nginx called nginx on this namespace

kubectl create namespace mynamespace

kubectl run nginx --image=nginx --restart=Never -n mynamespace

2##Create the pod that was just described using YAML

Easily generate YAML with:

kubectl run nginx --image=nginx --restart=Never --dry-run=client -o yaml | kubectl create -n mynamespace -f -

3##Create a busybox pod (using kubectl command) that runs the command "env". Run it and see the output

kubectl run busybox --image=busybox --command --restart=Never -it -- env # -it will help in seeing the output

or, just run it without -it

kubectl run busybox --image=busybox --command --restart=Never -- env

and then, check its logs

kubectl logs busybox

4##Create a busybox pod (using YAML) that runs the command "env". Run it and see the output

create a YAML template with this command

kubectl run busybox --image=busybox --restart=Never --dry-run -o yaml --command -- env > envpod.yaml

see it

cat envpod.yaml

5##Get the YAML for a new namespace called 'myns' without creating it

kubectl create namespace myns -o yaml --dry-run

6##Get the YAML for a new ResourceQuota called 'myrq' with hard limits of 1 CPU, 1G memory and 2 pods without creating it

kubectl create quota myrq --hard=cpu=1,memory=1G,pods=2 --dry-run -o yaml

7##Get pods on all namespaces

kubectl get po --all-namespaces

Alternatively

kubectl get po -A

8##Create a pod with image nginx called nginx and allow traffic on port 80

kubectl run nginx --image=nginx --restart=Never --port=80

9##Change pod's image to nginx:1.7.1. Observe that the pod will be killed and recreated as soon as the image gets pulled

kubectl set image POD/POD_NAME CONTAINER_NAME=IMAGE_NAME:TAG

kubectl set image pod/nginx nginx=nginx:1.7.1

kubectl describe po nginx # you will see an event 'Container will be killed and recreated'

kubectl get po nginx -w # watch it

10##Get nginx pod's ip created in previous step, use a temp busybox image to wget its '/'

kubectl get po -o wide # get the IP, will be something like '10.1.1.131'

create a temp busybox pod

kubectl run busybox --image=busybox --rm -it --restart=Never -- wget -O- 10.1.1.131:80

11##Get pod's YAML

kubectl get po nginx -o yaml

or

kubectl get po nginx -oyaml

or

kubectl get po nginx --output yaml

or

kubectl get po nginx --output=yaml

12##Get information about the pod, including details about potential issues (e.g. pod hasn't started)

kubectl describe po nginx

13##Get pod logs

kubectl logs nginx

14##If pod crashed and restarted, get logs about the previous instance

kubectl logs nginx -p

15##Execute a simple shell on the nginx pod

kubectl exec -it nginx -- /bin/sh

16##Create a busybox pod that echoes 'hello world' and then exits

kubectl run busybox --image=busybox -it --restart=Never -- echo 'hello world'

or

kubectl run busybox --image=busybox -it --restart=Never -- /bin/sh -c 'echo hello world'

17##Do the same, but have the pod deleted automatically when it's completed

kubectl run busybox --image=busybox -it --rm --restart=Never -- /bin/sh -c 'echo hello world'

kubectl get po # nowhere to be found :)

18##Create an nginx pod and set an env value as 'var1=val1'. Check the env value existence within the pod

kubectl run nginx --image=nginx --restart=Never --env=var1=val1

then

kubectl exec -it nginx -- env

or

kubectl exec -it nginx -- sh -c 'echo $var1'

or

kubectl describe po nginx | grep val1

or

kubectl run nginx --restart=Never --image=nginx --env=var1=val1 -it --rm -- env