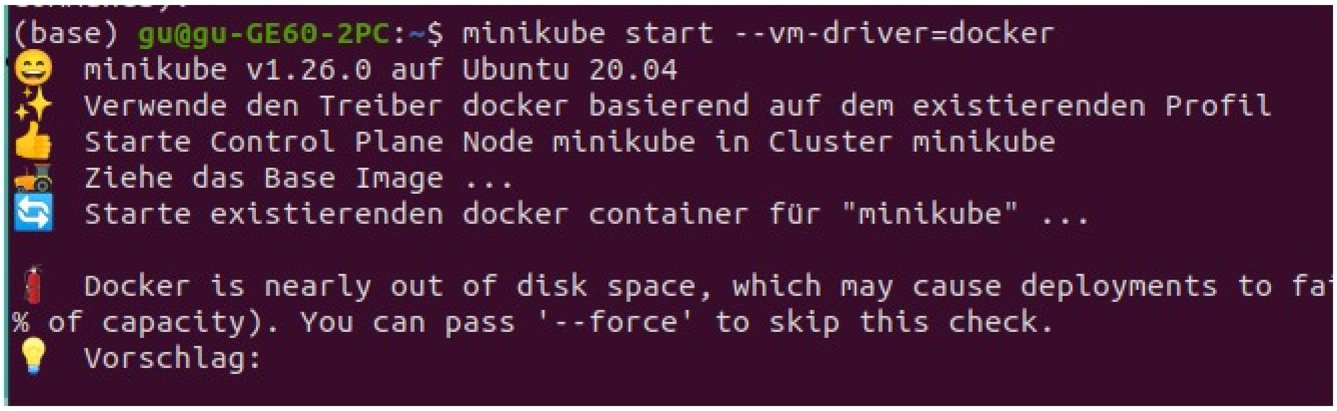
~~Set env name for minikube kubectl --~~
~~alias kubectl="minikube kubectl --"~~

1. minikube start --vm-driver=hyperkit vm hyperkit for MacOS
2. install minikube: dependency like kubectl is also installed.

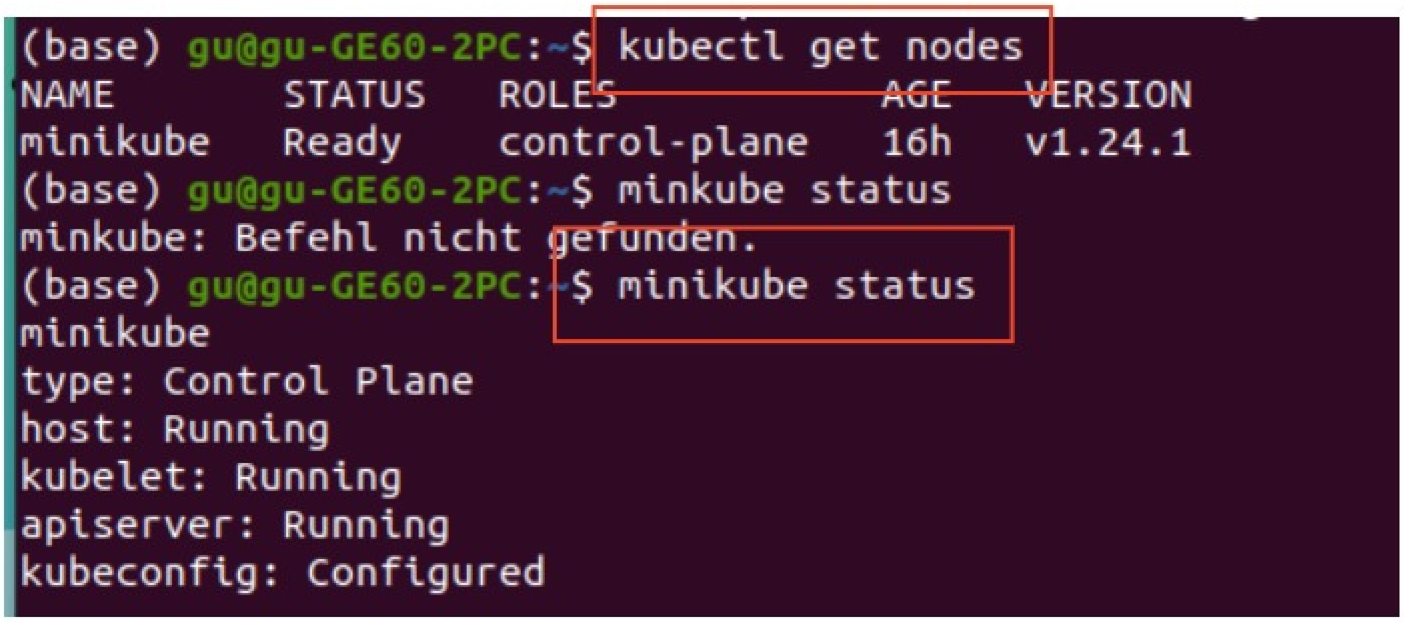
Interaction with k8s cluster: **kubectl**
Configure anyting is talk with Api Server

1. minikube start --vm-driver=docker (minikube has docker runtime preinstalled)

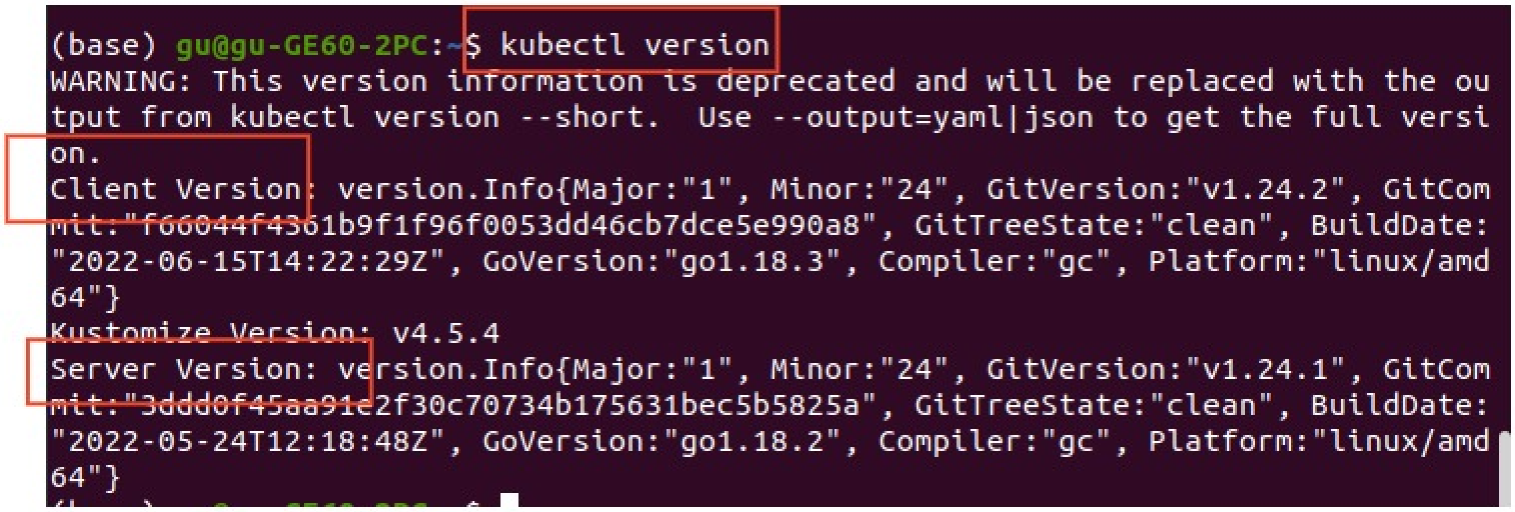


```
(base) gu@gu-GE60-2PC:~$ minikube start --vm-driver=docker
    minikube v1.26.0 auf Ubuntu 20.04
    Verwende den Treiber docker basierend auf dem existierenden Profil
    Starte Control Plane Node minikube in Cluster minikube
    Ziehe das Base Image ...
    Starte existierenden docker container für "minikube" ...

    Docker is nearly out of disk space, which may cause deployments to fa
% of capacity). You can pass '--force' to skip this check.
    Vorschlag:
```

Minikube cluster is setup and kubectl is also connected to the kube cluster

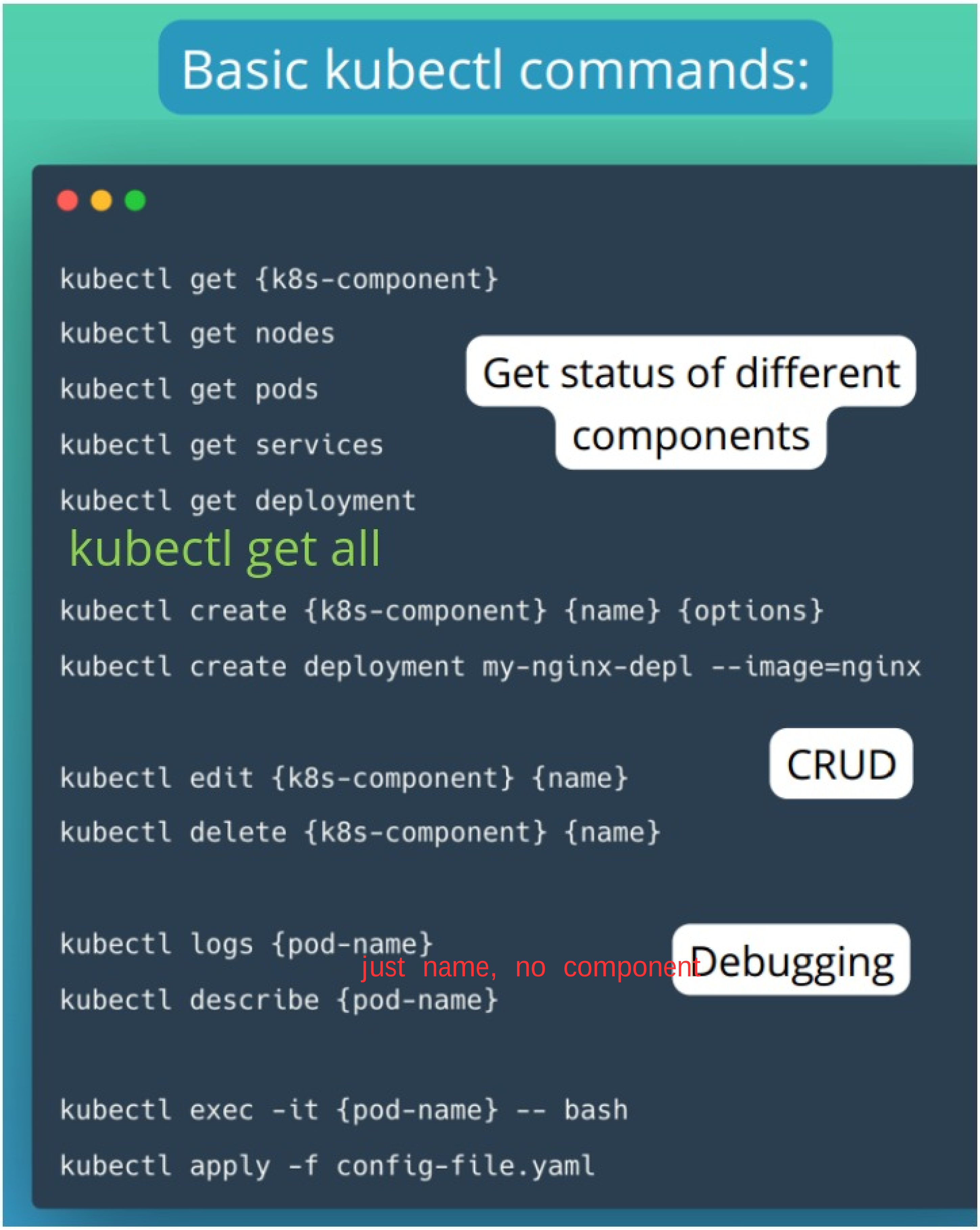


```
(base) gu@gu-GE60-2PC:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   16h   v1.24.1
(base) gu@gu-GE60-2PC:~$ minkube status
minkube: Befehl nicht gefunden.
(base) gu@gu-GE60-2PC:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```



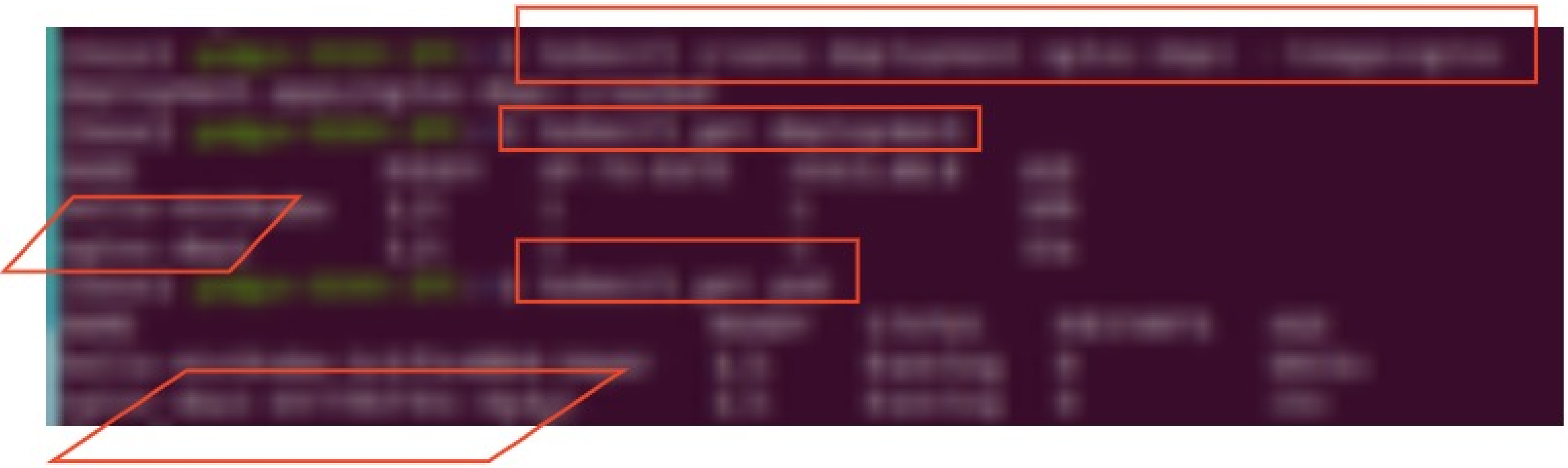
```
(base) gu@gu-GE60-2PC:~$ kubectl version
WARNING: This version information is deprecated and will be replaced with the ou
tput from kubectl version --short.  Use --output=yaml|json to get the full versi
on.
Client Version: version.Info{Major:"1", Minor:"24", GitVersion:"v1.24.2", GitCom
mit:"f66044f4361b9f1f96f0053dd46cb7dce5e990a8", GitTreeState:"clean", BuildDate:
"2022-06-15T14:22:29Z", GoVersion:"go1.18.3", Compiler:"gc", Platform:"linux/amd
64"}
Kustomize Version: v4.5.4
Server Version: version.Info{Major:"1", Minor:"24", GitVersion:"v1.24.1", GitCom
mit:"3ddd0f45aa91e2f30c70734b175631bec5b5825a", GitTreeState:"clean", BuildDate:
"2022-05-24T12:18:48Z", GoVersion:"go1.18.2", Compiler:"gc", Platform:"linux/amd
64"}
```

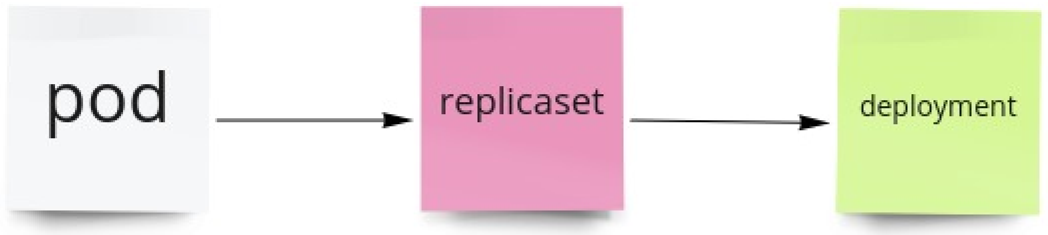
```
kubectl version
minikube status

kubectl get nodes
```



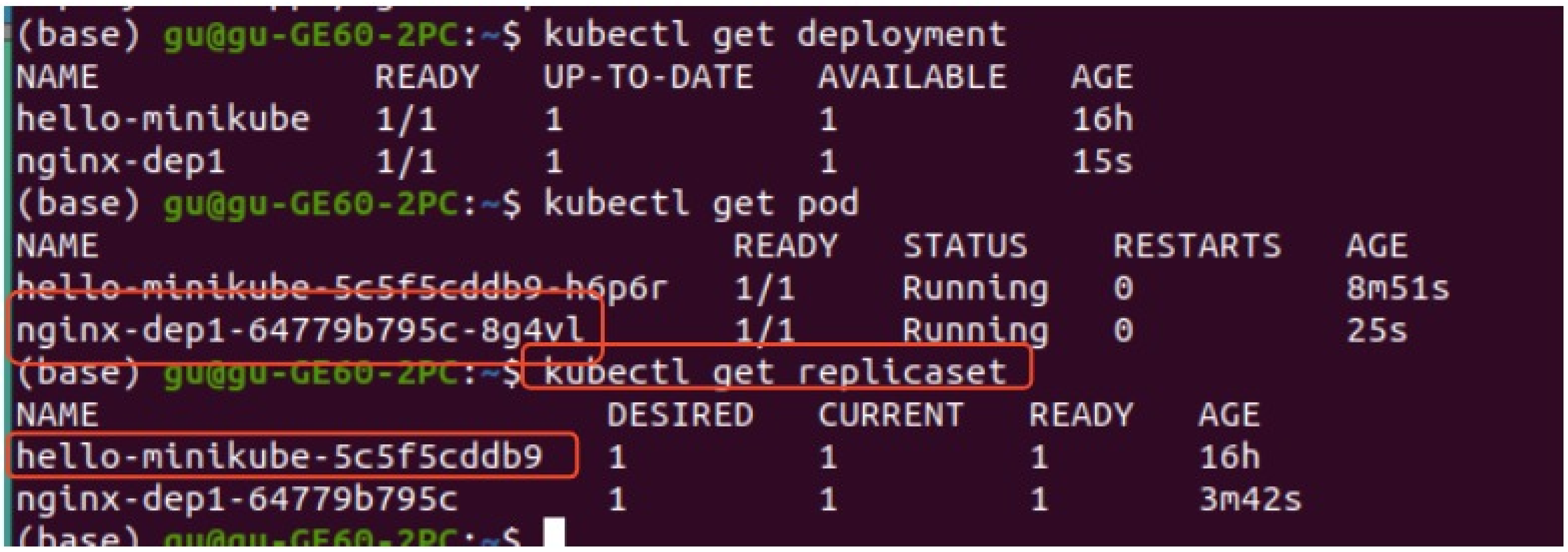
## Basic kubectl commands:

```
kubectl get {k8s-component}
kubectl get nodes
kubectl get pods
kubectl get services
kubectl get deployment
kubectl get all
kubectl create {k8s-component} {name} {options}
kubectl create deployment my-nginx-depl --image=nginx

kubectl edit {k8s-component} {name}
kubectl delete {k8s-component} {name}

kubectl logs {pod-name}
kubectl describe {pod-name}

kubectl exec -it {pod-name} -- bash
kubectl apply -f config-file.yaml
```

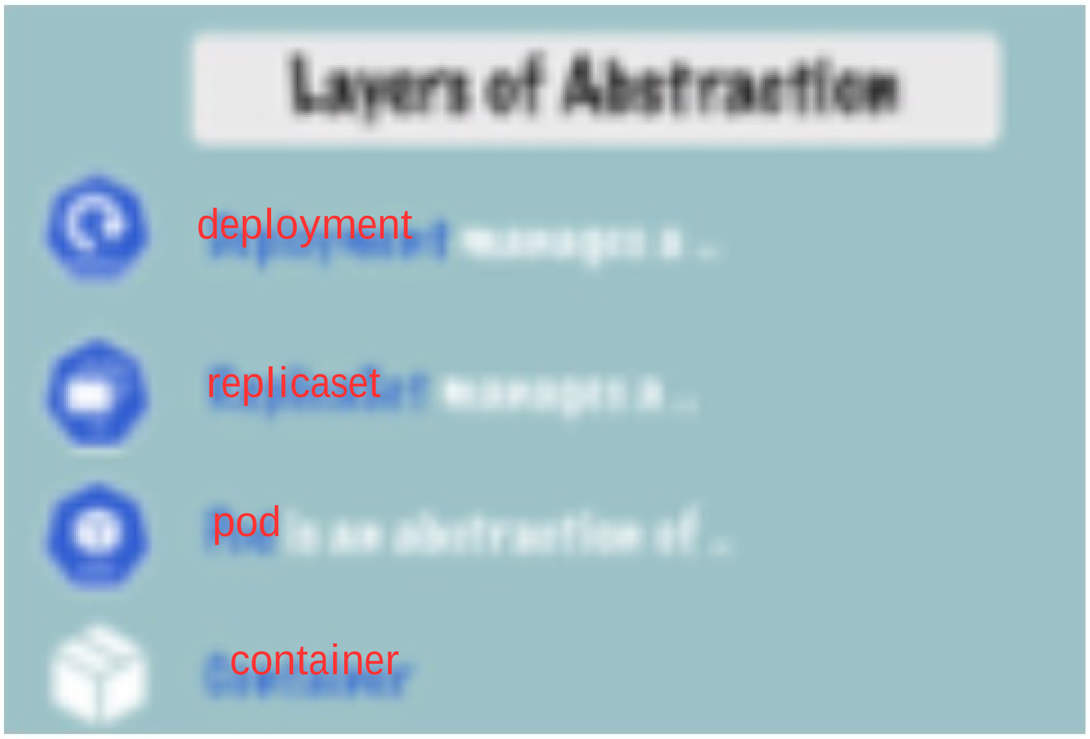




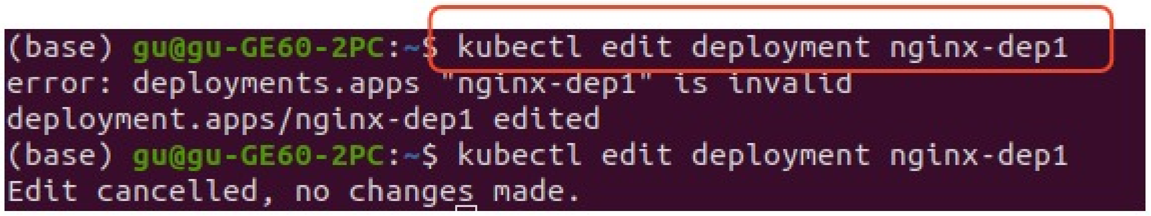
Replicaset  is managing the replicas of a pod



```
(base) gu@gu-GE60-2PC:~$ kubectl get deployment
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube   1/1     1            1           16h
nginx-dep1       1/1     1            1           15s
(base) gu@gu-GE60-2PC:~$ kubectl get pod
NAME                              READY   STATUS    RESTARTS   AGE
hello-minikube-5c5f5cddb9-h6p6r   1/1     Running   0          8m51s
nginx-dep1-64779b795c-8g4vl       1/1     Running   0          25s
(base) gu@gu-GE60-2PC:~$ kubectl get replicaset
NAME                        DESIRED   CURRENT   READY   AGE
hello-minikube-5c5f5cddb9   1         1         1       16h
nginx-dep1-64779b795c       1         1         1       3m42s
(base) gu@gu-GE60-2PC:~$
```

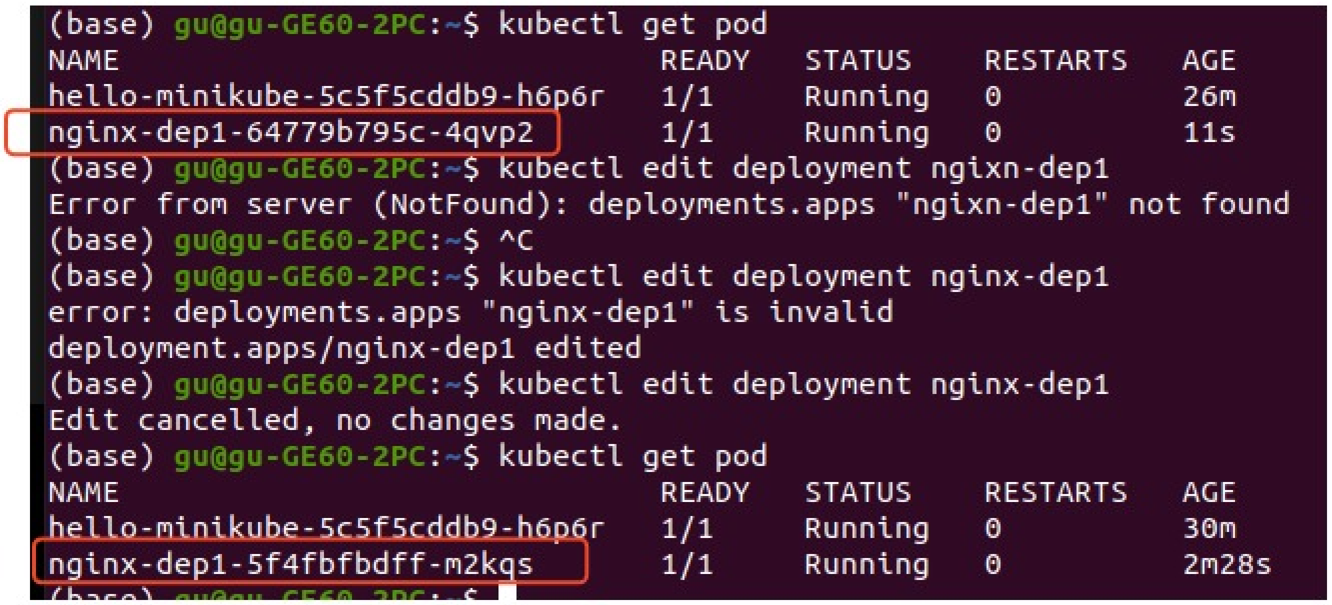


```
]-]$ kubectl edit deployment nginx-de
```

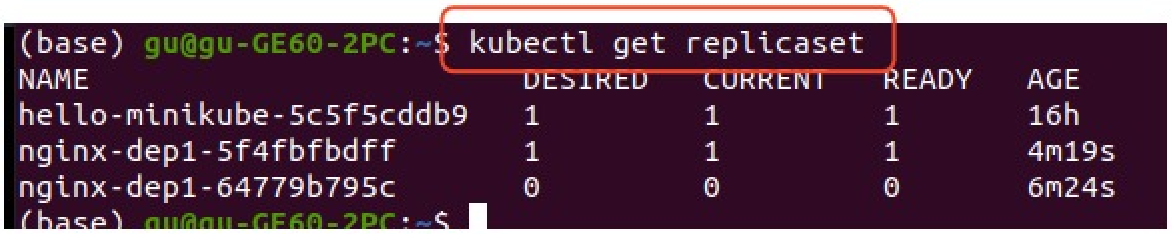


```
(base) gu@gu-GE60-2PC:~$ kubectl edit deployment nginx-dep1
error: deployments.apps "nginx-dep1" is invalid
deployment.apps/nginx-dep1 edited
(base) gu@gu-GE60-2PC:~$ kubectl edit deployment nginx-dep1
Edit cancelled, no changes made.
```

Auto-generated configuration file with default values -> change the version of nginx image->
the old pod is terminated and a new pod is created

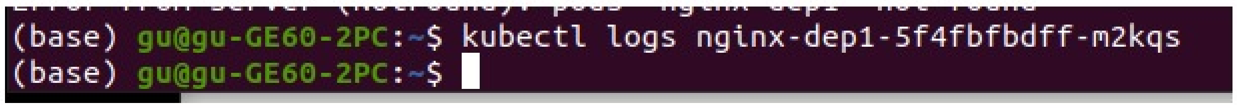


```
(base) gu@gu-GE60-2PC:~$ kubectl get pod
NAME                               READY   STATUS    RESTARTS   AGE
hello-minikube-5c5f5cddb9-h6p6r    1/1     Running   0          26m
nginx-dep1-64779b795c-4qvp2        1/1     Running   0          11s
(base) gu@gu-GE60-2PC:~$ kubectl edit deployment ngixn-dep1
Error from server (NotFound): deployments.apps "ngixn-dep1" not found
(base) gu@gu-GE60-2PC:~$ ^C
(base) gu@gu-GE60-2PC:~$ kubectl edit deployment nginx-dep1
error: deployments.apps "nginx-dep1" is invalid
deployment.apps/nginx-dep1 edited
(base) gu@gu-GE60-2PC:~$ kubectl edit deployment nginx-dep1
Edit cancelled, no changes made.
(base) gu@gu-GE60-2PC:~$ kubectl get pod
NAME                               READY   STATUS    RESTARTS   AGE
hello-minikube-5c5f5cddb9-h6p6r    1/1     Running   0          30m
nginx-dep1-5f4fbfbdff-m2kqs        1/1     Running   0          2m28s
(base) gu@gu-GE60-2PC:~$
```

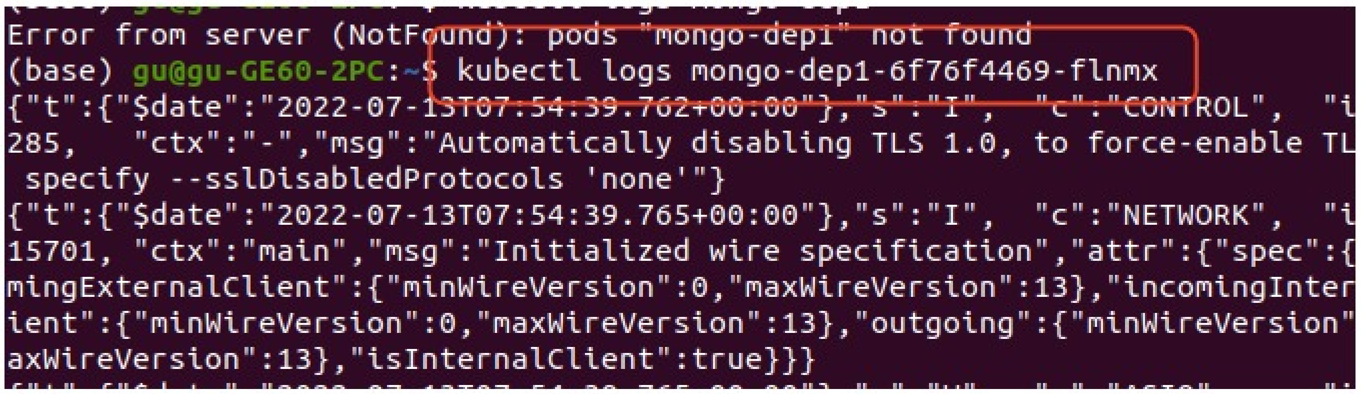


```
(base) gu@gu-GE60-2PC:~$ kubectl get replicaset
NAME                           DESIRED   CURRENT   READY   AGE
hello-minikube-5c5f5cddb9      1         1         1       16h
nginx-dep1-5f4fbfbdff          1         1         1       4m19s
nginx-dep1-64779b795c          0         0         0       6m24s
(base) gu@gu-GE60-2PC:~$
```

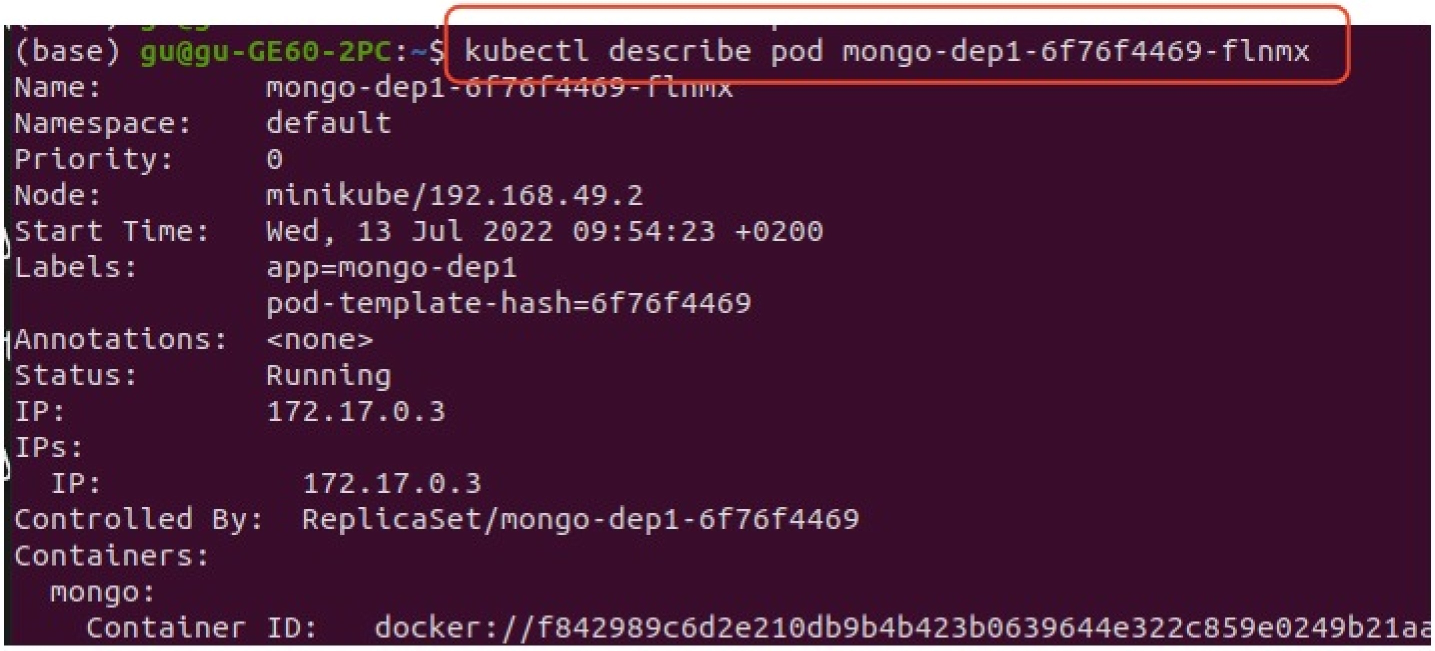
Get the  logs in a pd. (here nothing done)



```
Error from server (NotFound): pods "nginx-dep1" not found
(base) gu@gu-GE60-2PC:~$ kubectl logs nginx-dep1-5f4fbfbdff-m2kqs
(base) gu@gu-GE60-2PC:~$
```

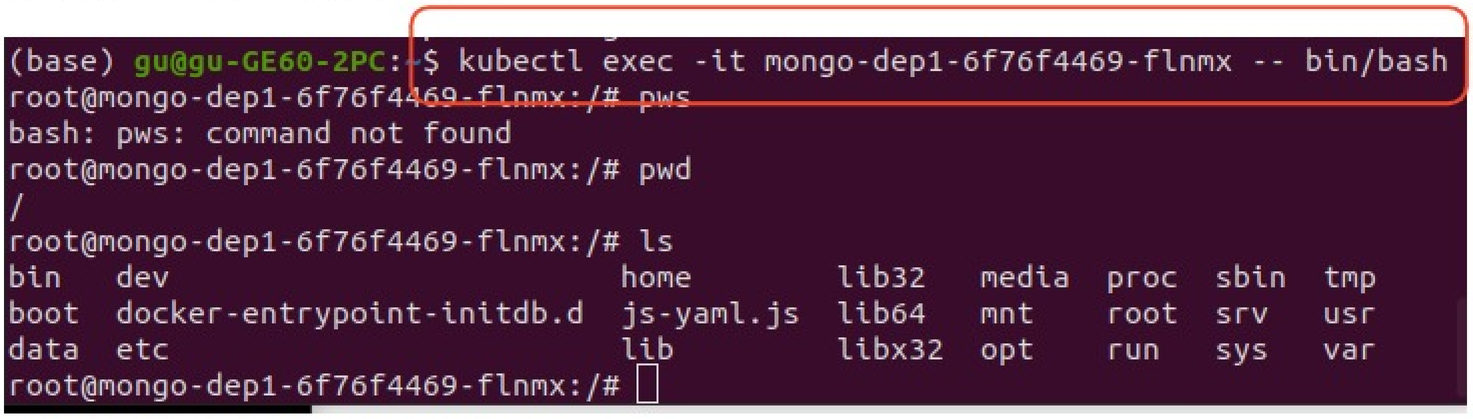


```
(base) gu@gu-GE60-2PC:~$ kubectl logs mongo-dep1
Error from server (NotFound): pods "mongo-dep1" not found
(base) gu@gu-GE60-2PC:~$ kubectl logs mongo-dep1-6f76f4469-flnmx
{"t":{"$date":"2022-07-13T07:54:39.762+00:00"},"s":"I",  "c":"CONTROL",  "i
285,   "ctx":"-","msg":"Automatically disabling TLS 1.0, to force-enable TL
 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-07-13T07:54:39.765+00:00"},"s":"I",  "c":"NETWORK",  "i
15701, "ctx":"main","msg":"Initialized wire specification","attr":{"spec":{
mingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInter
ient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion"
axWireVersion":13},"isInternalClient":true}}}
{"t":{"$date":"2022-07-13T07:54:39.765+00:00"},"s":"W",  "c":"ASIO",     "i
```



```
(base) gu@gu-GE60-2PC:~$ kubectl describe pod mongo-dep1-6f76f4469-flnmx
Name:           mongo-dep1-6f76f4469-flnmx
Namespace:      default
Priority:       0
Node:           minikube/192.168.49.2
Start Time:     Wed, 13 Jul 2022 09:54:23 +0200
Labels:         app=mongo-dep1
                pod-template-hash=6f76f4469
Annotations:    <none>
Status:         Running
IP:             172.17.0.3
IPs:
  IP:           172.17.0.3
Controlled By:  ReplicaSet/mongo-dep1-6f76f4469
Containers:
  mongo:
    Container ID:   docker://f842989c6d2e210db9b4b423b0639644e322c859e0249b21aa
```

interactive command line: -- bin/bash

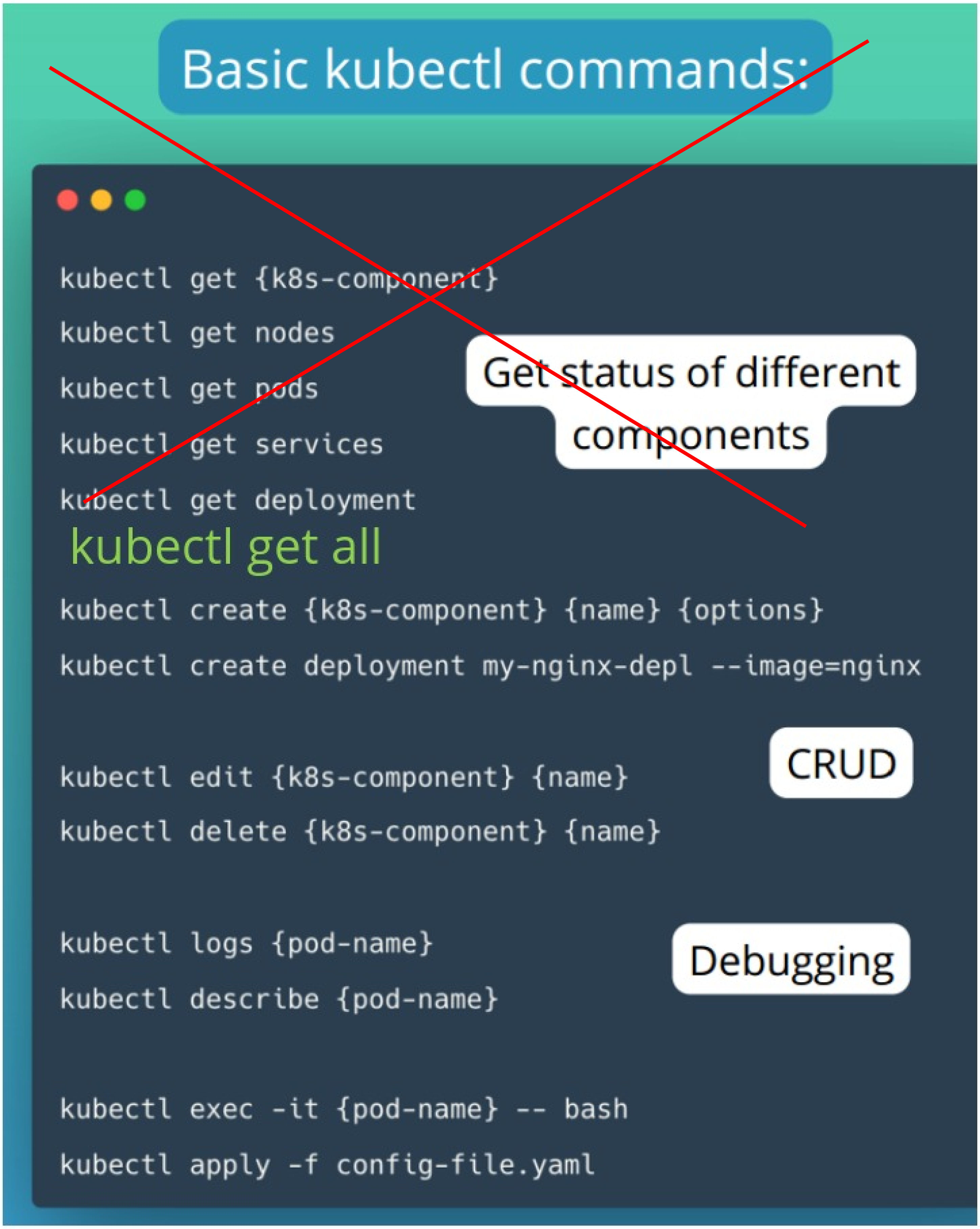


```
(base) gu@gu-GE60-2PC:~$ kubectl exec -it mongo-dep1-6f76f4469-flnmx -- bin/bash
root@mongo-dep1-6f76f4469-flnmx:/# pws
bash: pws: command not found
root@mongo-dep1-6f76f4469-flnmx:/# pwd
/
root@mongo-dep1-6f76f4469-flnmx:/# ls
bin   dev                          home        lib32   media   proc   sbin   tmp
boot  docker-entrypoint-initdb.d  js-yaml.js  lib64   mnt     root   srv    usr
data  etc                          lib         libx32  opt     run    sys    var
root@mongo-dep1-6f76f4469-flnmx:/# 
```

Write too much in the command line -> not good -> configuration file -> using kubectl apply command (takes a file)

kubectl version
minikube status

kubectl get nodes



## ~~Basic kubectl commands:~~

~~kubectl get {k8s-component}~~
~~kubectl get nodes~~
~~kubectl get pods~~
~~kubectl get services~~
~~kubectl get deployment~~

~~Get status of different components~~

kubectl get all

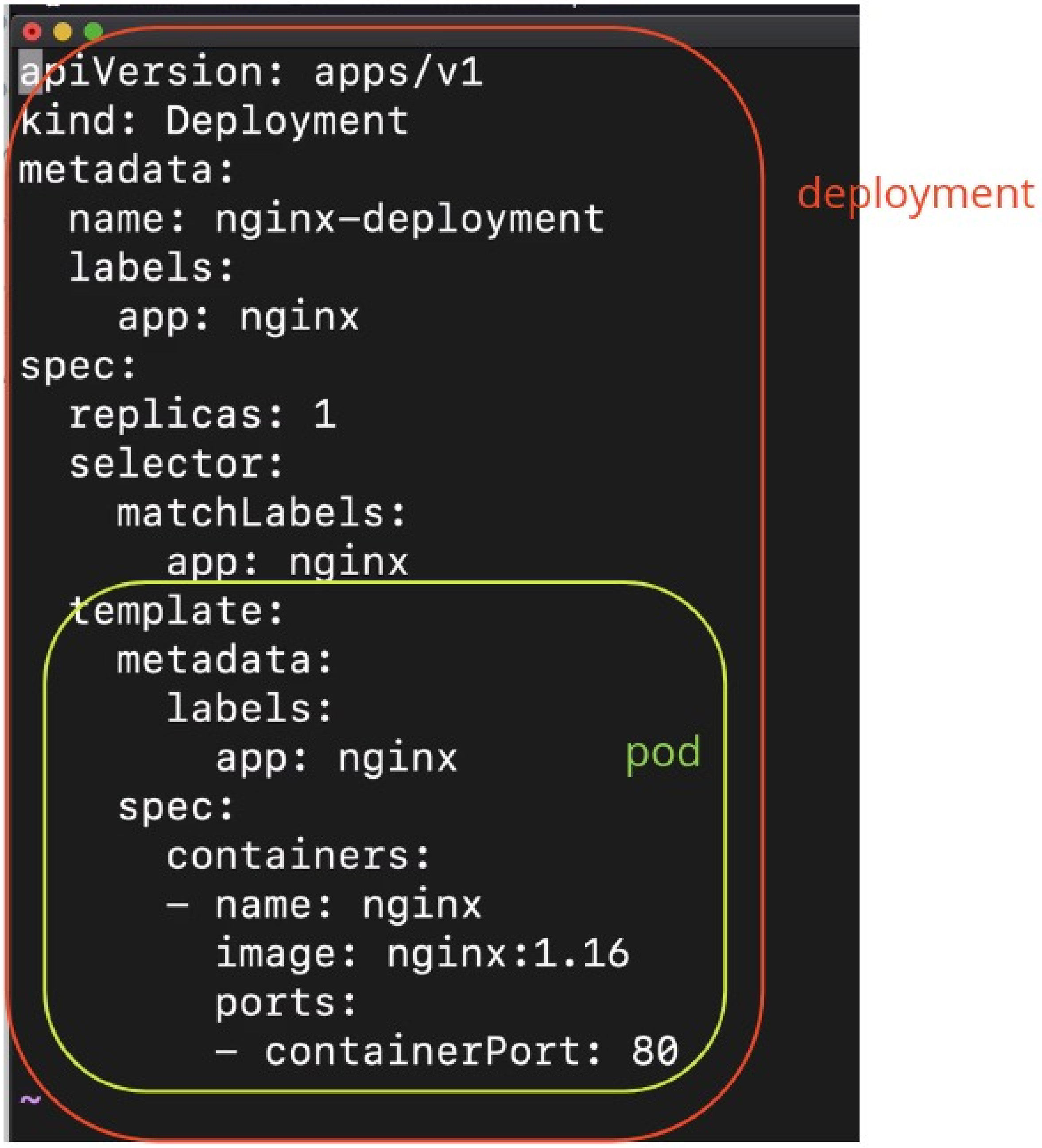
```
kubectl create {k8s-component} {name} {options}
kubectl create deployment my-nginx-depl --image=nginx

kubectl edit {k8s-component} {name}
kubectl delete {k8s-component} {name}
```

CRUD

```
kubectl logs {pod-name}
kubectl describe {pod-name}
```

Debugging

```
kubectl exec -it {pod-name} -- bash
kubectl apply -f config-file.yaml
```

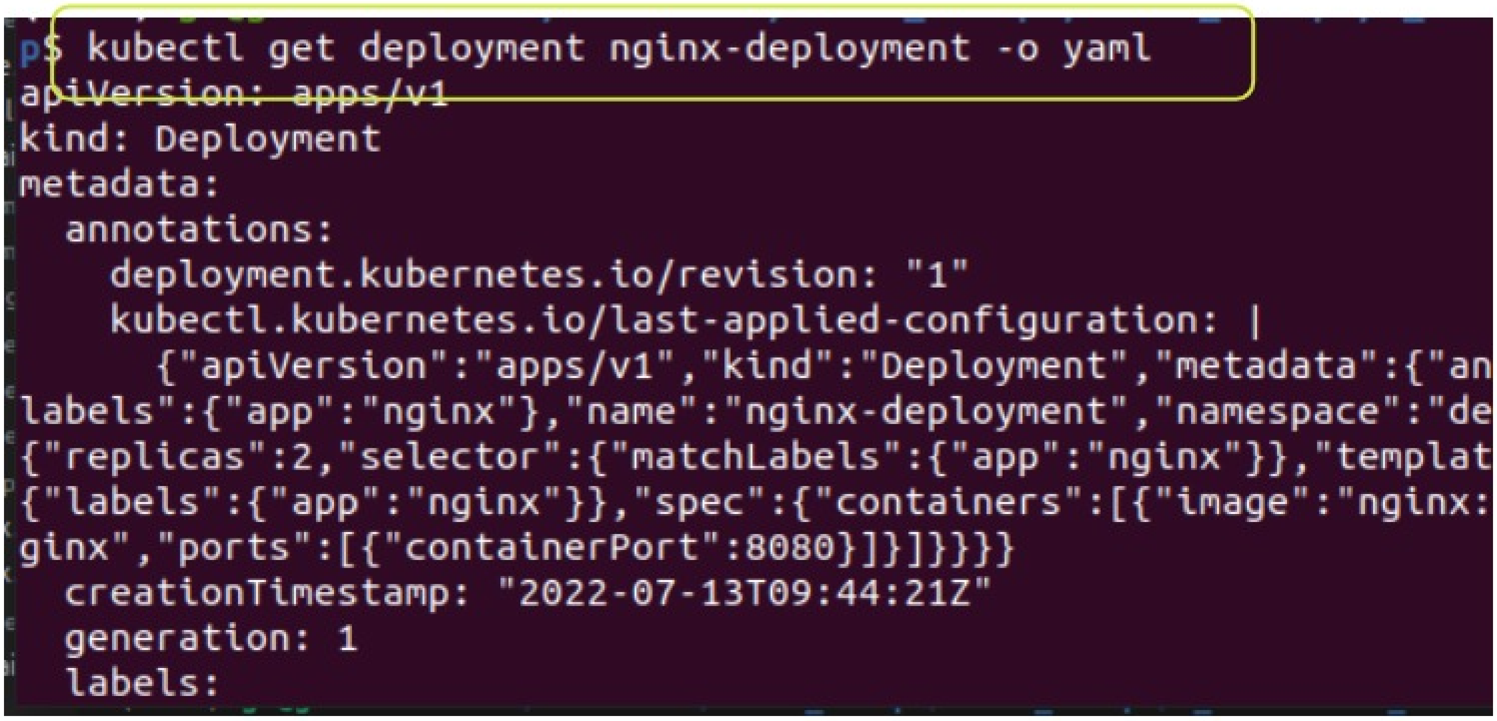


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.16
        ports:
        - containerPort: 80
~
```

With kubectl apply you can create or update a deployment
update config-file -> then kubectl apply again with same name ->
kubectl get deployment not change , but there are two pods, old one and new one.

status in a yaml format

use get deployment and output as a yaml file



```
p$ kubectl get deployment nginx-deployment -o yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"apps/v1","kind":"Deployment","metadata":{"an
labels":{"app":"nginx"},"name":"nginx-deployment","namespace":"de
{"replicas":2,"selector":{"matchLabels":{"app":"nginx"}},"templat
{"labels":{"app":"nginx"}},"spec":{"containers":[{"image":"nginx:
ginx","ports":[{"containerPort":8080}]}]}}}}
  creationTimestamp: "2022-07-13T09:44:21Z"
  generation: 1
  labels:
```

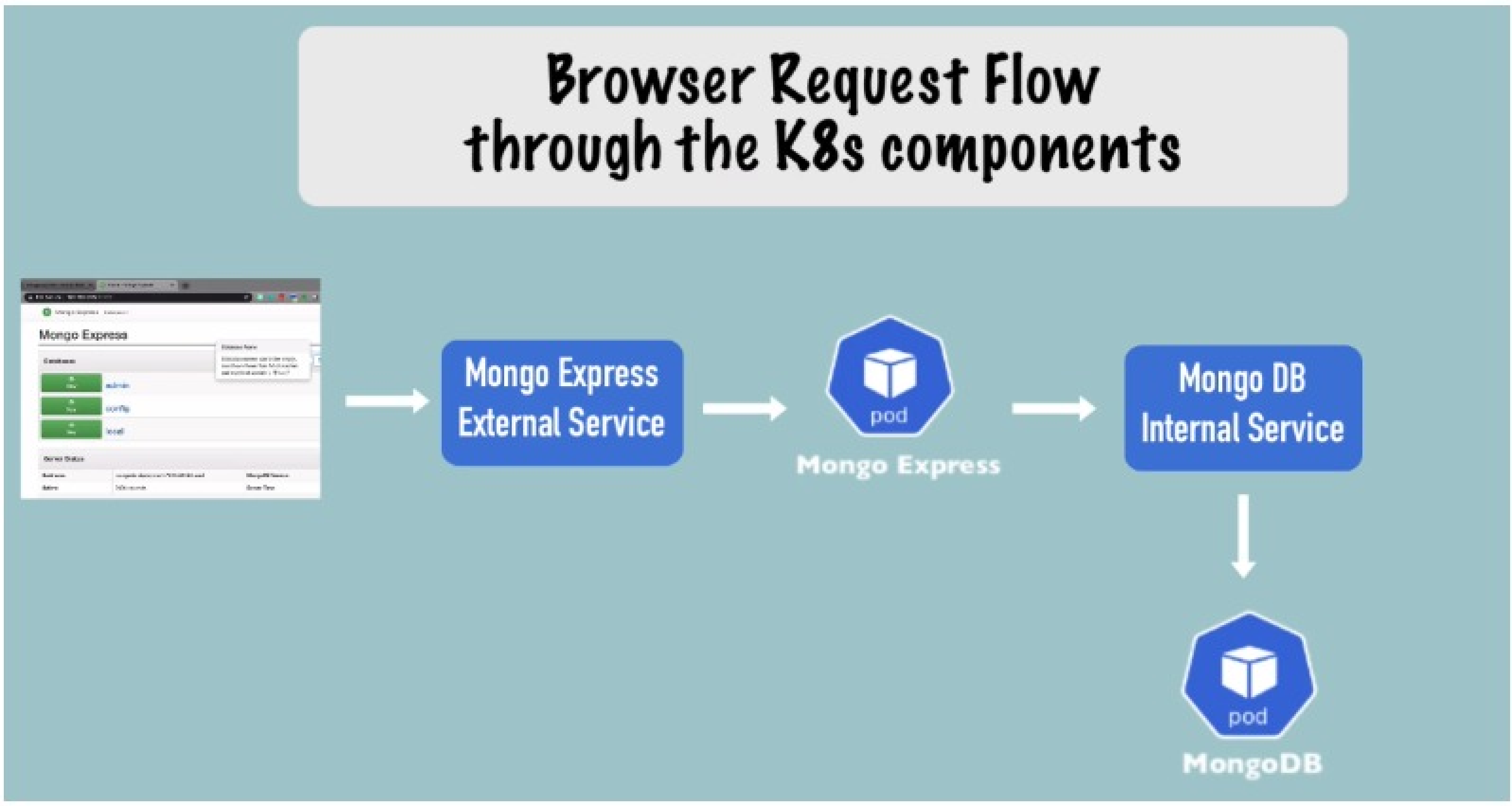
If you want to deploy another deployment, the generated yaml must be cleaned, then use as blueprint

## Demo project

**mongoDB** : internal service (no external request)

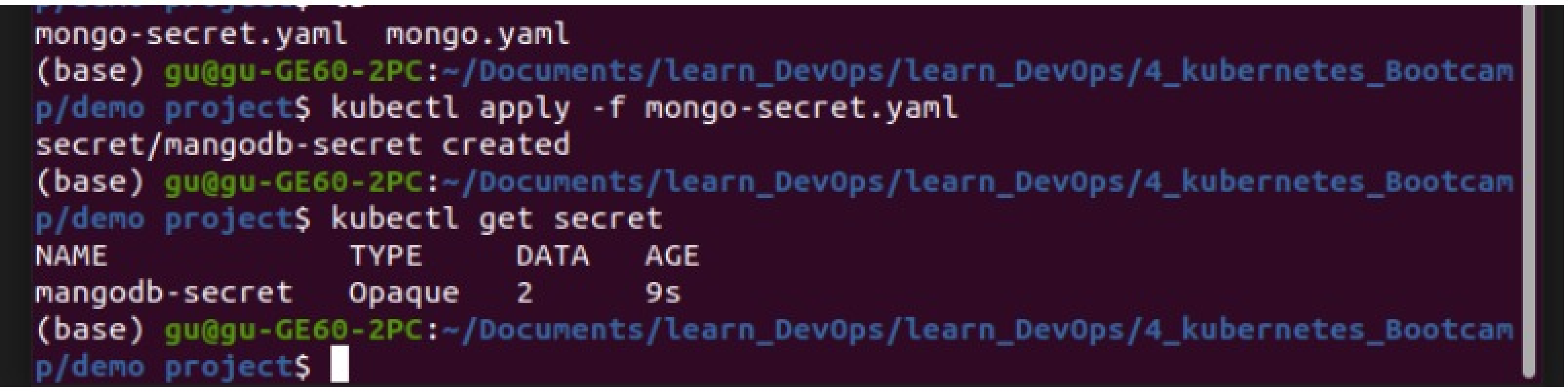
**mongo Express** (database url, connect to mongodb, authenticate) -> deployment,yaml (configMap(db url), secret(DB User, DB pwd))

mongoDB excess from external: External serivce



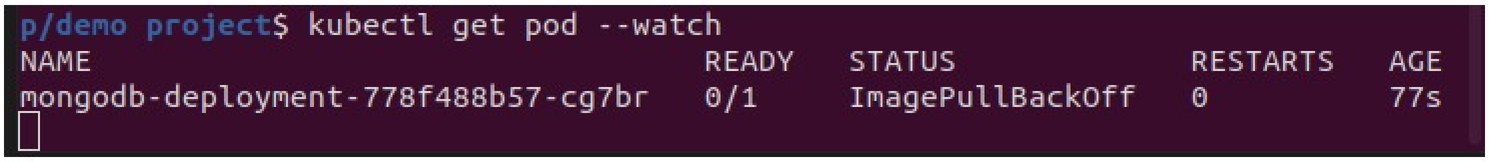
Browser Request Flow through the K8s components
Mongo Express
Mongo Express External Service
pod
Mongo Express
Mongo DB Internal Service
pod
MongoDB

1. create a mongo DB deployment
2. create secret for mongo user and mongo passward
    a. how to create user and passward text
3. ACTUNG: first deploy secret then mongdb so that you can reference it.

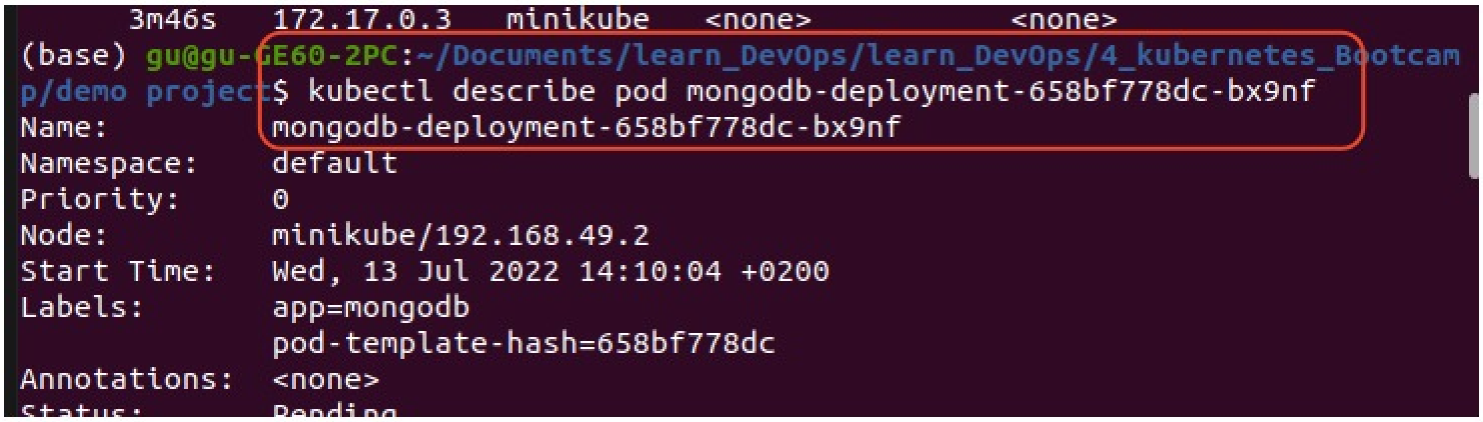


```
mongo-secret.yaml  mongo.yaml
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ kubectl apply -f mongo-secret.yaml
secret/mangodb-secret created
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ kubectl get secret
NAME             TYPE     DATA   AGE
mangodb-secret   Opaque   2      9s
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ 
```

Then we can deploy mongo



```
p/demo project$ kubectl get pod --watch
NAME                                  READY   STATUS             RESTARTS   AGE
mongodb-deployment-778f488b57-cg7br   0/1     ImagePullBackOff   0          77s
```

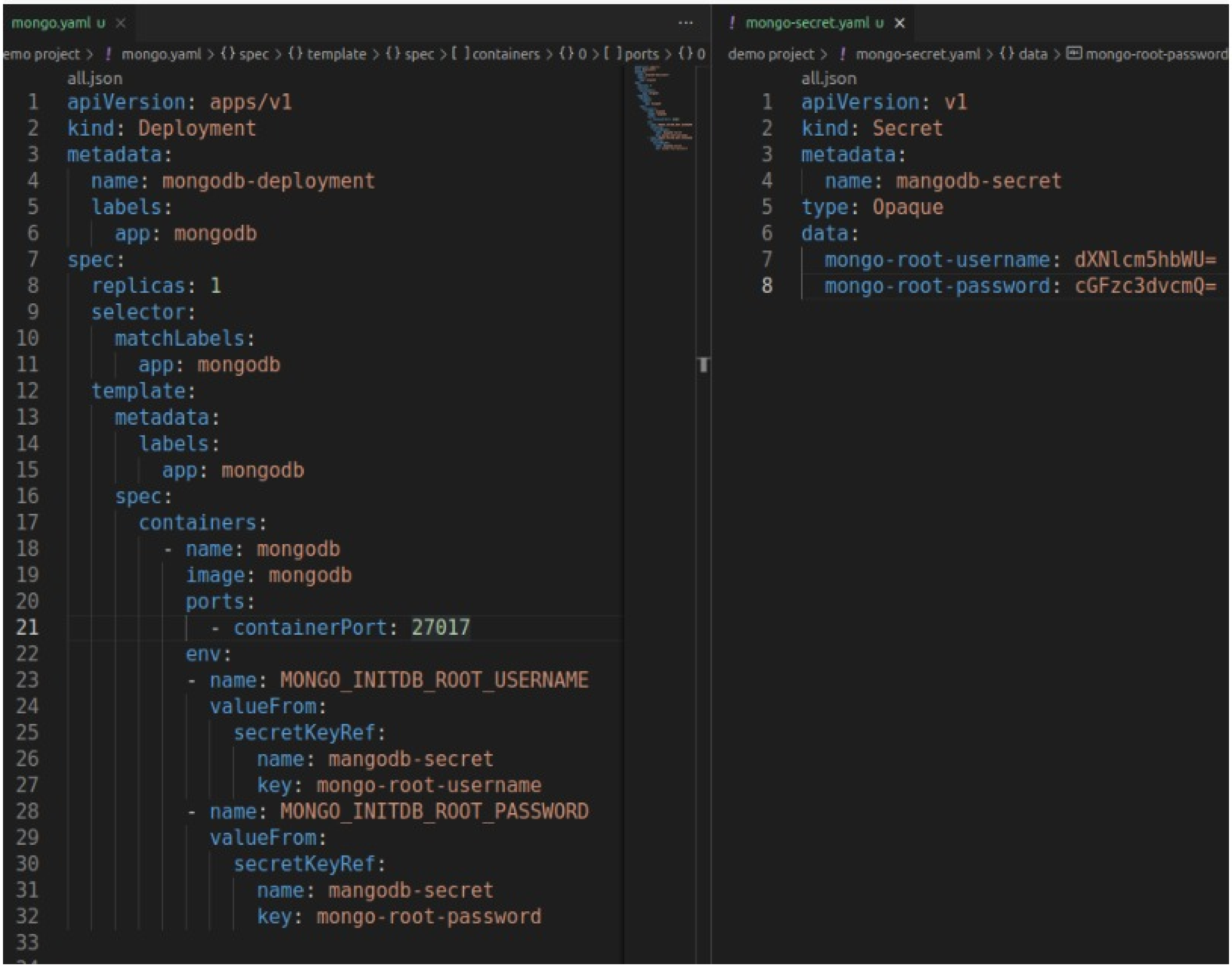


```
      3m46s   172.17.0.3   minikube   <none>           <none>
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ kubectl describe pod mongodb-deployment-658bf778dc-bx9nf
Name:         mongodb-deployment-658bf778dc-bx9nf
Namespace:    default
Priority:     0
Node:         minikube/192.168.49.2
Start Time:   Wed, 13 Jul 2022 14:10:04 +0200
Labels:       app=mongodb
              pod-template-hash=658bf778dc
Annotations:  <none>
```

Create a tunnel service -> communicate with mongodb

in yaml we can ~~put different files together. --- document seperation~~

We need to create service



mongo.yaml U × | mongo-secret.yaml U ×

emo project > mongo.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] ports > {} 0

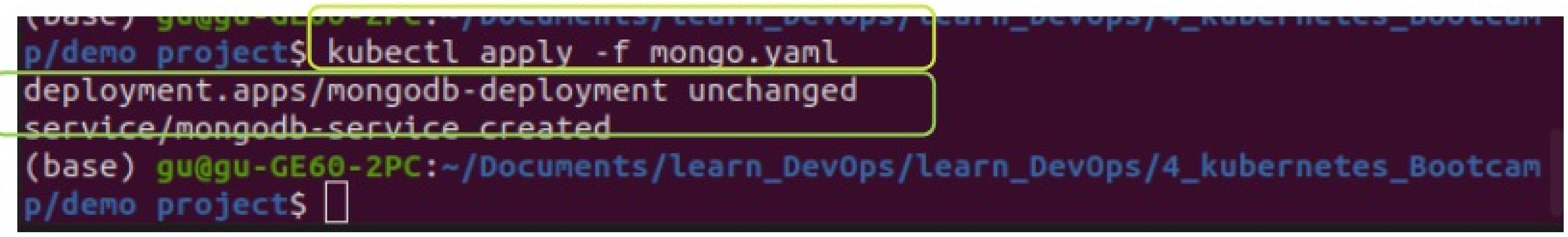
all.json

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb-deployment
  labels:
    app: mongodb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
        - name: mongodb
          image: mongodb
          ports:
            - containerPort: 27017
          env:
          - name: MONGO_INITDB_ROOT_USERNAME
            valueFrom:
              secretKeyRef:
                name: mangodb-secret
                key: mongo-root-username
          - name: MONGO_INITDB_ROOT_PASSWORD
            valueFrom:
              secretKeyRef:
                name: mangodb-secret
                key: mongo-root-password
```

demo project > mongo-secret.yaml > {} data > mongo-root-password

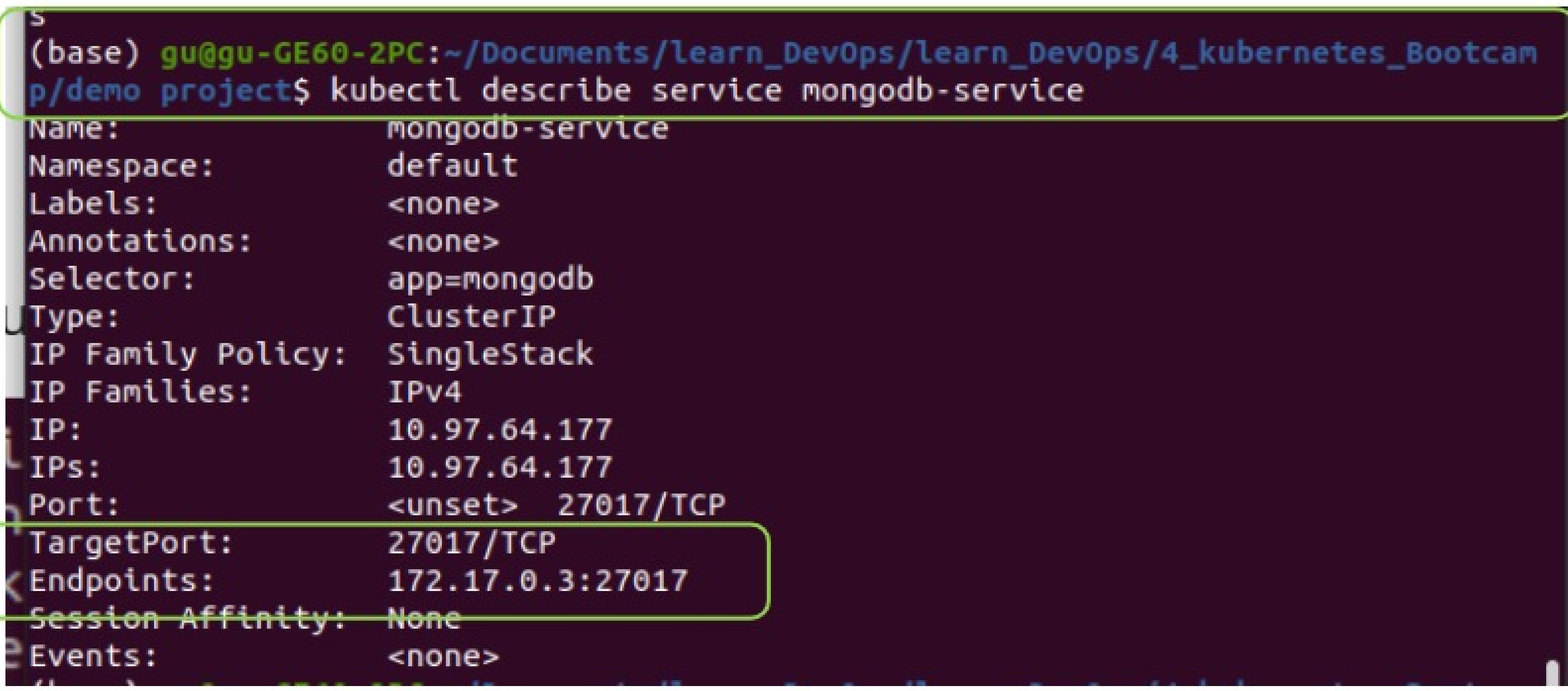
all.json

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mangodb-secret
type: Opaque
data:
  mongo-root-username: dXNlcm5hbWU=
  mongo-root-password: cGFzc3dvcmQ=
```



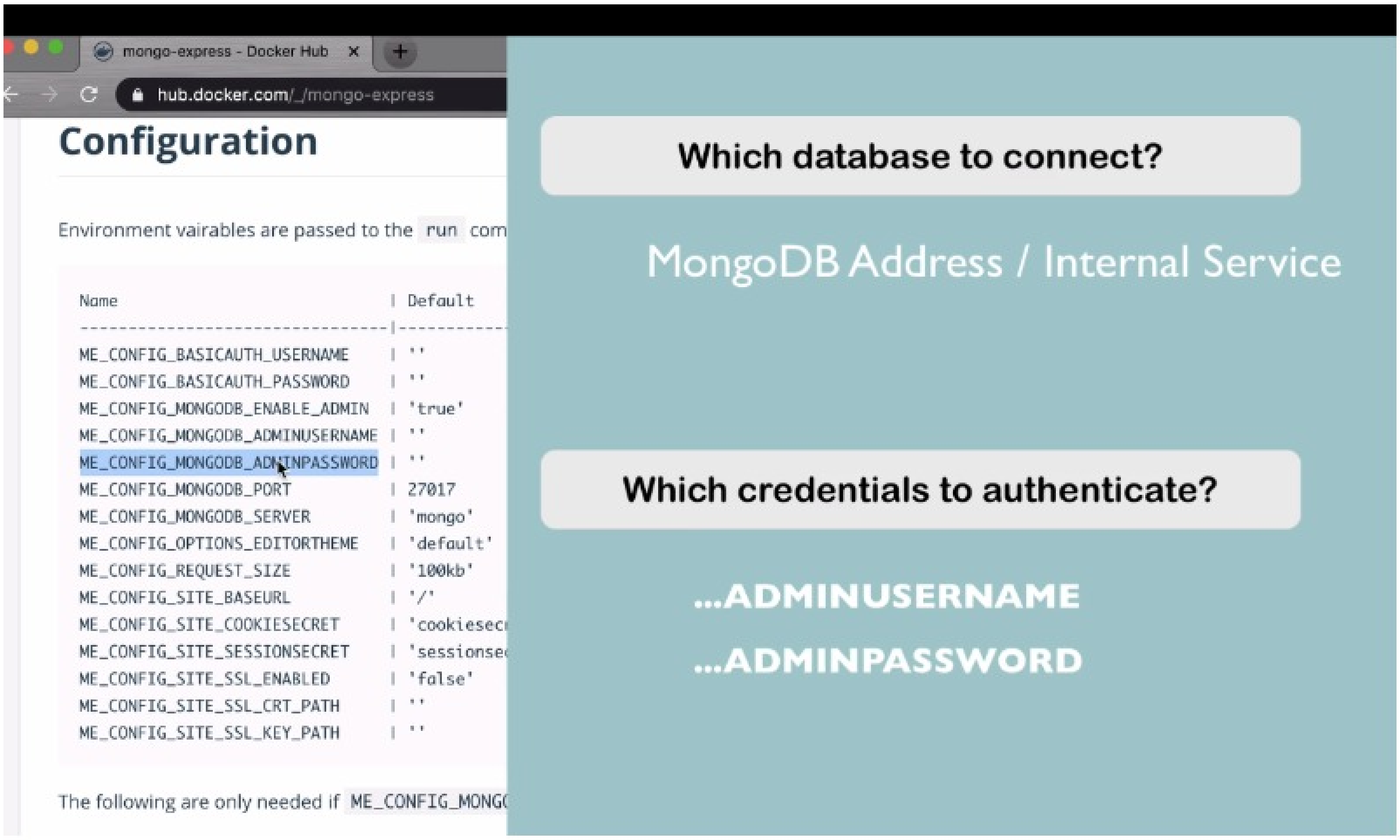
```
p/demo project$ kubectl apply -f mongo.yaml
deployment.apps/mongodb-deployment unchanged
service/mongodb-service created
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ 
```

write the service deployment together with mongodb deployment seperated with ---
Then kubectl apply the file again.



```
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ kubectl describe service mongodb-service
Name:              mongodb-service
Namespace:         default
Labels:            <none>
Annotations:       <none>
Selector:          app=mongodb
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.97.64.177
IPs:               10.97.64.177
Port:              <unset>  27017/TCP
TargetPort:        27017/TCP
Endpoints:         172.17.0.3:27017
Session Affinity:  None
Events:            <none>
```

Next step is to create: Mongo express, Mongo express externel service ConfigMap DB url



## Configuration

Environment vairables are passed to the `run` com

```
Name                            | Default
--------------------------------|-----------
ME_CONFIG_BASICAUTH_USERNAME    | ''
ME_CONFIG_BASICAUTH_PASSWORD    | ''
ME_CONFIG_MONGODB_ENABLE_ADMIN  | 'true'
ME_CONFIG_MONGODB_ADMINUSERNAME | ''
ME_CONFIG_MONGODB_ADMINPASSWORD | ''
ME_CONFIG_MONGODB_PORT          | 27017
ME_CONFIG_MONGODB_SERVER        | 'mongo'
ME_CONFIG_OPTIONS_EDITORTHEME   | 'default'
ME_CONFIG_REQUEST_SIZE          | '100kb'
ME_CONFIG_SITE_BASEURL          | '/'
ME_CONFIG_SITE_COOKIESECRET     | 'cookiesec
ME_CONFIG_SITE_SESSIONSECRET    | 'sessionse
ME_CONFIG_SITE_SSL_ENABLED      | 'false'
ME_CONFIG_SITE_SSL_CRT_PATH     | ''
ME_CONFIG_SITE_SSL_KEY_PATH     | ''
```

The following are only needed if `ME_CONFIG_MONGO`

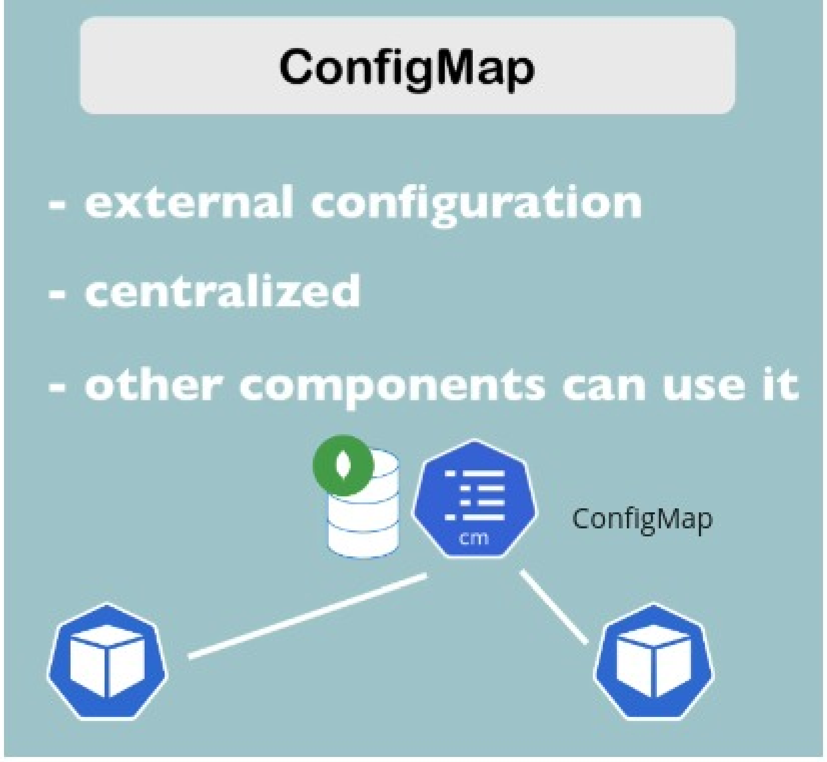
**Which database to connect?**

MongoDB Address / Internal Service

**Which credentials to authenticate?**

**...ADMINUSERNAME**

**...ADMINPASSWORD**



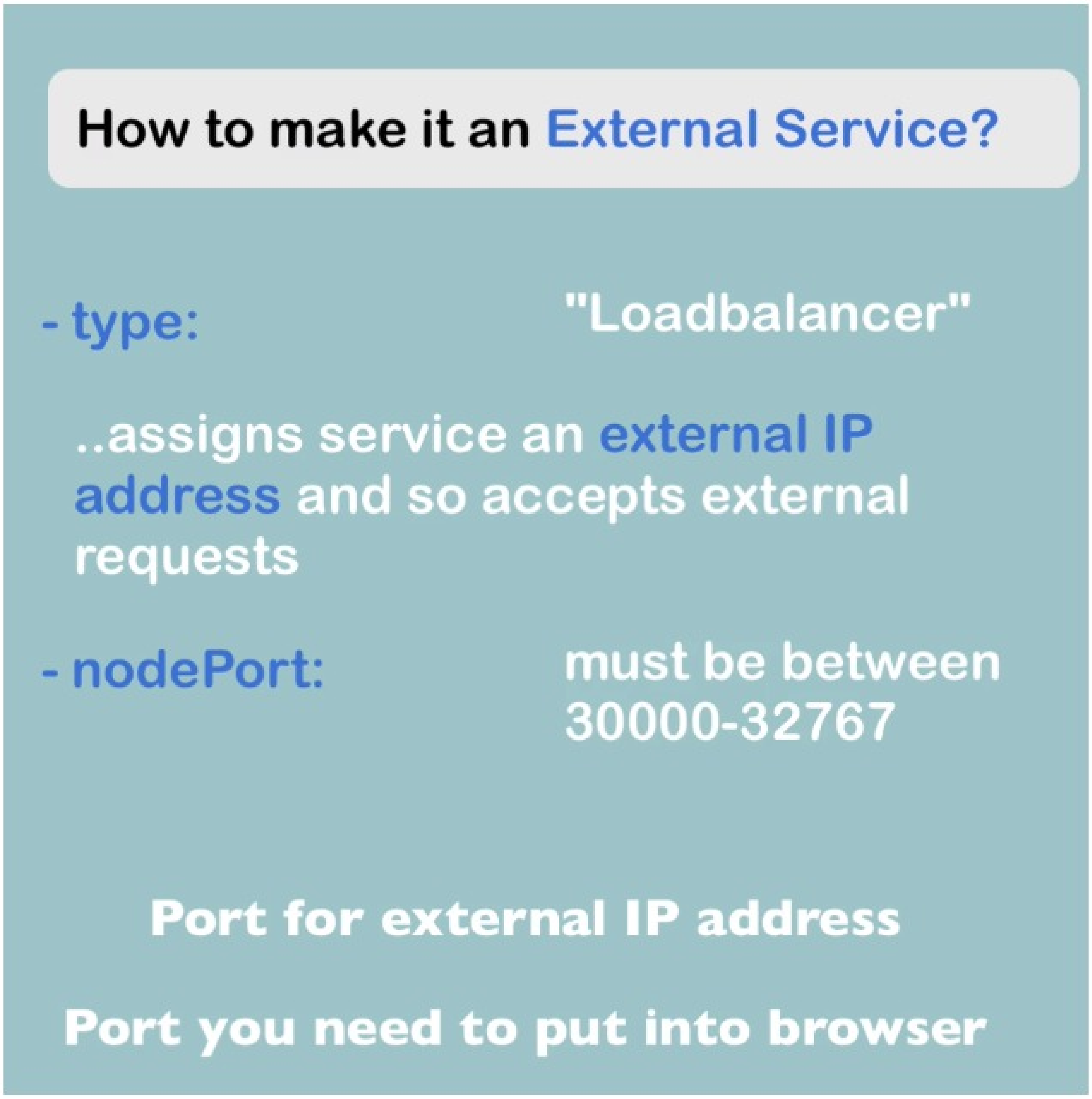
info of database store in the ConfigMap. Other application can also use it

20:05

ConfigMap must already be in the k8s cluster when referencing it!

ConfigMap -> mongo-express ( while mongo-express needs the ME_CONFIG_MONGODB_SERVER , which data is database_url

- deploy ConfigMap
- deploy mongo-express
- check if it works with kubectl logs + podID
- next step is to access mongoDB in the browser => Tunel service



## How to make it an External Service?

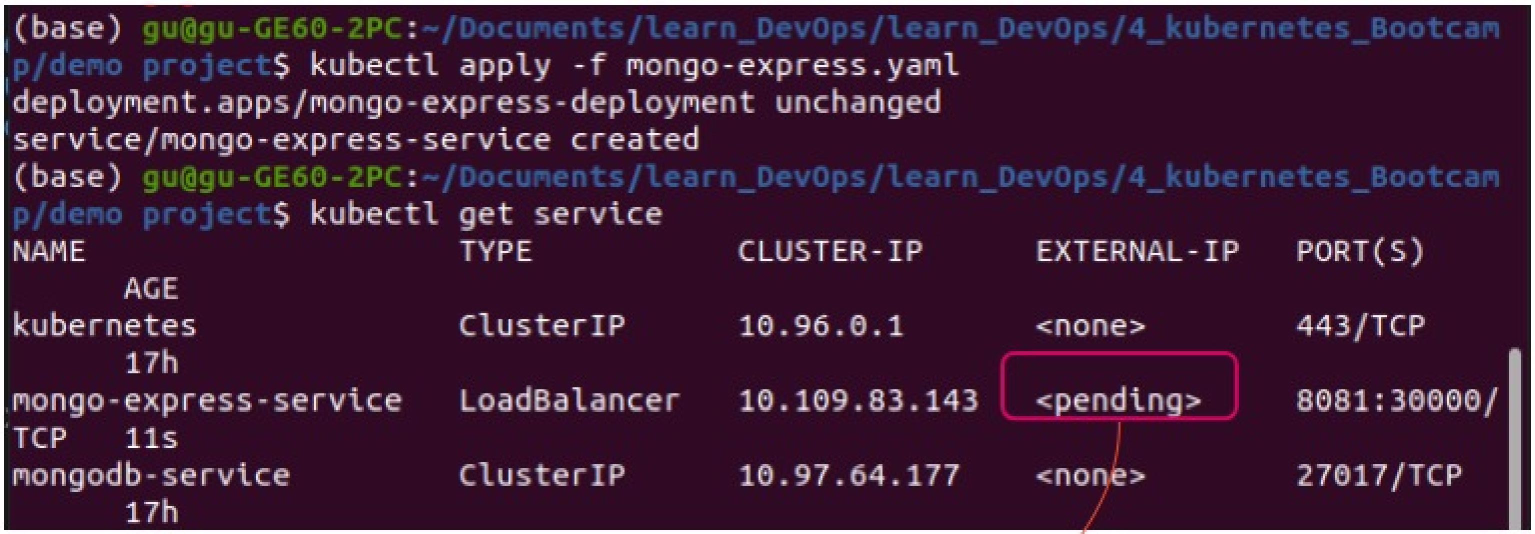
- type: "Loadbalancer"

  ..assigns service an external IP address and so accepts external requests

- nodePort: must be between 30000-32767

**Port for external IP address**

**Port you need to put into browser**

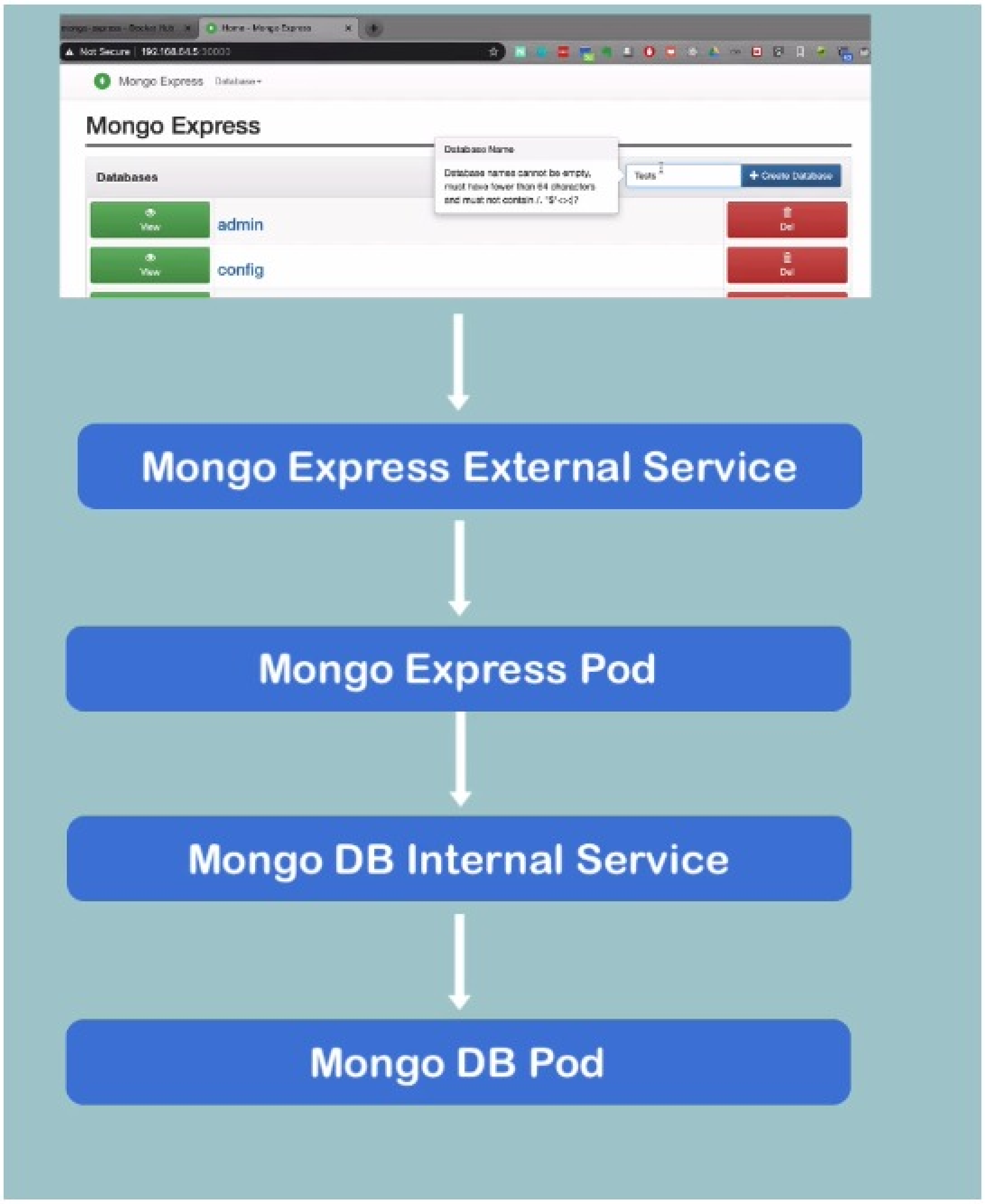


```
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ kubectl apply -f mongo-express.yaml
deployment.apps/mongo-express-deployment unchanged
service/mongo-express-service created
(base) gu@gu-GE60-2PC:~/Documents/learn_DevOps/learn_DevOps/4_kubernetes_Bootcam
p/demo project$ kubectl get service
NAME                    TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)
      AGE
kubernetes              ClusterIP      10.96.0.1       <none>        443/TCP
      17h
mongo-express-service   LoadBalancer   10.109.83.143   <pending>     8081:30000/
TCP    11s
mongodb-service         ClusterIP      10.97.64.177    <none>        27017/TCP
      17h
```

Internal service or cluster IP is default

minikube for loadbanlance pending is external IP
do: minikube service mongo-express-service
You get external service

## Progresses of a request



Database names cannot be empty, must have fewer than 64 characters and must not contain /. "$*<>:|?