KUBERNATES

==========Kind installation steps==========
curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.18.0/kind-linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind

==========kubectl installation steps==========
curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl"

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

==========configfile==========
```
# three node (two workers) cluster config
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
- role: worker
- role: worker
```
==========create kind cluster==========

kind create cluster --config configfile

If nodes are in NotReady state do below steps
sudo sysctl net/netfilter/nf_conntrack_max=131072
kind delete cluster
kind create cluster --config configfile

****

kubectl get nodes (command is used to retrieve information about the nodes in a Kubernetes cluster)

```
File  Actions  Edit  View  Help
root@22fa3c5b802c:~# kubectl get nodes
NAME                 STATUS   ROLES           AGE   VERSION
kind-control-plane   Ready    control-plane   16m   v1.26.3
kind-worker          Ready    <none>          16m   v1.26.3
kind-worker2         Ready    <none>          16m   v1.26.3
root@22fa3c5b802c:~#
```

kubectl api-resources (command is used to list the available API resources in a Kubernetes cluster. It shows all the resources that can be accessed using the Kubernetes API server.)

```
root@22fa3c5b802c:~# kubectl api-resources
NAME                              SHORTNAMES   APIVERSION                             NAMESPACED   KIND
bindings                                       v1                                     true         Binding
componentstatuses                 cs           v1                                     false        ComponentStatus
configmaps                        cm           v1                                     true         ConfigMap
endpoints                         ep           v1                                     true         Endpoints
events                            ev           v1                                     true         Event
limitranges                       limits       v1                                     true         LimitRange
namespaces                        ns           v1                                     false        Namespace
nodes                             no           v1                                     false        Node
persistentvolumeclaims            pvc          v1                                     true         PersistentVolumeClaim
persistentvolumes                 pv           v1                                     false        PersistentVolume
pods                              po           v1                                     true         Pod
podtemplates                                   v1                                     true         PodTemplate
replicationcontrollers            rc           v1                                     true         ReplicationController
resourcequotas                    quota        v1                                     true         ResourceQuota
secrets                                        v1                                     true         Secret
serviceaccounts                   sa           v1                                     true         ServiceAccount
services                          svc          v1                                     true         Service
mutatingwebhookconfigurations                  admissionregistration.k8s.io/v1        false        MutatingWebhookConfiguration
validatingwebhookconfigurations                admissionregistration.k8s.io/v1        false        ValidatingWebhookConfiguration
customresourcedefinitions         crd,crds     apiextensions.k8s.io/v1                false        CustomResourceDefinition
apiservices                                    apiregistration.k8s.io/v1              false        APIService
controllerrevisions                            apps/v1                                true         ControllerRevision
daemonsets                        ds           apps/v1                                true         DaemonSet
deployments                       deploy       apps/v1                                true         Deployment
replicasets                       rs           apps/v1                                true         ReplicaSet
statefulsets                      sts          apps/v1                                true         StatefulSet
```

root@22fa3c5b802c:~# kubectl run first-pod --image=kaushalnitb/myfuturehub:1.0 port=8080 (command to create pod in kubernates)

```
root@22fa3c5b802c:~# kubectl  run first-pod --image=kaushalnitb/myfuturehub:1.0 port=8080
pod/first-pod created
root@22fa3c5b802c:~#
```

root@22fa3c5b802c:~# kubectl get pods

The command list all created pods.

root@22fa3c5b802c:~# kubectl get po -o wide

used to list all the running pods in the current Kubernetes namespace, along with additional details such as node name, IP address, and node status

root@22fa3c5b802c:~# kubectl describe po first-pod
used to get detailed information about a specific pod in the current namespace

- Pod name and namespace
- Labels and annotations associated with the pod
- Pod IP address and node name
- Pod status, including whether it is running, pending, or has failed
- Container information, including the image being used and the status of each container
- Events associated with the pod, including any errors or warnings

```
root@22fa3c5b802c:~# kubectl describe po first-pod
Name:             first-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             kind-worker/172.20.0.2
Start Time:       Thu, 20 Apr 2023 03:34:24 +0000
Labels:           run=first-pod
Annotations:      <none>
Status:           Running
IP:               10.244.2.4
IPs:
  IP:  10.244.2.4
Containers:
  first-pod:
    Container ID:  containerd://5b4c34582e5dd4a23a6f9ff795e17afc9a2ecd429228b9f079a728db260e56d8
    Image:         kaushalnitb/myfuturehub:1.0
    Image ID:      docker.io/kaushalnitb/myfuturehub@sha256:d67c9c42a21c48af3a77f61de2432ab3d39b7f3ebba68f14e76615c3b95a4825
    Port:          <none>
    Host Port:     <none>
    Args:
      port=8080
    State:          Waiting
      Reason:       CrashLoopBackOff
    Last State:     Terminated
      Reason:       StartError
      Message:      failed to create containerd task: failed to create shim task: OCI runtime create failed: runc create failed: unab
le to start container process: exec: "port=8080": executable file not found in $PATH: unknown
      Exit Code:    128
      Started:      Thu, 01 Jan 1970 00:00:00 +0000
      Finished:     Thu, 20 Apr 2023 03:40:07 +0000
    Ready:          False
```

****
Kubernates deployment
#kubectl create deploy nginx --image=nginx:latest --port=80 --dry-run=client -o yaml > deploy.yaml
(used to create a deployment of the Nginx container image)

```
cluster1-controlplane ~ ➜  kubectl create deploy nginx --image=nginx:latest --port=80 --dry-run=client -o yaml > deploy.yaml

cluster1-controlplane ~ ➜  ls
deploy.yaml
```

Deploy.yaml :

Terminal 1 +

```
cluster1-controlplane ~ ➜  cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:latest
        name: nginx
        ports:
        - containerPort: 80
        resources: {}
status: {}
```

#kubectl apply -f deploy.yaml (used to apply a configuration file to a Kubernetes cluster)

```
cluster1-controlplane ~ ➜  kubectl apply -f deploy.yaml
deployment.apps/nginx created

cluster1-controlplane ~ ➜  kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/nginx-cd55c47f5-7d58r    1/1     Running   0          21s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   12m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   1/1     1            1           21s

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-cd55c47f5    1         1         1       21s

cluster1-controlplane ~ ➜
```

```
cluster1-controlplane ~ ➜  kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-cd55c47f5-7d58r    1/1     Running   0          5m53s

cluster1-controlplane ~ ➜
```

If you wanted to more pods we should update replicas: 8 Then we can do in three ways

(1) kubectl scale deploy nginx --replicas=8 (This scaling type is called Imperative command)

```
cluster1-controlplane ~ ➜  kubectl scale deploy nginx --replicas=8
deployment.apps/nginx scaled
```

I deleted one pod then see the below screenshot,the deployment object automatically creating the pods.

#kubectl delete po nginx-cd55c47f5-7d58r

```
cluster1-controlplane ~ ➜  kubectl get po
NAME                     READY   STATUS    RESTARTS   AGE
nginx-cd55c47f5-7d58r    1/1     Running   0          16m
nginx-cd55c47f5-86rvr    1/1     Running   0          87s
nginx-cd55c47f5-fgqwt    1/1     Running   0          87s
nginx-cd55c47f5-fhf4r    1/1     Running   0          87s
nginx-cd55c47f5-hbxxm    1/1     Running   0          87s
nginx-cd55c47f5-l22nf    1/1     Running   0          87s
nginx-cd55c47f5-q85bj    1/1     Running   0          87s
nginx-cd55c47f5-vl67p    1/1     Running   0          87s

cluster1-controlplane ~ ➜  kubectl delete po nginx-cd55c47f5-7d58r
pod "nginx-cd55c47f5-7d58r" deleted

cluster1-controlplane ~ ➜  kubectl get po
NAME                     READY   STATUS    RESTARTS   AGE
nginx-cd55c47f5-86rvr    1/1     Running   0          110s
nginx-cd55c47f5-fgqwt    1/1     Running   0          110s
nginx-cd55c47f5-fhf4r    1/1     Running   0          110s
nginx-cd55c47f5-hbxxm    1/1     Running   0          110s
nginx-cd55c47f5-l22nf    1/1     Running   0          110s
nginx-cd55c47f5-lmj7f    1/1     Running   0          4s
nginx-cd55c47f5-q85bj    1/1     Running   0          110s
nginx-cd55c47f5-vl67p    1/1     Running   0          110s

cluster1-controlplane ~ ➜
```

```
cluster1-controlplane ~ ➜  kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/nginx-cd55c47f5-86rvr    1/1     Running   0          3m34s
pod/nginx-cd55c47f5-fgqwt    1/1     Running   0          3m34s
pod/nginx-cd55c47f5-fhf4r    1/1     Running   0          3m34s
pod/nginx-cd55c47f5-hbxxm    1/1     Running   0          3m34s
pod/nginx-cd55c47f5-l22nf    1/1     Running   0          3m34s
pod/nginx-cd55c47f5-lmj7f    1/1     Running   0          108s
pod/nginx-cd55c47f5-q85bj    1/1     Running   0          3m34s
pod/nginx-cd55c47f5-vl67p    1/1     Running   0          3m34s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   30m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   8/8     8            8           18m

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-cd55c47f5    8         8         8       18m
```

In the same way we can scale down the pods as well.please see the below screenshot.

```
cluster1-controlplane ~ ➜  kubectl scale deploy nginx --replicas=2
deployment.apps/nginx scaled

cluster1-controlplane ~ ➜  kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/nginx-cd55c47f5-86rvr    1/1     Running   0          4m33s
pod/nginx-cd55c47f5-hbxxm    1/1     Running   0          4m33s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   31m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   2/2     2            2           19m

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-cd55c47f5    2         2         2       19m
```

(2) Second way to scale up /down

We need to update replicas (replicas: 7) in deploy.yaml then apply the yaml,using the below command.

#kubectl apply -f deploy.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 7
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx:latest
        name: nginx
        ports:
        - containerPort: 80
        resources: {}
status: {}
```

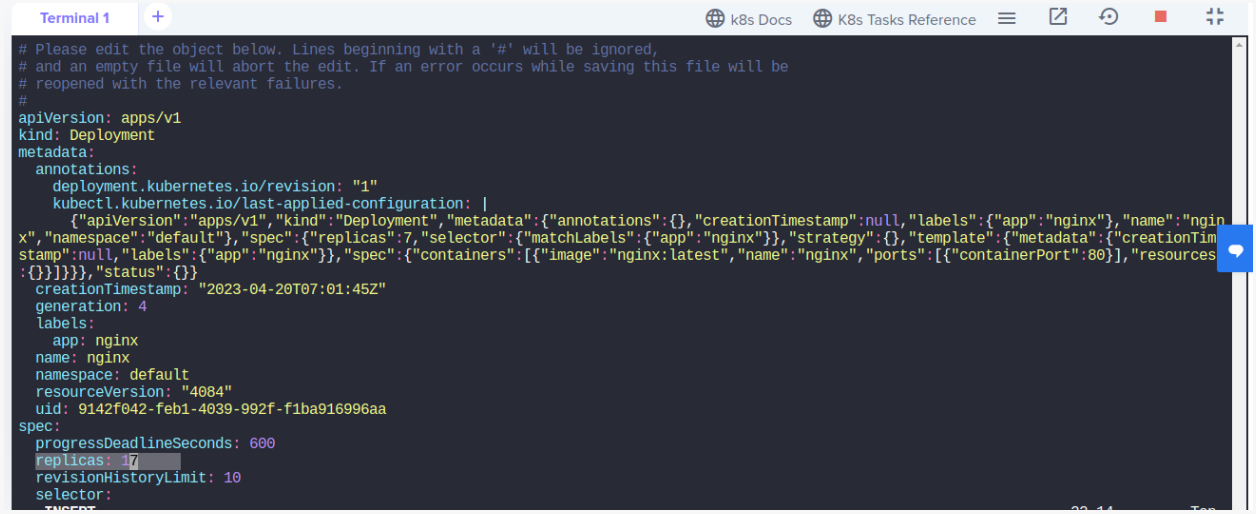
```
:wq
```

```
cluster1-controlplane ~ ➜  kubectl  apply -f deploy.yaml
deployment.apps/nginx configured

cluster1-controlplane ~ ➜  kubectl get all
NAME                         READY   STATUS    RESTARTS   AGE
pod/nginx-cd55c47f5-86rvr    1/1     Running   0          11m
pod/nginx-cd55c47f5-8nr4f    1/1     Running   0          8s
pod/nginx-cd55c47f5-cbr5b    1/1     Running   0          9s
pod/nginx-cd55c47f5-hbxxm    1/1     Running   0          11m
pod/nginx-cd55c47f5-qhncb    1/1     Running   0          8s
pod/nginx-cd55c47f5-sww9w    1/1     Running   0          9s
pod/nginx-cd55c47f5-xwppk    1/1     Running   0          9s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   39m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   7/7     7            7           26m

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-cd55c47f5    7         7         7       26m
```

(3) Third method is modifying live objects.

We need to modify live nginx object using below command

#kubectl edit deploy nginx (Then api server will automatically create scale up/down the pods).please find the detail in below screenshot.



Terminal 1 +

```
NAME                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-cd55c47f5       7         7         7       26m

cluster1-controlplane ~ ➜  kubectl edit deploy nginx
deployment.apps/nginx edited

cluster1-controlplane ~ ➜  kubectl get po
NAME                     READY   STATUS    RESTARTS   AGE
nginx-cd55c47f5-4s44r    1/1     Running   0          12s
nginx-cd55c47f5-65x2k    1/1     Running   0          12s
nginx-cd55c47f5-86rvr    1/1     Running   0          16m
nginx-cd55c47f5-8m4kj    1/1     Running   0          12s
nginx-cd55c47f5-8nr4f    1/1     Running   0          4m32s
nginx-cd55c47f5-9lx5j    1/1     Running   0          12s
nginx-cd55c47f5-cbr5b    1/1     Running   0          4m33s
nginx-cd55c47f5-dsfnv    1/1     Running   0          12s
nginx-cd55c47f5-dsv6f    1/1     Running   0          12s
nginx-cd55c47f5-gmprp    1/1     Running   0          12s
nginx-cd55c47f5-hbxxm    1/1     Running   0          16m
nginx-cd55c47f5-hwlnp    1/1     Running   0          12s
nginx-cd55c47f5-jzqnq    1/1     Running   0          12s
nginx-cd55c47f5-qhncb    1/1     Running   0          4m32s
nginx-cd55c47f5-swqtr    1/1     Running   0          12s
nginx-cd55c47f5-sww9w    1/1     Running   0          4m33s
nginx-cd55c47f5-xwppk    1/1     Running   0          4m33s
```

Note :Prod environment we should use imperative command.

If you want to see how many containers running in a pod then describe the pod.

#kubectl describe po nginx-cd55c47f5-4s44r

```
cluster1-controlplane ~ ➜  kubectl describe po nginx-cd55c47f5-4s44r
Name:             nginx-cd55c47f5-4s44r
Namespace:        default
Priority:         0
Service Account:  default
Node:             cluster1-node02/192.11.65.20
Start Time:       Thu, 20 Apr 2023 03:32:30 -0400
Labels:           app=nginx
                  pod-template-hash=cd55c47f5
Annotations:      <none>
Status:           Running
IP:               10.244.64.9
IPs:
  IP:           10.244.64.9
Controlled By:  ReplicaSet/nginx-cd55c47f5
Containers:
  nginx:
    Container ID:   containerd://01a019a93c07ec92bfb407d7a1a0a8f2d325d25d04060fb3da3c4b5ba4e1e380
    Image:          nginx:latest
    Image ID:       docker.io/library/nginx@sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
```

****

Suppose your manager said we need to upgrade our app and we need to update app version without downtime.we will see in the below lab.

#kubectl rollout status deploy nginx

#kubectl rollout history deploy nginx

```
cluster1-controlplane ~ ➜  kubectl rollout status deploy nginx
deployment "nginx" successfully rolled out

cluster1-controlplane ~ ➜  kubectl rollout history deploy nginx
deployment.apps/nginx
REVISION  CHANGE-CAUSE
1         <none>
```

Below screenshot shows some pods are creating and some are terminating.

```
cluster1-controlplane ~ ➜  kubectl rollout status deploy nginx
Waiting for deployment "nginx" rollout to finish: 5 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 5 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 5 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 4 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 4 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 4 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 3 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 3 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 3 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 2 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 2 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 2 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "nginx" rollout to finish: 14 of 18 updated replicas are available...
Waiting for deployment "nginx" rollout to finish: 15 of 18 updated replicas are available...
Waiting for deployment "nginx" rollout to finish: 16 of 18 updated replicas are available...
Waiting for deployment "nginx" rollout to finish: 17 of 18 updated replicas are available...
deployment "nginx" successfully rolled out
```

Below screenshot shows all pods rolled out.

```
cluster1-controlplane ~ ➜  kubectl get po
NAME                     READY   STATUS    RESTARTS   AGE
nginx-8689885776-28kjc   1/1     Running   0          9m38s
nginx-8689885776-4qgp5   1/1     Running   0          9m28s
nginx-8689885776-6qhcn   1/1     Running   0          9m38s
nginx-8689885776-7lrd6   1/1     Running   0          9m38s
nginx-8689885776-8mw8n   1/1     Running   0          9m24s
nginx-8689885776-cfxkn   1/1     Running   0          9m22s
nginx-8689885776-crjl7   1/1     Running   0          9m38s
nginx-8689885776-h8r58   1/1     Running   0          9m38s
nginx-8689885776-ldp7p   1/1     Running   0          9m38s
nginx-8689885776-nmfsl   1/1     Running   0          9m23s
nginx-8689885776-nwr9k   1/1     Running   0          9m27s
nginx-8689885776-pfx2p   1/1     Running   0          9m38s
nginx-8689885776-ps2qb   1/1     Running   0          9m28s
nginx-8689885776-snpmb   1/1     Running   0          9m38s
nginx-8689885776-tjq85   1/1     Running   0          9m26s
nginx-8689885776-vj6tx   1/1     Running   0          9m37s
nginx-8689885776-xp4vn   1/1     Running   0          9m26s
nginx-8689885776-zq2zz   1/1     Running   0          9m25s
```

Below screenshot shows old replicas are 0 and new created replicas are 18.mean rollout is successful.

```
cluster1-controlplane ~ ➜  kubectl get all
NAME                            READY   STATUS    RESTARTS   AGE
pod/nginx-8689885776-28kjc      1/1     Running   0          11m
pod/nginx-8689885776-4qgp5      1/1     Running   0          11m
pod/nginx-8689885776-6qhcn      1/1     Running   0          11m
pod/nginx-8689885776-7lrd6      1/1     Running   0          11m
pod/nginx-8689885776-8mw8n      1/1     Running   0          11m
pod/nginx-8689885776-cfxkn      1/1     Running   0          11m
pod/nginx-8689885776-crjl7      1/1     Running   0          11m
pod/nginx-8689885776-h8r58      1/1     Running   0          11m
pod/nginx-8689885776-ldp7p      1/1     Running   0          11m
pod/nginx-8689885776-nmfsl      1/1     Running   0          11m
pod/nginx-8689885776-nwr9k      1/1     Running   0          11m
pod/nginx-8689885776-pfx2p      1/1     Running   0          11m
pod/nginx-8689885776-ps2qb      1/1     Running   0          11m
pod/nginx-8689885776-snpmb      1/1     Running   0          11m
pod/nginx-8689885776-tjq85      1/1     Running   0          11m
pod/nginx-8689885776-vj6tx      1/1     Running   0          11m
pod/nginx-8689885776-xp4vn      1/1     Running   0          11m
pod/nginx-8689885776-zq2zz      1/1     Running   0          11m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   36m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   18/18   18           18          18m

NAME                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-8689885776      18        18        18      11m
replicaset.apps/nginx-cd55c47f5       0         0         0       18m
```

#kubectl describe pod/nginx-8689885776-zq2zz
Below screenshot shows the latest rolledout version is (Image: nginx:1.7.8)

#kubectl rollout history deploy nginx

```
cluster1-controlplane ~ ➜  kubectl rollout history deploy nginx
deployment.apps/nginx
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

#kubectl rollout history deploy nginx

```
cluster1-controlplane ~ ✖ kubectl describe  pod/nginx-8689885776-zq2zz
Name:             nginx-8689885776-zq2zz
Namespace:        default
Priority:         0
Service Account:  default
Node:             cluster1-node02/192.25.201.21
Start Time:       Thu, 20 Apr 2023 10:52:12 -0400
Labels:           app=nginx
                  pod-template-hash=8689885776
Annotations:      <none>
Status:           Running
IP:               10.244.64.6
IPs:
  IP:           10.244.64.6
Controlled By:  ReplicaSet/nginx-8689885776
Containers:
  nginx:
    Container ID:   containerd://b11a601d26246766cca6800fe8bf40ebb3896530b823527d5488e59b62d9d4a5
    Image:          nginx:1.7.8
    Image ID:       sha256:9362f6342d6920db0a6a2ccc1b13dd2a99cc6b421055940b48c468cda3494126
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 20 Apr 2023 10:52:15 -0400
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sckml (ro)
```

#kubectl rollout history deploy nginx
#kubectl set image deploy nginx nginx=nginx:1.7.9 --record

```
cluster1-controlplane ~ ➜  kubectl rollout history deploy nginx
deployment.apps/nginx
REVISION  CHANGE-CAUSE
1         <none>
2         <none>


cluster1-controlplane ~ ➜  kubectl set image deploy nginx nginx=nginx:1.7.9 --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/nginx image updated

cluster1-controlplane ~ ➜  kubectl rollout history deploy nginx
deployment.apps/nginx
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
3         kubectl set image deploy nginx nginx=nginx:1.7.9 --record=true
```

#kubectl get all (below screenshot shows 3rd replicaset got created.old pods terminated.)

```
cluster1-controlplane ~ ➜  kubectl get all
NAME                          READY   STATUS    RESTARTS   AGE
pod/nginx-7c48cc69bf-28l8k    1/1     Running   0          3m5s
pod/nginx-7c48cc69bf-bcbtd    1/1     Running   0          3m9s
pod/nginx-7c48cc69bf-d8xzc    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-dqwrd    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-h5jvr    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-j6blz    1/1     Running   0          3m6s
pod/nginx-7c48cc69bf-jcxpf    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-jgwf7    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-k7777    1/1     Running   0          3m8s
pod/nginx-7c48cc69bf-ptsbz    1/1     Running   0          3m10s
pod/nginx-7c48cc69bf-q8rp5    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-rmlzs    1/1     Running   0          3m12s
pod/nginx-7c48cc69bf-rwc7s    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-tpbdm    1/1     Running   0          3m6s
pod/nginx-7c48cc69bf-ttmc8    1/1     Running   0          3m5s
pod/nginx-7c48cc69bf-vzxl8    1/1     Running   0          3m12s
pod/nginx-7c48cc69bf-xwxdv    1/1     Running   0          3m21s
pod/nginx-7c48cc69bf-zgv4m    1/1     Running   0          3m21s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   59m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   18/18   18           18          40m

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-7c48cc69bf   18        18        18      3m21s
replicaset.apps/nginx-8689885776   0         0         0       33m
replicaset.apps/nginx-cd55c47f5    0         0         0       40m
```

#kubectl rollout undo deploy nginx --to-revision=2
(this command rollback to nginx:1.7.8 version and terminates nginx:1.7.9 version nginx.also revision 2 will change to revision4).please verify in below screenshot)

```
cluster1-controlplane ~ ➜  kubectl rollout history deploy nginx
deployment.apps/nginx
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
3         kubectl set image deploy nginx nginx=nginx:1.7.9 --record=true


cluster1-controlplane ~ ➜  kubectl rollout undo deploy nginx --to-revison=2
error: unknown flag: --to-revison
See 'kubectl rollout undo --help' for usage.

cluster1-controlplane ~ ✖ kubectl rollout undo deploy nginx --to-revision=2
deployment.apps/nginx rolled back

cluster1-controlplane ~ ➜  kubectl rollout history deploy nginx
deployment.apps/nginx
REVISION  CHANGE-CAUSE
1         <none>
3         kubectl set image deploy nginx nginx=nginx:1.7.9 --record=true
4         <none>


cluster1-controlplane ~ ➜
```

Below screenshot shows we have successfully rollback nginx:1.7.8 version.

```
cluster1-controlplane ~ ➜  kubectl describe pod/nginx-8689885776-hx6mb
Name:             nginx-8689885776-hx6mb
Namespace:        default
Priority:         0
Service Account:  default
Node:             cluster1-node02/192.28.32.10
Start Time:       Thu, 20 Apr 2023 11:49:05 -0400
Labels:           app=nginx
                  pod-template-hash=8689885776
Annotations:      <none>
Status:           Running
IP:               10.244.64.4
IPs:
  IP:           10.244.64.4
Controlled By:  ReplicaSet/nginx-8689885776
Containers:
  nginx:
    Container ID:   containerd://ef0e75fb61e96f0bfd3c78defe96c67207789c4b0c1d71e2525600b8a95735c8
    Image:          nginx:1.7.8
    Image ID:       sha256:9362f6342d6920db0a6a2ccc1b13dd2a99cc6b421055940b48c468cda3494126
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 20 Apr 2023 11:49:07 -0400
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-mcvvp (ro)
```

****

Persistent volume:
#kubectl explain pv.spec

```
File  Actions  Edit  View  Help
root@22fa3c5b801c:~# kubectl explain pv.spec
KIND:       PersistentVolume
VERSION:    v1

FIELD: spec <PersistentVolumeSpec>

DESCRIPTION:
    spec defines a specification of a persistent volume owned by the cluster.
    Provisioned by an administrator. More info:
    https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistent-volumes
    PersistentVolumeSpec is the specification of a persistent volume.

FIELDS:
  accessModes   <[]string>
    accessModes contains all ways the volume can be mounted. More info:
    https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes

  awsElasticBlockStore  <AWSElasticBlockStoreVolumeSource>
    awsElasticBlockStore represents an AWS Disk resource that is attached to a
    kubelet's host machine and then exposed to the pod. More info:
    https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

  azureDisk     <AzureDiskVolumeSource>
    azureDisk represents an Azure Data Disk mount on the host and bind mount to
    the pod.

  azureFile     <AzureFilePersistentVolumeSource>
    azureFile represents an Azure File Service mount on the host and bind mount
    to the pod.

  capacity      <map[string]Quantity>
    capacity is the description of the persistent volume's resources and
    capacity. More info:
```

#kubectl explain pv.spec.awsElasticBlockStore

```
root@22fa3c5b801c:~# kubectl explain pv.spec.awsElasticBlockStore
KIND:       PersistentVolume
VERSION:    v1

FIELD: awsElasticBlockStore <AWSElasticBlockStoreVolumeSource>

DESCRIPTION:
    awsElasticBlockStore represents an AWS Disk resource that is attached to a
    kubelet's host machine and then exposed to the pod. More info:
    https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
    Represents a Persistent Disk resource in AWS.

    An AWS EBS disk must exist before mounting to a container. The disk must
    also be in the same AWS zone as the kubelet. An AWS EBS disk can only be
    mounted as read/write once. AWS EBS volumes support ownership management and
    SELinux relabeling.

FIELDS:
  fsType        <string>
    fsType is the filesystem type of the volume that you want to mount. Tip:
    Ensure that the filesystem type is supported by the host operating system.
    Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if
    unspecified. More info:
    https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

  partition     <integer>
    partition is the partition in the volume that you want to mount. If omitted,
    the default is to mount by volume name. Examples: For volume /dev/sda1, you
    specify the partition as "1". Similarly, the volume partition for /dev/sda
    is "0" (or you can leave the property empty).
```

There is no imperative command to automatic generate the template.we need to write pv.yaml file manually.

root@22fa3c5b801c:~# cat pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mypv
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: normal
  capacity:
    storage: 1G
  hostPath:
    path: /opt
```

```
root@22fa3c5b801c:~# kubectl get pv
No resources found
root@22fa3c5b801c:~# kubectl get pvc
No resources found in default namespace.
root@22fa3c5b801c:~#
```

Abovescreenshot shows persistent volume is not specific of namespace.

root@22fa3c5b801c:~# kubectl apply -f pv.yaml

```
root@22fa3c5b801c:~# kubectl get pvc
No resources found in default namespace.
root@22fa3c5b801c:~# kubectl apply -f pv.yaml
persistentvolume/mypv created
```

root@22fa3c5b801c:~# kubectl get pv

```
root@22fa3c5b801c:~# kubectl get pv
NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
mypv   1G         RWX            Retain           Available           normal                  72s
root@22fa3c5b801c:~#
```

The persistent volume is available.if we want to bind it we need to create pvc.yaml (persistant volume claim)

root@22fa3c5b801c:~# cat pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
    - ReadWriteMany
  storageClasseName: normal
  resources:
    requests:
      storage: 1G
```

root@22fa3c5b801c:~#

```
root@22fa3c5b801c:~# cat pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
    - ReadWriteMany
  storageClasseName: normal
  resources:
    requests:
       storage: 1G
root@22fa3c5b801c:~#
```

root@22fa3c5b801c:~# kubectl apply -f pvc.yaml

Below screenshot shows in status is bound now.

```
root@22fa3c5b801c:~# vim pvc.yaml
root@22fa3c5b801c:~# kubectl apply -f pvc.yaml
persistentvolumeclaim/mypvc created
root@22fa3c5b801c:~# kubectl get pv
NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM           STORAGECLASS   REASON   AGE
mypv   1G         RWX            Retain           Bound    default/mypvc   normal                  15m
root@22fa3c5b801c:~#
```

```
root@22fa3c5b801c:~# kubectl apply -f pvc.yaml
persistentvolumeclaim/mypvc created
root@22fa3c5b801c:~# kubectl get pv
NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM           STORAGECLASS   REASON   AGE
mypv   1G         RWX            Retain           Bound    default/mypvc   normal                  15m
root@22fa3c5b801c:~# kubectl get pvc
NAME    STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mypvc   Bound    mypv     1G         RWX            normal         2m39s
root@22fa3c5b801c:~#
```

****

Now we will learn how to use pvc with kubernates pods.

root@22fa3c5b801c:~# kubectl run nginx --image=nginx --port=80 --dry-run=client -o yaml > pod.yaml

```
root@22fa3c5b801c:~# kubectl run  nginx --image=nginx --port=80 --dry-run=client -o yaml > pod.yaml
```

Using above command pod.yaml file is created

root@22fa3c5b801c:~# cat pod.yaml

apiVersion: v1

kind: Pod

metadata:

 creationTimestamp: null

 labels:

  run: nginx

 name: nginx

```
spec:
  containers:
  - image: nginx
    name: nginx
    ports:
    - containerPort: 80
    resources: {}
    volumeMounts:
    - name: myvol
      mountPath: /etc/lala
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  volumes:
  - name: myvol
    persistentVolumeClaim:
      claimName: mypvc
status: {}
```

Bold yaml file configuration we added for persistent volume.

```
root@22fa3c5b801c:~#
```

```
root@22fa3c5b801c:~# cat pod.yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
    ports:
    - containerPort: 80
    resources: {}
    volumeMounts:
      - name: myvol
        mountPath: /etc/lala
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  volumes:
    - name: myvol
      persistentVolumeClaim:
        claimName: mypvc
status: {}
root@22fa3c5b801c:~#
```

root@22fa3c5b801c:~# kubectl apply -f pod.yaml
This command created the pods.

```
root@22fa3c5b801c:~# vim pod.yaml
root@22fa3c5b801c:~# kubectl apply -f pod.yaml
pod/nginx created
root@22fa3c5b801c:~#
```

Below screenshot shows the container is mounted at /etc/lala/
root@22fa3c5b801c:~# kubectl exec -it nginx bash

root@nginx:/# cd /etc/lala

```
root@22fa3c5b801c:~# kubectl exec -it nginx bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
root@nginx:/# cd /etc/lala
root@nginx:/etc/lala# ls
cni  containerd
root@nginx:/etc/lala# ls -lrth
total 8.0K
drwxr-xr-x 3 root root 4.0K Mar 30 01:42 cni
drwx--x--x 4 root root 4.0K Mar 30 06:36 containerd
root@nginx:/etc/lala#
```

Note: binding PV to PVC is one to one but we can bing one PVC to many pods (mean we can share PVC. volume and volumeMounts are list so we can attach as many volume in pods.

****

HELM chart

Chart.yaml contains
apiVersion: v2
name: mychart
description: A Helm chart for my application
version: 1.0.0
appVersion: 1.0.0

`Values.yaml`

```
# Default values for mychart.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.
replicaCount: 1
image:
  repository: nginx
  tag: "1.19.10"
  pullPolicy: IfNotPresent
service:
  name: nginx
  type: ClusterIP
  port: 80
```

The `templates/` folder is a required directory in a Helm chart. It contains the Kubernetes manifest templates used to create the Kubernetes resources for your application.

templates/

- `deployment.yaml`
- `service.yaml`

- `configmap.yaml`
- `secret.yaml`

root@22fa3c5b801c:~# helm create myapp

```
root@22fa3c5b801c:~# helm create myapp
Creating myapp
root@22fa3c5b801c:~# cd myapp
root@22fa3c5b801c:~/myapp# ls
Chart.yaml  charts  templates  values.yaml
root@22fa3c5b801c:~/myapp#
```

We create deploy.yaml using below command
root@22fa3c5b801c:~# kubectl create deploy nginx --image=nginx:latest --port=80 --dry-run=client -o yaml > deploy.yaml

Copy to template folder
root@22fa3c5b801c:~/myapp/templates# cp /root/deploy.yaml

```
root@22fa3c5b801c:~# kubectl create deploy nginx --image=nginx:latest --port=80 --dry-run=client -o yaml > deploy.yaml
root@22fa3c5b801c:~# ls
configfile  deploy.yaml  myapp  pod.yaml  pv.yaml  pvc.yaml  snap
root@22fa3c5b801c:~# pwd
/root
root@22fa3c5b801c:~# cd myapp/
root@22fa3c5b801c:~/myapp# ls
Chart.yaml  templates
root@22fa3c5b801c:~/myapp# cd templates/
root@22fa3c5b801c:~/myapp/templates# cp /root/deploy.yaml .
root@22fa3c5b801c:~/myapp/templates# ls
deploy.yaml
```

update the deploy.yaml file.below code is
root@22fa3c5b801c:~/myapp/templates# cat deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: {{.Values.app.name}}
  name: {{.Values.app.name}}
spec:
  replicas: {{.Values.app.replicas}}
  selector:
    matchLabels:
```

```
      app: {{.Values.app.name}}
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: {{.Values.app.name}}
    spec:
      containers:
      - image: {{.Values.app.image}}
        name: {{.Values.app.name}}
        ports:
        - containerPort: {{.Values.app.port}}
        resources: {}
status: {}
```

```
root@22fa3c5b801c:~/myapp/templates# cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: {{.Values.app.name}}
  name: {{.Values.app.name}}
spec:
  replicas: {{.Values.app.replicas}}
  selector:
    matchLabels:
      app: {{.Values.app.name}}
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: {{.Values.app.name}}
    spec:
      containers:
      - image: {{.Values.app.image}}
        name: {{.Values.app.name}}
        ports:
        - containerPort: {{.Values.app.port}}
        resources: {}
status: {}
root@22fa3c5b801c:~/myapp/templates# cat   ../values.yaml
app:
```

Update values.yaml file.below is values.yaml file

root@22fa3c5b801c:~/myapp# cat values.yaml

app:

  name: myapp

  image: nginx

  replicas: 10

  port: 8080

  type: NodePort

```
root@22fa3c5b801c:~/myapp# cat values.yaml
app:
    name: myapp
    image: nginx
    replicas: 10
    port: 8080
    type: NodePort
root@22fa3c5b801c:~/myapp#
```

root@22fa3c5b801c:~/myapp/templates# cat service.yaml
apiVersion: v1
kind: Service
metadata:
 name: {{.Values.app.name}}
spec:
 selector:
  app: {{.Values.app.name}}
 type: {{.Values.app.type}}
 ports:
  - port: {{.Values.app.port}}
    targetPort: {{.Values.app.port}}

root@22fa3c5b801c:~/myapp/templates# cd .

```
root@22fa3c5b801c:~/myapp/templates# cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: {{.Values.app.name}}
spec:
  selector:
    app: {{.Values.app.name}}
  type: {{.Values.app.type}}
  ports:
    - port: {{.Values.app.port}}
      targetPort: {{.Values.app.port}}

root@22fa3c5b801c:~/myapp/templates# cd ..
```

Now we can run below helm template command.
root@22fa3c5b801c:~# helm template myapp myapp/

The output will be like

```
root@22fa3c5b801c:~# helm template myapp myapp/
---
# Source: myapp/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: myapp
spec:
  selector:
    app: myapp
  type: NodePort
  ports:
    - port: 8080
      targetPort: 8080
---
# Source: myapp/templates/deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: myapp
  name: myapp
spec:
  replicas: 10
  selector:
    matchLabels:
      app: myapp
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: myapp
    spec:
      containers:
      - image: nginx
        name: myapp
```

```
    ports:
    - containerPort: 8080
    resources: {}
status: {}
root@22fa3c5b801c:~# kubectl get all
```

```
root@22fa3c5b801c:~# helm template myapp myapp/
---
# Source: myapp/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: myapp
spec:
  selector:
    app: myapp
  type: NodePort
  ports:
    - port: 8080
      targetPort: 8080
---
# Source: myapp/templates/deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: myapp
  name: myapp
spec:
  replicas: 10
  selector:
    matchLabels:
      app: myapp
  strategy: {}
  template:
    metadata:
```

Below screenshot is of above screenshot continuation.

```
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: myapp
    spec:
      containers:
      - image: nginx
        name: myapp
        ports:
        - containerPort: 8080
        resources: {}
status: {}
root@22fa3c5b801c:~# kubectl get all
```

Before running helm install command.

```
root@22fa3c5b801c:~# kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   9h
root@22fa3c5b801c:~# kubectl get po
No resources found in default namespace.
```

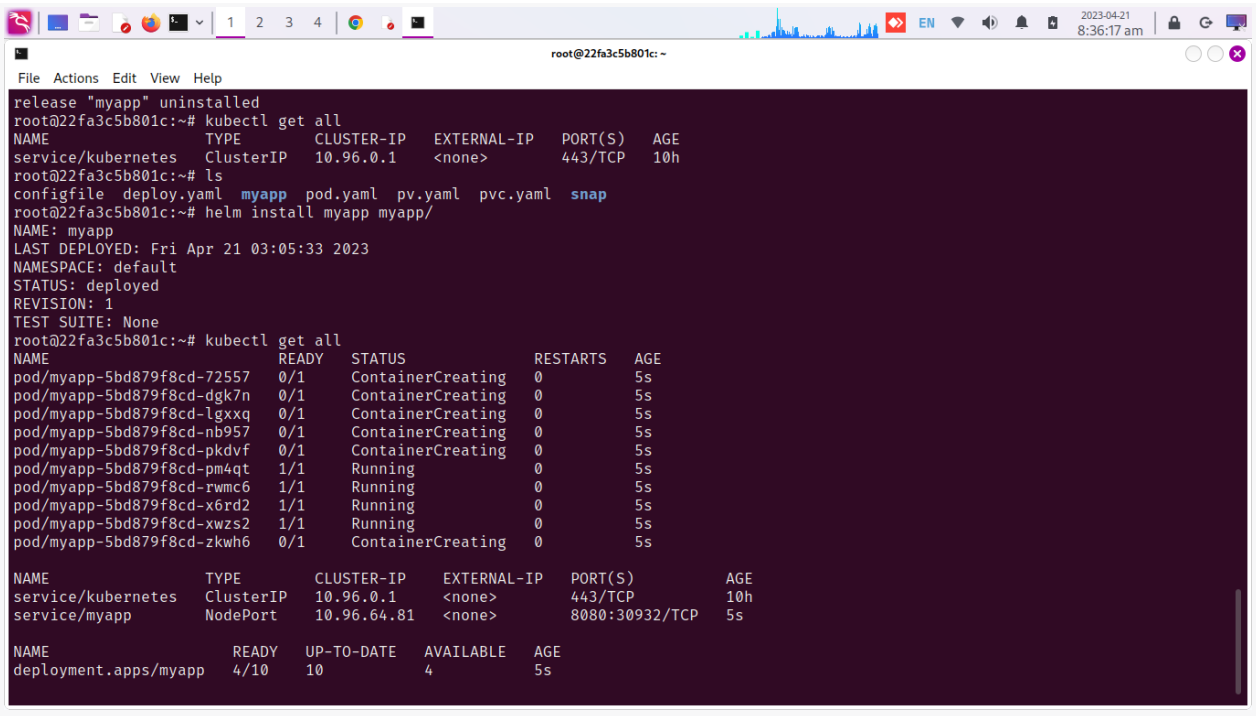
After running
root@22fa3c5b801c:~# helm install myapp myapp/

After running the command
root@22fa3c5b801c:~# helm uninstall myapp

```
root@22fa3c5b801c:~# kubectl get all
NAME                            READY   STATUS    RESTARTS   AGE
pod/myapp-5bd879f8cd-47wf6      1/1     Running   0          19m
pod/myapp-5bd879f8cd-97d7m      1/1     Running   0          19m
pod/myapp-5bd879f8cd-fbnq4      1/1     Running   0          19m
pod/myapp-5bd879f8cd-hgc4m      1/1     Running   0          19m
pod/myapp-5bd879f8cd-rkcl9      1/1     Running   0          19m
pod/myapp-5bd879f8cd-rnrk8      1/1     Running   0          19m
pod/myapp-5bd879f8cd-sj772      1/1     Running   0          19m
pod/myapp-5bd879f8cd-srx9k      1/1     Running   0          19m
pod/myapp-5bd879f8cd-w5jlv      1/1     Running   0          19m
pod/myapp-5bd879f8cd-wdhrv      1/1     Running   0          19m

NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP          10h
service/myapp        NodePort    10.96.232.16   <none>        8080:30597/TCP   19m

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/myapp   10/10   10           10          19m

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/myapp-5bd879f8cd   10        10        10      19m
root@22fa3c5b801c:~# helm uninstall myapp
release "myapp" uninstalled
root@22fa3c5b801c:~# kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   10h
root@22fa3c5b801c:~#
```

Again we executed
root@22fa3c5b801c:~# helm install myapp myapp/

Now deployment is done



```
root@22fa3c5b801c:~# helm install myapp myapp/
NAME: myapp
LAST DEPLOYED: Fri Apr 21 02:44:00 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
root@22fa3c5b801c:~# kubectl get all
NAME                          READY   STATUS    RESTARTS   AGE
pod/myapp-5bd879f8cd-47wf6    1/1     Running   0          22s
pod/myapp-5bd879f8cd-97d7m    1/1     Running   0          22s
pod/myapp-5bd879f8cd-fbnq4    1/1     Running   0          22s
pod/myapp-5bd879f8cd-hgc4m    1/1     Running   0          22s
pod/myapp-5bd879f8cd-rkcl9    1/1     Running   0          22s
pod/myapp-5bd879f8cd-rnrk8    1/1     Running   0          22s
pod/myapp-5bd879f8cd-sj772    1/1     Running   0          22s
pod/myapp-5bd879f8cd-srx9k    1/1     Running   0          22s
pod/myapp-5bd879f8cd-w5jlv    1/1     Running   0          22s
pod/myapp-5bd879f8cd-wdhrv    1/1     Running   0          22s

NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP          9h
service/myapp        NodePort    10.96.232.16   <none>        8080:30597/TCP   22s

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/myapp   10/10   10           10          22s

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/myapp-5bd879f8cd   10        10        10      22s
root@22fa3c5b801c:~# cd myapp/
```

To verify what app is managing helm.run the below command.

root@22fa3c5b801c:~# helm list

```
root@22fa3c5b801c:~# helm list
NAME    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART           APP VERSION
myapp   default         1               2023-04-21 03:05:33.947997979 +0000 UTC deployed        myapp-0.1.0     1.16.0
root@22fa3c5b801c:~#
```

If we want to upgrade the nginx then ,update the values.yaml file.

root@22fa3c5b801c:~/myapp# cat values.yaml

app:

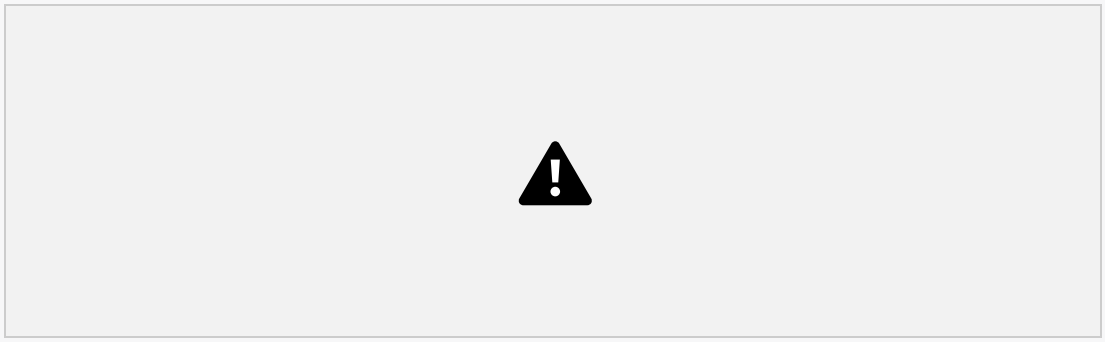
  name: myapp

  image: nginx:1.7.9

  replicas: 10

  port: 8080

  type: NodePort



Then execute the command.
root@22fa3c5b801c:~# helm upgrade myapp myapp/

```
root@22fa3c5b801c:~# helm upgrade myapp myapp/
Release "myapp" has been upgraded. Happy Helming!
NAME: myapp
LAST DEPLOYED: Fri Apr 21 03:11:41 2023
NAMESPACE: default
STATUS: deployed
REVISION: 2
TEST SUITE: None
root@22fa3c5b801c:~#
```

.