Kubectl Commands

Pods

To List running pods in node:

command: `kubectl get pods`

```
fazil Sep 15 22:58 ~ > kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-bbcb89c6c-x8hdm    1/1     Running   0          53s
```

To Delete running pod in node:

command: `kubectl delete pods [POD_NAME]`

```
fazil Sep 15 23:11 ~ > kubectl delete pods hello-minikube-bbcb89c6c-x8hdm
pod "hello-minikube-bbcb89c6c-x8hdm" deleted from default namespace
```

Deployments

To Create a deployment:

command: `kubectl create deployment [DEPLOYEMNT_NAME] --image=[CONTAINER_IMAGE_NAME]`

```
fazil Sep 15 23:21 ~ > kubectl create deployment nginx-depl --image=nginx
deployment.apps/nginx-depl created
```

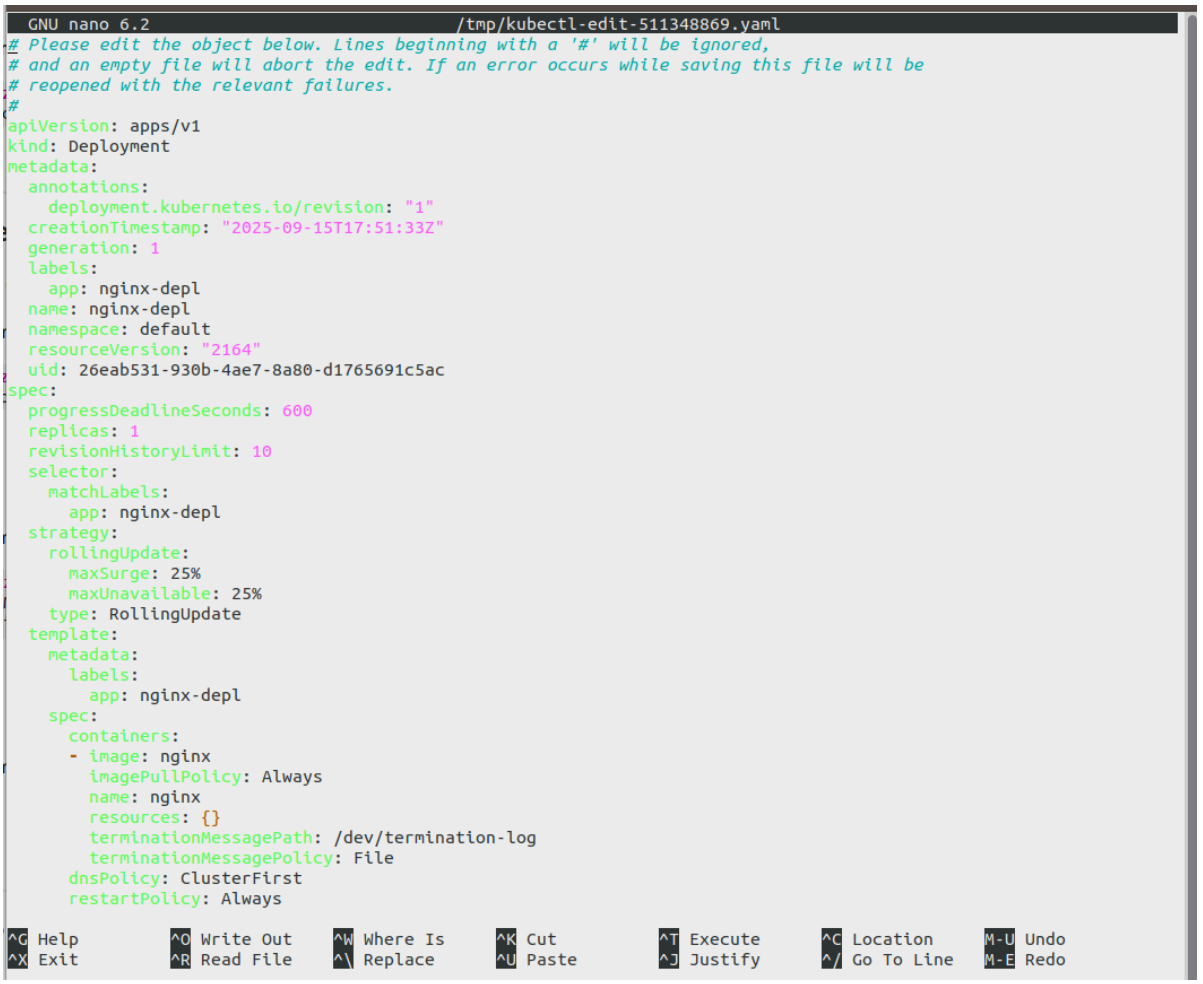
To List deployments:

command: `kubectl get deployments`

```
fazil Sep 15 23:21 ~ > kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
nginx-depl   1/1     1            1           17s
```

To Edit a deployment:

command: `kubectl edit deployment [DEPLOYMENT_NAME]`



Note: When I edited the `spec->containers->image` from **nginx** to **nginx:1.16**

```
fazil Sep 15 23:44 ~ > kubectl get pods
NAME                          READY   STATUS              RESTARTS   AGE
nginx-depl-5fcbf6fffd-jwgtz   1/1     Running             0          22m
nginx-depl-7d9b57bb64-znkn2   0/1     ContainerCreating   0          11s
```

- The new container `nginx-depl-7d9b57bb64-znkn2` is being created
- Once the new container is active, old container will be deleted
- Above example is with respect to **Pods**, but the same will happen with **ReplicaSets**

To expose a Deployment to outside network:

command: `kubectl expose deployment [DEPLOYMENT_NAME] --type=NodePort --port=80`

```
fazil Sep 15 23:53 ~ > kubectl expose deployment nginx-depl --type=NodePort --port=80
service/nginx-depl exposed
```

after exposing the deployment, run `kubectl get services` to get external IP address

```
fazil Sep 15 23:55 ~ > kubectl get services
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>          443/TCP        25h
nginx-depl   NodePort    10.109.84.242   192.168.1.100   80:30965/TCP   4m58s
```

Services

To Check running services in node:

command: `kubectl get services`

```
fazil Sep 15 22:58 ~ > kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   24h
```

ReplicaSet

To Check running replcaset in node:

command: `kubectl get replicaset`

```
fazil Sep 15 23:21 ~ > kubectl get replicaset
NAME                    DESIRED   CURRENT   READY   AGE
nginx-depl-5fcbf6fffd   1         1         1       9m11s
```

Useful Commands

To check logs of a Pod

command: `kubectl logs [POD_NAME]`

```
fazil Sep 15 23:55 ~ > kubectl logs nginx-depl-7d9b57bb64-znkn2
10.244.0.1 - - [15/Sep/2025:18:25:32 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/7.81.0" "-"
```

To get detailed info of a Pod

command: `kubectl describe pod [POD_NAME]`

```
fazil Sep 16 0:11 ~ > kubectl describe pod nginx-depl-7d9b57bb64-znkn2
Name:             nginx-depl-7d9b57bb64-znkn2
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Mon, 15 Sep 2025 23:44:05 +0530
Labels:           app=nginx-depl
                  pod-template-hash=7d9b57bb64
Annotations:      <none>
Status:           Running
IP:               10.244.0.10
IPs:
  IP:           10.244.0.10
Controlled By:  ReplicaSet/nginx-depl-7d9b57bb64
Containers:
  nginx:
    Container ID:   docker://fe6bd92e9cb2842b67231080bad7656986e830f3cd5a5271b0a21d2177d0ff8d
    Image:          nginx:1.16
    Image ID:       docker-pullable://nginx@sha256:d20aa6d1cae56fd17cd458f4807e0de462caf2336f0b70b5eeb69fcaaf30dd9c
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Mon, 15 Sep 2025 23:45:16 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-hmq2f (ro)
```

To enter into interactivity mode(log in to the pod) of the Pod

command: `kubectl exec -it [POD_NAME] -- bin/bash`

above command will start the bash command of respective Pod

```
fazil Sep 16 0:14 ~ > kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
nginx-depl-7d9b57bb64-znkn2   1/1     Running   0          30m
fazil Sep 16 0:14 ~ > kubectl exec -it nginx-depl-7d9b57bb64-znkn2 -- bin/bash
root@nginx-depl-7d9b57bb64-znkn2:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@nginx-depl-7d9b57bb64-znkn2:/# exit
exit
```

To generate a basic Deployment Config file

command: `kubectl create deployment [DEPLOYMENT_NAME] --image=[CONTAINER_IMAGE_NAME] --dry-run=client -o yaml`

- The `--dry-run=client` option simulates the action, without actually running it
- The `-o yaml` option prints the file in YAML format

```
fazil Sep 16 0:22 ~ > kubectl create deployment nginx --image=nginx --dry-run=client -o yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
```

To Create a Deployment using a Config file

command: `kubectl apply -f nginx-deployment.yaml`

```
fazil Sep 16 0:29 ~/Desktop/Kubernetes > kubectl create deployment nginx --image=nginx --dry-run=client -o yaml > nginx-deployment.yaml
fazil Sep 16 0:29 ~/Desktop/Kubernetes > cat nginx-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
fazil Sep 16 0:29 ~/Desktop/Kubernetes > kubectl get deployments
No resources found in default namespace.
fazil Sep 16 0:29 ~/Desktop/Kubernetes > kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx created
fazil Sep 16 0:29 ~/Desktop/Kubernetes > kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   0/1     1            0           2s
```

- After creating the deployment using config file, when i update the **replicas to 2** in config file and apply the deployment again, observe in the below image.
    - instead of `deployment.apps/nginx created` as shown in image above
    - it is showing `deployment.apps/nginx configured`
    - this is because, kubernetes is aware when to create new deployment, and when to update

```
fazil Sep 16 0:33 ~/Desktop/Kubernetes > kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx configured
fazil Sep 16 0:33 ~/Desktop/Kubernetes > kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   1/2     2            1           3m57s
fazil Sep 16 0:33 ~/Desktop/Kubernetes > kubectl get pods
NAME                     READY   STATUS              RESTARTS   AGE
nginx-66686b6766-hdnng   0/1     ContainerCreating   0          11s
nginx-66686b6766-sgssw   1/1     Running             0          3m59s
```

NOTE:

```
fazil Sep 15 23:33 ~ > kubectl get replicaset
NAME                      DESIRED   CURRENT   READY   AGE
nginx-depl-5fcbf6fffd   1         1         1       11m
fazil Sep 15 23:33 ~ > kubectl get pod
NAME                            READY   STATUS    RESTARTS   AGE
nginx-depl-5fcbf6fffd-jwgtz   1/1     Running   0          11m
```

ReplicaSet is managing the replicas of **Pod**

- **nginx-depl-5fcbf6fffd** : ReplicaSet
- **nginx-depl-5fcbf6fffd**-jwgtz : Pod

- Deployment manages a ReplicaSet
- ReplicaSet manages Pod
- Pod is a abstraction of a Container

- Everything below a Deployment will be managed by Kubernetes