

###Please Explain all the instructions and commands you used to complete each Point in MD file

EX:

Q1:

explanation

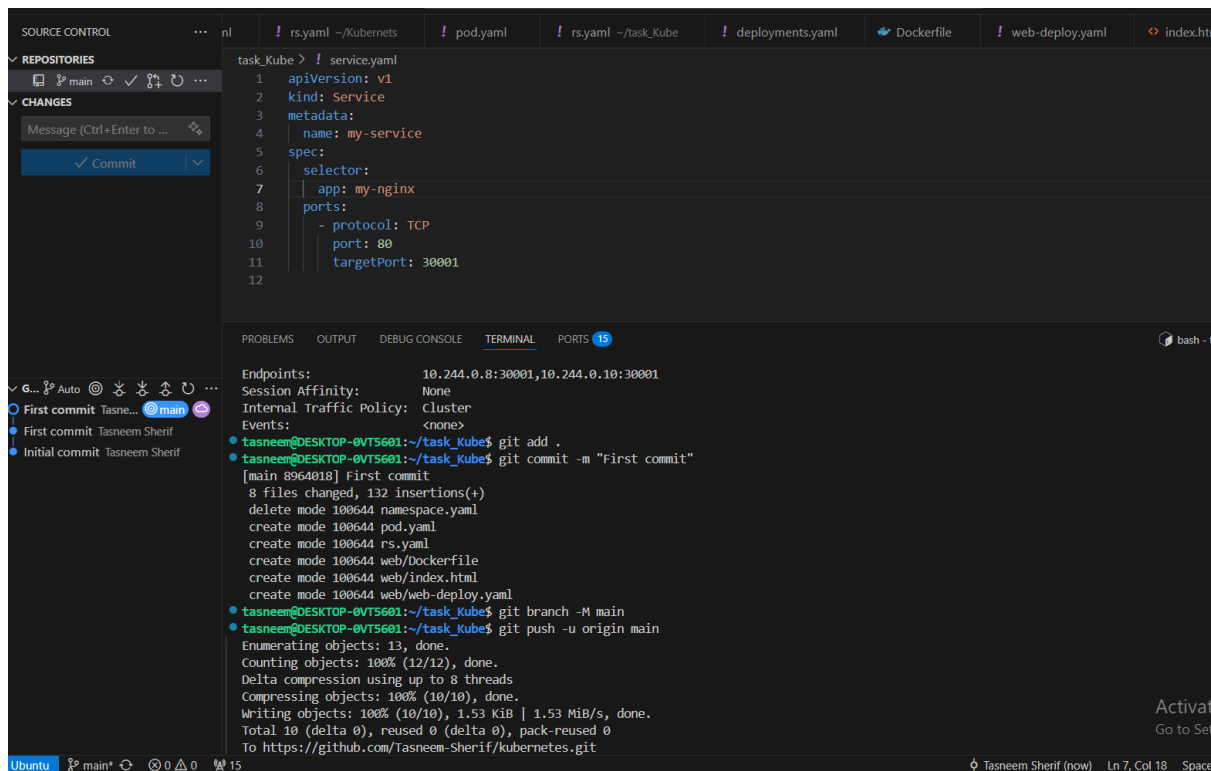
Command used

Output of the command

Screenshot for terminal and browser in case exposing service

- Provide YAML files for all Kubernetes objects created (namespace.yaml, deployments.yaml, service.yaml, etc.). Include the output of relevant kubectl commands that verify the successful completion of each step. Provide screenshots showing the services running and accessible

- Push all of them into your GitHub account.



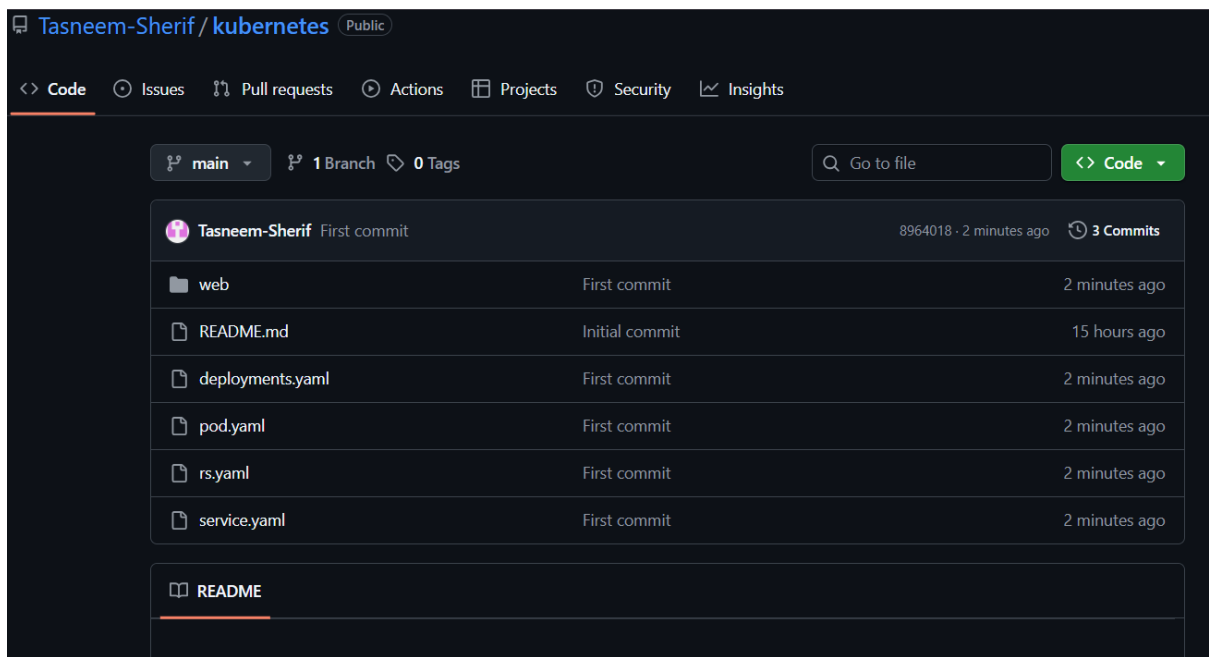
The screenshot shows a VS Code editor with a dark theme. The top bar displays several open files: `rs.yaml ~/Kubernetes`, `pod.yaml`, `rs.yaml ~/task_Kube`, `deployments.yaml`, `Dockerfile`, `web-deploy.yaml`, and `index.ht`. The left sidebar shows the 'REPOSITORIES' and 'CHANGES' panels. The 'CHANGES' panel shows a commit message 'First commit' and a 'Commit' button. The main editor area shows the `task_Kube > ! service.yaml` file with the following content:

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: my-service
5 spec:
6   selector:
7     app: my-nginx
8   ports:
9     - protocol: TCP
10     port: 80
11     targetPort: 30001
12
```

Below the editor, the 'TERMINAL' panel is active, showing the output of several git commands:

```
Endpoints: 10.244.0.8:30001,10.244.0.10:30001
Session Affinity: None
Internal Traffic Policy: Cluster
Events: <none>
tasneem@DESKTOP-0VT5601:~/task_Kube$ git add .
tasneem@DESKTOP-0VT5601:~/task_Kube$ git commit -m "First commit"
[main 8964010] First commit
8 files changed, 132 insertions(+)
delete mode 100644 namespace.yaml
create mode 100644 pod.yaml
create mode 100644 rs.yaml
create mode 100644 web/Dockerfile
create mode 100644 web/index.html
create mode 100644 web/web-deploy.yaml
tasneem@DESKTOP-0VT5601:~/task_Kube$ git branch -M main
tasneem@DESKTOP-0VT5601:~/task_Kube$ git push -u origin main
Enumerating objects: 13, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.53 KiB | 1.53 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Tasneem-Sherif/kubernetes.git
```

The bottom status bar shows 'Ubuntu', 'main*', '0 0 0', and '15'.



=====

Part1:

1- create pod nginx with name my nginx direct from command don't use yaml file

kubectl run my-nginx --image=nginx

kubectl get pods

```
Switched to context "minikube".
tasneem@DESKTOP-0VT5601:~$ kubectl run my-nginx --image=nginx
pod/my-nginx created
tasneem@DESKTOP-0VT5601:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-nginx      0/1     ContainerCreating   0          19s
tasneem@DESKTOP-0VT5601:~$ kubectl describe pod my-nginx
Name:          my-nginx
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Mon, 01 Sep 2025 03:12:42 +0300
Labels:        run=my-nginx
Annotations:    <none>
Status:        Pending
IP:            <none>
IPs:           <none>
Containers:
  my-nginx:
    Container ID:
    Image:        nginx
    Image ID:
    Port:         <none>
    Host Port:    <none>
    State:        Waiting
      Reason:     ContainerCreating
    Ready:        False
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-nrp9f (ro)
Conditions:
  State:        Waiting
  Reason:       ContainerCreating
  Ready:        False
  Restart Count: 0
  Environment:  <none>
  Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-nrp9f (ro)
Conditions:
  Type              Status
  PodReadyToStartContainers  False
  Initialized        True
  Ready              False
  ContainersReady    False
  PodScheduled       True
Volumes:
  kube-api-access-nrp9f:
    Type:          Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:  kube-root-ca.crt
    Optional:      false
    DownwardAPI:   true
QoS Class:        BestEffort
Node-Selectors:    <none>
Tolerations:       node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                   node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From          Message
  ----     -
  Normal   Scheduled   44s   default-scheduler   Successfully assigned default/my-nginx to minikube
  Normal   Pulling     42s   kubelet          Pulling image "nginx"
tasneem@DESKTOP-0VT5601:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-nginx      1/1     Running   0          3m1s
tasneem@DESKTOP-0VT5601:~$
```

2- create pod nginx with name my nginx command and use Image nginx123 direct from command don't use yaml file

kubectl run my-nginx --image=nginx123

kubectl get pods

kubectl describe pod my-nginx

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 14
tasneem@DESKTOP-0VT5601:~$ kubectl run my-nginx --image=nginx123
Error from server (AlreadyExists): pods "my-nginx" already exists
tasneem@DESKTOP-0VT5601:~$ kubectl delete pod my-nginx
pod "my-nginx" deleted
tasneem@DESKTOP-0VT5601:~$ kubectl run my-nginx --image=nginx123
pod/my-nginx created
tasneem@DESKTOP-0VT5601:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
my-nginx 0/1 ErrImagePull 0 7s
tasneem@DESKTOP-0VT5601:~$ kubectl describe pod my-nginx
Name: my-nginx
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.49.2
Start Time: Mon, 01 Sep 2025 03:16:40 +0300
Labels: run-my-nginx
Annotations: <none>
Status: Pending
IP: 10.244.0.7
IPs:
IP: 10.244.0.7
Containers:
  my-nginx:
    Container ID:
    Image: nginx123
    Image ID:
    Port: <none>
    Host Port: <none>
    State: Waiting
    Reason: ErrImagePull
    Ready: False
    Restart Count: 0
    Environment: <none>
```

3- check the status and why it doesn't work

kubectl describe pod my-nginx

```
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-tfqkij (ro)
Conditions:
  Type           Status
  PodReadyToStartContainers True
  Initialized     True
  Ready           False
  ContainersReady False
  PodScheduled    True
Volumes:
  kube-api-access-tfqkij:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    Optional: false
    DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From          Message
  ----     -
  Normal   Scheduled   13s   default-scheduler Successfully assigned default/my-nginx to minikube
  Normal   Pulling     12s   kubelet       Pulling image "nginx123"
  Warning   Failed      9s    kubelet       Failed to pull image "nginx123": Error response from daemon: pull access denied for nginx123, repository does not exist or
may require 'docker login': denied: requested access to the resource is denied
  Warning   Failed      9s    kubelet       Error: ErrImagePull
  Normal   BackOff     9s    kubelet       Back-off pulling image "nginx123"
  Warning   Failed      9s    kubelet       Error: ImagePullBackOff
```

4- I need to know node name - IP - Image Of the POD

kubectl get pods -o wide

```
tasneem@DESKTOP-0VT5601:~$ kubectl get pods -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
my-nginx 0/1 ImagePullBackOff 0 12m 10.244.0.7 minikube <none> <none>
```

5- delete the pod

kubectl delete pod my-nginx

```
tasneem@DESKTOP-0VT5601:~$ kubectl get pods -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
my-nginx 0/1 ImagePullBackOff 0 12m 10.244.0.7 minikube <none> <none>
tasneem@DESKTOP-0VT5601:~$ kubectl delete pod my-nginx
pod "my-nginx" deleted
tasneem@DESKTOP-0VT5601:~$
```

6- create another one with yaml file and use label

kubectl apply -f pod.yaml

kubectl get pods

The screenshot shows a VS Code editor with a file named `pod.yaml` open. The file contains a Kubernetes Pod definition. Below the editor, the terminal shows the execution of `kubectl apply -f pod.yaml` and `kubectl get pods`, resulting in a single pod named `my-nginx` in a `Running` state.

```
task_Kube > ! pod.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: my-nginx
5    labels:
6      app: my-nginx
7  spec:
8    containers:
9      - name: nginx-container
10        image: nginx:latest
11        ports:
12          - containerPort: 80

task_Kube > ! rs.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: my-nginx-rs
5    labels:
6      app: my-nginx
7  spec:
8    replicas: 3
9    selector:
10      matchLabels:
11        app: my-nginx
12  template:
13    metadata:
14      labels:
15        app: my-nginx
16    spec:
17      containers:
18        - name: nginx-container
19          image: nginx:latest
20          ports:
21            - containerPort: 80

task_Kube > ! deployments.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: my-nginx-deployment
5    labels:
6      app: my-nginx
7  spec:
8    replicas: 3
9    selector:
10      matchLabels:
11        app: my-nginx
12  template:
13    metadata:
14      labels:
15        app: my-nginx
16    spec:
17      containers:
18        - name: nginx-container
19          image: nginx:latest
20          ports:
21            - containerPort: 80

task_Kube > ! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: my-nginx-service
5    labels:
6      app: my-nginx
7  spec:
8    selector:
9      app: my-nginx
10    ports:
11      - port: 80
12        targetPort: 80
```

```
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl apply -f pod.yaml
pod/my-nginx created
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
my-nginx    1/1     Running   0           14s
tasneem@DESKTOP-0VT5601:~/task_Kube$
```

7-create Riplicaset with 3 replicas using nginx Image

kubectl apply -f rs.yaml

kubectl get rs

kubectl get pods

The screenshot shows a VS Code editor with a file named `rs.yaml` open. The file contains a Kubernetes ReplicaSet definition. Below the editor, the terminal shows the execution of `kubectl apply -f rs.yaml`, `kubectl get rs`, and `kubectl get pods`. The output shows a ReplicaSet named `my-nginx-rs` with 3 replicas, and three pods in a `Running` state.

```
task_Kube > ! rs.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: my-nginx-rs
5    labels:
6      app: my-nginx
7  spec:
8    replicas: 3
9    selector:
10      matchLabels:
11        app: my-nginx
12  template:
13    metadata:
14      labels:
15        app: my-nginx
16    spec:
17      containers:
18        - name: nginx-container
19          image: nginx:latest
20          ports:
21            - containerPort: 80

task_Kube > ! deployments.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: my-nginx-deployment
5    labels:
6      app: my-nginx
7  spec:
8    replicas: 3
9    selector:
10      matchLabels:
11        app: my-nginx
12  template:
13    metadata:
14      labels:
15        app: my-nginx
16    spec:
17      containers:
18        - name: nginx-container
19          image: nginx:latest
20          ports:
21            - containerPort: 80

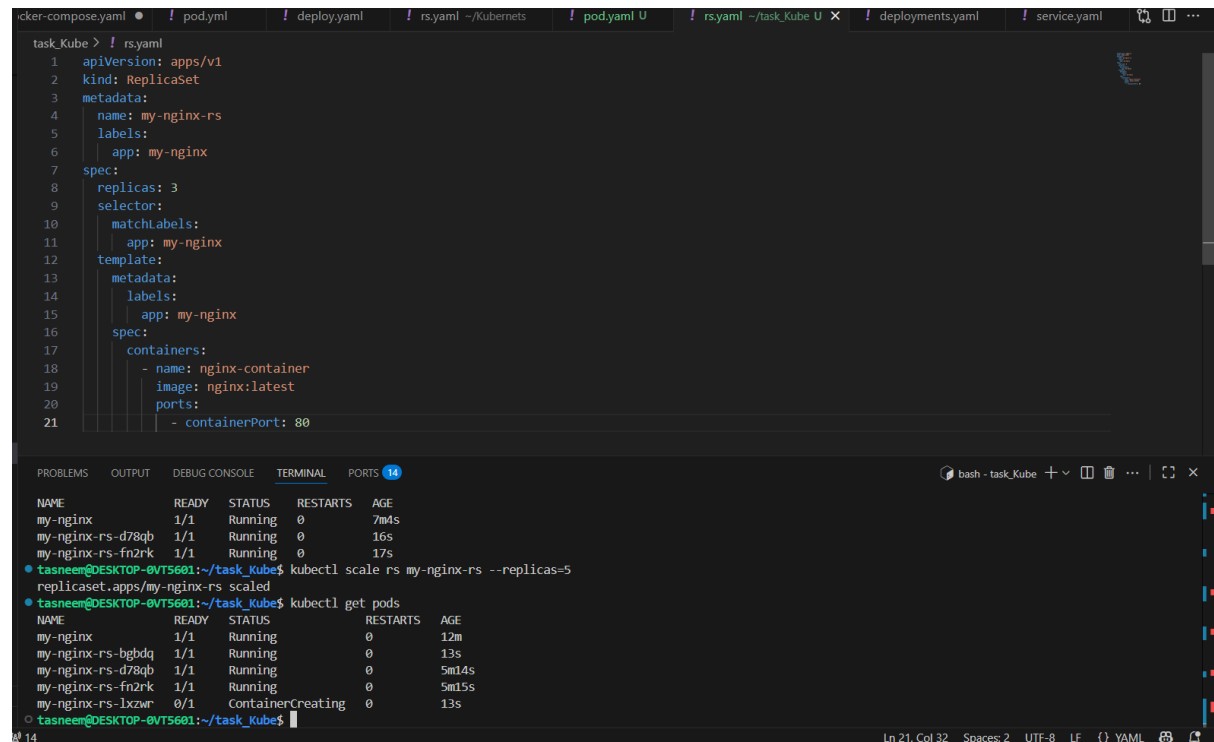
task_Kube > ! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: my-nginx-service
5    labels:
6      app: my-nginx
7  spec:
8    selector:
9      app: my-nginx
10    ports:
11      - port: 80
12        targetPort: 80
```

```
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl apply -f rs.yaml
replicaset.apps/my-nginx-rs created
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get rs
NAME          DESIRED   CURRENT   READY   AGE
my-nginx-rs   3         3         3       11s
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
my-nginx            1/1     Running   0           7m4s
my-nginx-rs-d78qb   1/1     Running   0           16s
my-nginx-rs-fn2rk   1/1     Running   0           17s
tasneem@DESKTOP-0VT5601:~/task_Kube$
```

8-scale the replicas to 5 without edit in the Yaml file

kubectl scale rs my-nginx-rs --replicas=5

kubectl get pods



```
task_Kube > ! rs.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: my-nginx-rs
5    labels:
6      app: my-nginx
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: my-nginx
12   template:
13     metadata:
14       labels:
15         app: my-nginx
16     spec:
17       containers:
18       - name: nginx-container
19         image: nginx:latest
20         ports:
21         - containerPort: 80
```

```
bash - task_Kube
NAME          READY   STATUS    RESTARTS   AGE
my-nginx       1/1     Running   0           7m4s
my-nginx-rs-d78qb 1/1     Running   0           16s
my-nginx-rs-fn2rk 1/1     Running   0           17s
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl scale rs my-nginx-rs --replicas=5
replicaset.apps/my-nginx-rs scaled
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-nginx       1/1     Running   0           12m
my-nginx-rs-bgbdq 1/1     Running   0           13s
my-nginx-rs-d78qb 1/1     Running   0           5m14s
my-nginx-rs-fn2rk 1/1     Running   0           5m15s
my-nginx-rs-lxzw 0/1     ContainerCreating   0           13s
tasneem@DESKTOP-0VT5601:~/task_Kube$
```

9-Delete any one of the 5 pods and check what happen and explain

If you delete a Pod that was created by a ReplicaSet or a Deployment, Kubernetes will immediately create a new Pod to replace it.

This happens because the Deployment (or ReplicaSet) always tries to maintain the desired number of replicas.

kubectl delete pod my-nginx-rs-bgbdq

kubectl get pods

The screenshot shows a VS Code editor with a file named `rs.yaml` open. The file contains a Kubernetes `ReplicaSet` configuration for an application named `my-nginx`. The configuration specifies 3 replicas and uses the `nginx:latest` image. The terminal output shows the `kubectl` commands being executed to delete a pod and then list all pods.

```
task_Kube > ! rs.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: my-nginx-rs
5    labels:
6      app: my-nginx
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: my-nginx
12   template:
13     metadata:
14       labels:
15         app: my-nginx
16     spec:
17       containers:
18       - name: nginx-container
19         image: nginx:latest
20         ports:
21         - containerPort: 80
```

Below the editor, the terminal output is displayed:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 14
my-nginx-rs-bgbdq 1/1 Running 0 13s
my-nginx-rs-d78qb 1/1 Running 0 5m14s
my-nginx-rs-fn2rk 1/1 Running 0 5m15s
my-nginx-rs-lxzw 0/1 ContainerCreating 0 13s
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl delete pod my-nginx-rs-bgbdq
pod "my-nginx-rs-bgbdq" deleted
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get pods
NAME READY STATUS RESTARTS AGE
my-nginx 1/1 Running 0 14m
my-nginx-rs-67gs2 1/1 Running 0 12s
my-nginx-rs-d78qb 1/1 Running 0 7m23s
my-nginx-rs-fn2rk 1/1 Running 0 7m24s
my-nginx-rs-lxzw 1/1 Running 0 2m22s
tasneem@DESKTOP-0VT5601:~/task_Kube$
```

10-Scale down the pods aging to 2 without scale command use terminal

in Yaml file:

spec:

replicas: 2

`kubectl apply -f rs.yaml`

`kubectl get pods`

```
task_Kube > ! rs.yaml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: my-nginx-rs
5    labels:
6      app: my-nginx
7  spec:
8    replicas: 2
9    selector:
10     matchLabels:
11       app: my-nginx
12   template:
13     metadata:
14       labels:
15         app: my-nginx
16     spec:
17       containers:
18         - name: nginx-container
19           image: nginx:latest
20           ports:
21             - containerPort: 80
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 14

my-nginx-rs-67gs2 1/1 Running 0 12s
my-nginx-rs-d78qb 1/1 Running 0 7m23s
my-nginx-rs-fn2rk 1/1 Running 0 7m24s
my-nginx-rs-lxzwv 1/1 Running 0 2m22s

tasneem@DESKTOP-0VT5601:~/task_Kube\$ kubectl apply -f rs.yaml
replicaset.apps/my-nginx-rs configured

tasneem@DESKTOP-0VT5601:~/task_Kube\$ kubectl get rs

NAME	DESIRED	CURRENT	READY	AGE
my-nginx-rs	2	2	2	17m

tasneem@DESKTOP-0VT5601:~/task_Kube\$ kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
my-nginx	1/1	Running	0	24m
my-nginx-rs-d78qb	1/1	Running	0	17m

tasneem@DESKTOP-0VT5601:~/task_Kube\$

11- find out the issue in the below Yaml (don't use AI)

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: replicaset-2

spec:

replicas: 2

selector:

matchLabels:

tier: frontend

template:

metadata:

labels:

tier: nginx

spec:

containers:

- name: nginx

image: nginx

The issue is that the selector labels do not match the template labels. A ReplicaSet only manages Pods that match its selector. Since tier: frontend don't match tier: nginx, the ReplicaSet won't manage the created Pods. To fix it, make both labels the same

Sol:

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: replicaset-2

spec:

replicas: 2

selector:

matchLabels:

tier: nginx

template:

metadata:

labels:

tier: nginx

spec:

containers:

- name: nginx

image: nginx

12- find out the issue in the below Yaml (don't use AI)

apiVersion: apps/v1

```
kind: deployment
metadata:
  name: deployment-1
spec:
  replicas: 2
  selector:
    matchLabels:
      name: busybox-pod
  template:
    metadata:
      labels:
        name: busybox-pod
    spec:
      containers:
        - name: busybox-container
          image: busybox
          command:
            - sh
            - "-c"
            - echo Hello Kubernetes! && sleep 3600
```

The issue is that kind is written as deployment with a small 'd'. Kubernetes is case-sensitive, so it should be Deployment

```
sol:
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-1
spec:
```

replicas: 2

selector:

matchLabels:

name: busybox-pod

template:

metadata:

labels:

name: busybox-pod

spec:

containers:

- name: busybox-container

image: busybox

command:

- sh

- "-c"

- echo Hello Kubernetes! && sleep 3600

The screenshot shows a VS Code editor with a file named `deployments.yaml` open. The file contains a Kubernetes Deployment manifest. Below the editor, a terminal window is open, showing the output of several commands. The first command is `kubectl get pods`, which shows a list of pods. The second command is `kubectl apply -f deployments.yaml`, which fails with an error: `error: the path "deployments.yaml" does not exist`. The third command is `kubectl apply -f deployments.yaml`, which fails with an error: `Error from server (BadRequest): error when creating "deployments.yaml": deployment in version "v1" cannot be handled as a Deployment: no kind "deployment" is registered for version "apps/v1" in scheme "pkg/api/legacyscheme/scheme.go:30"`.

```
task_Kube > ! deployments.yaml
1  apiVersion: apps/v1
2
3  kind: deployment
4
5  metadata:
6    name: deployment-1
7
8  spec:
9
10   replicas: 2
11
12   selector:
13     matchLabels:
14       name: busybox-pod
15
16   template:
17     metadata:
18       labels:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 14

NAME DESIRED CURRENT READY AGE
my-nginx-rs 2 2 2 17m

tasneem@DESKTOP-0VT5601:~/task_Kube\$ kubectl get pods

NAME READY STATUS RESTARTS AGE
my-nginx 1/1 Running 0 24m
my-nginx-rs-d78qb 1/1 Running 0 17m

tasneem@DESKTOP-0VT5601:~/task_Kube\$ kubectl apply -f deployments.yaml
error: the path "deployments.yaml" does not exist

tasneem@DESKTOP-0VT5601:~/task_Kube\$ kubectl apply -f deployments.yaml
Error from server (BadRequest): error when creating "deployments.yaml": deployment in version "v1" cannot be handled as a Deployment: no kind "deployment" is registered for version "apps/v1" in scheme "pkg/api/legacyscheme/scheme.go:30"

13- find out the issue in the below Yaml (don't use AI)

apiVersion: v1

kind: Deployment

metadata:

name: deployment-1

spec:

replicas: 2

selector:

matchLabels:

name: busybox-pod

template:

metadata:

labels:

name: busybox-pod

spec:

containers:

- name: busybox-container

image: busybox

command:

- sh

- "-c"

- echo Hello Kubernetes! && sleep 3600

The issue is that the apiVersion is set to v1. Deployments belong to the apps/v1 API group, not v1. Changing it to apps/v1 fixes the error

sol:

apiVersion: apps/v1

kind: Deployment

metadata:

name: deployment-1

spec:

replicas: 2

selector:

matchLabels:

name: busybox-pod

template:

metadata:

labels:

name: busybox-pod

spec:

containers:

- name: busybox-container

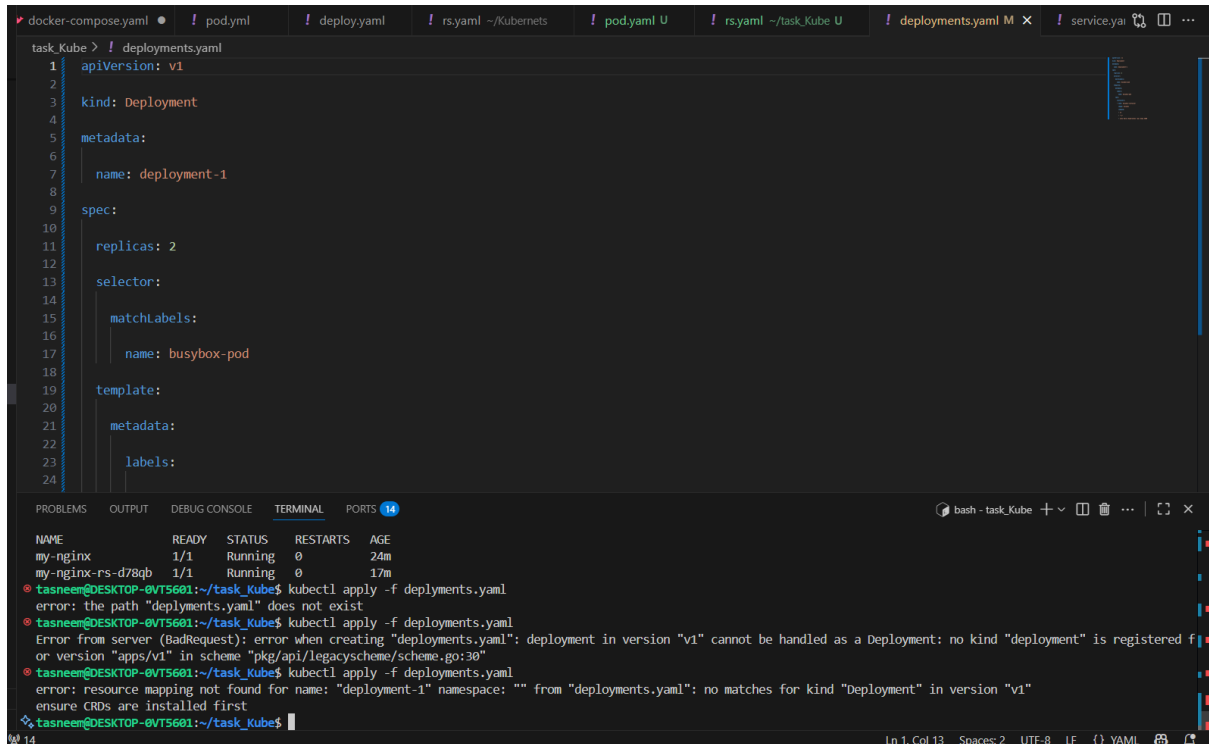
image: busybox

command:

- sh

- "-c"

- echo Hello Kubernetes! && sleep 3600



The screenshot shows a VS Code editor with a file named `deployments.yaml` open. The file content is as follows:

```
1 apiVersion: v1
2
3 kind: Deployment
4
5 metadata:
6   name: deployment-1
7
8 spec:
9
10   replicas: 2
11
12   selector:
13     matchLabels:
14       name: busybox-pod
15
16   template:
17     metadata:
18       labels:
```

The terminal window at the bottom shows the following commands and errors:

```
tasneem@DESKTOP-0VT5601:~/task_kube$ kubectl apply -f deployments.yaml
error: the path "deployments.yaml" does not exist
tasneem@DESKTOP-0VT5601:~/task_kube$ kubectl apply -f deployments.yaml
Error from server (BadRequest): error when creating "deployments.yaml": deployment in version "v1" cannot be handled as a Deployment: no kind "deployment" is registered f
tasneem@DESKTOP-0VT5601:~/task_kube$ kubectl apply -f deployments.yaml
error: resource mapping not found for name: "deployment-1" namespace: "" from "deployments.yaml": no matches for kind "Deployment" in version "v1"
ensure CRDs are installed first
tasneem@DESKTOP-0VT5601:~/task_kube$
```

The terminal also shows a table of running pods:

NAME	READY	STATUS	RESTARTS	AGE
my-nginx	1/1	Running	0	24m
my-nginx-rs-d78qb	1/1	Running	0	17m

14- what's command you use to know what Image name that running the deployment
kubectl get deploy -o wide

The screenshot shows a VS Code editor with a file named `deployments.yaml` open. The file contains a Kubernetes Deployment manifest. The terminal output shows the command `kubectl apply -f deployments.yaml` being executed, which results in an error: `error: resource mapping not found for name: "deployment-1" namespace: "" from "deployments.yaml": no matches for kind "Deployment" in version "v1"`. The error message suggests that the Deployment resource is not available in the current version of the API. The terminal also shows the command `kubectl get deploy` being executed, which returns the following output:

```
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1 0/2     2            0           7s
```

The terminal also shows the command `kubectl get deploy -o wide` being executed, which returns the following output:

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS	IMAGES	SELECTOR
deployment-1	2/2	2	2	29s	busybox-container	busybox	name=busybox-pod

15- create deployment using following data :

Name: httpd-frontend;

Replicas: 3;

Image: httpd:2.4-alpine

in Yaml file:

apiVersion: apps/v1

kind: Deployment

metadata:

name: httpd-frontend

spec:

replicas: 3

selector:

matchLabels:

name: httpd-frontend

template:

metadata:

labels:

name: httpd-frontend

spec:

containers:

- name: httpd-frontend

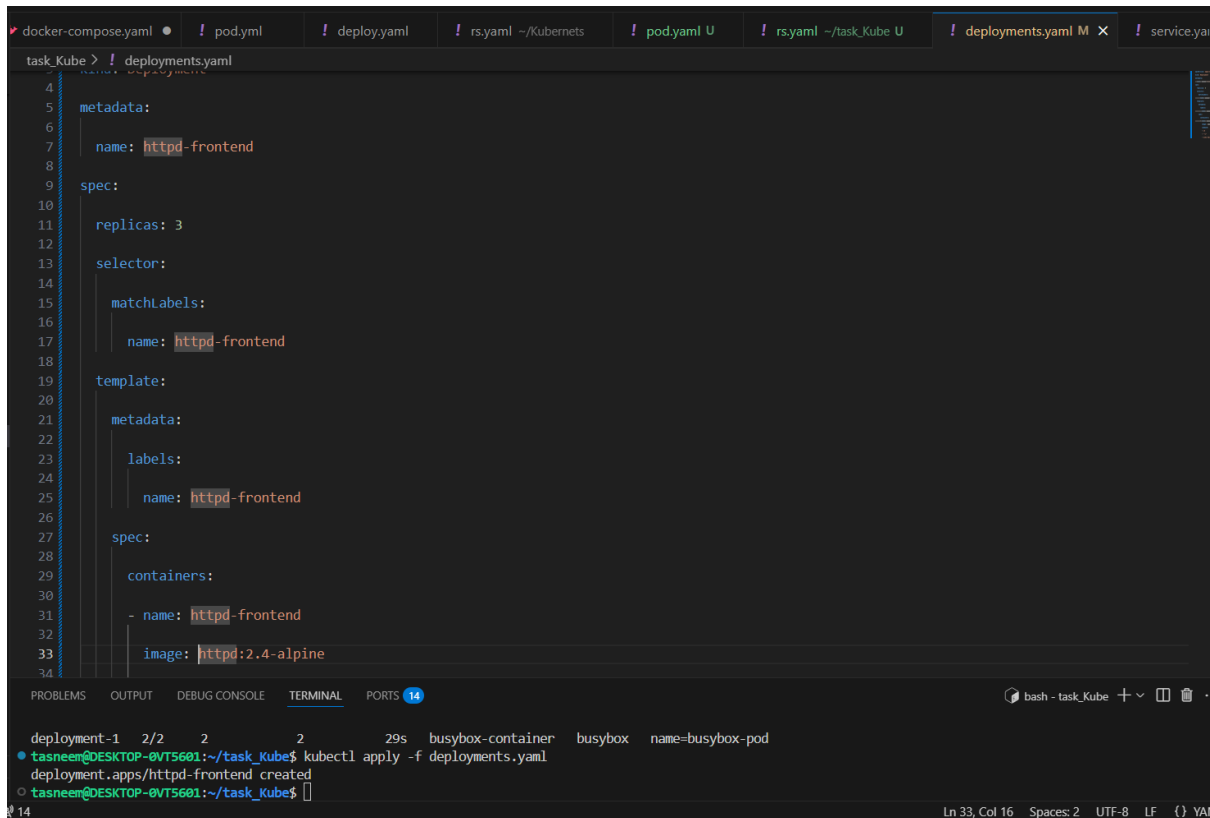
image: httpd:2.4-alpine

command:

- sh

- "-C"

- echo Hello Kubernetes! && sleep 3600



The screenshot shows a VS Code editor with a file explorer at the top displaying several files: `docker-compose.yaml`, `pod.yaml`, `deploy.yaml`, `rs.yaml`, `pod.yaml U`, `rs.yaml ~/task_Kube U`, `deployments.yaml M X`, and `service.yaml`. The main editor window displays the `deployments.yaml` file with the following content:

```
4
5 metadata:
6   name: httpd-frontend
7
8
9 spec:
10   replicas: 3
11
12   selector:
13     matchLabels:
14       name: httpd-frontend
15
16   template:
17     metadata:
18       labels:
19         name: httpd-frontend
20
21     spec:
22       containers:
23       - name: httpd-frontend
24         image: httpd:2.4-alpine
```

Below the editor is a terminal window with the following output:

```
deployment-1 2/2 2 2 29s busybox-container busybox name=busybox-pod
tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl apply -f deployments.yaml
deployment.apps/httpd-frontend created
tasneem@DESKTOP-0VT5601:~/task_Kube$
```

16- replace the image to nginx777 with command directly

`kubectl set image deploy httpd-frontend httpd-frontend=nginx777`

`kubectl describe pod httpd-frontend`

```

tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get deploy -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES           SELECTOR
deployment-1  2/2     2             2           9m31s  busybox-container  busybox          name=busybox-pod
httpd-frontend 3/3     3             3           2m37s  httpd-frontend  httpd:2.4-alpine name=httpd-frontend

tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
deployment-1-858c547685-2p5vx  1/1     Running   0           9m46s
deployment-1-858c547685-ckgxz  1/1     Running   0           9m46s
httpd-frontend-65c89bbdc5-5r6jb 1/1     Running   0           2m53s
httpd-frontend-65c89bbdc5-k4sxx 1/1     Running   0           2m53s
httpd-frontend-65c89bbdc5-ms85p 1/1     Running   0           2m53s
my-nginx      1/1     Running   0           63m
my-nginx-rs-d78qb 1/1     Running   0           56m

tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl set image deploy httpd-frontend httpd-frontend-nginx777
deployment.apps/httpd-frontend image updated

tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get deploy -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES           SELECTOR
deployment-1  2/2     2             2           14m   busybox-container  busybox          name=busybox-pod
httpd-frontend 3/3     1             3           7m16s  httpd-frontend  nginx777         name=httpd-frontend

tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
deployment-1-858c547685-2p5vx  1/1     Running   0           14m
deployment-1-858c547685-ckgxz  1/1     Running   0           14m
httpd-frontend-65c89bbdc5-5r6jb 1/1     Running   0           7m27s
httpd-frontend-65c89bbdc5-k4sxx 1/1     Running   0           7m27s
httpd-frontend-65c89bbdc5-ms85p 1/1     Running   0           7m27s
httpd-frontend-759779c458-6lr2s 0/1     ImagePullBackOff 0           21s
my-nginx      1/1     Running   0           68m
my-nginx-rs-d78qb 1/1     Running   0           61m

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 14
Conditions:
  Type      Status
  PodReadyToStartContainers True
  Initialized True
  Ready      False
  ContainersReady False
  PodScheduled True
Volumes:
  kube-api-access-64cxs:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    Optional:   false
    DownwardAPI: true
  QoS Class:   BestEffort
  Node-Selectors: <none>
  Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
               node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type      Reason      Age      From      Message
  ----      -
  Normal    Scheduled   7m11s    default-scheduler    Successfully assigned default/httpd-frontend-759779c458-6lr2s to minikube
  Normal    Pulling     3m49s    kubelet    Pulling image "nginx777"
  Warning   Failed     3m44s    kubelet    Failed to pull image "nginx777": Error response from daemon: pull access denied for nginx777, repository does not exist or may require 'docker login': denied: requested access to the resource is denied
  Warning   Failed     3m44s    kubelet    Error: ErrImagePull
  Warning   Failed     183s    kubelet    Error: ImagePullBackOff
  Normal    BackOff     90s     kubelet    Back-off pulling image "nginx777"

```

17- rollback to pervious version

kubectl rollout undo deploy httpd-frontend

```

tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl rollout undo deploy httpd-frontend
deployment.apps/httpd-frontend rolled back

tasneem@DESKTOP-0VT5601:~/task_Kube$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
deployment-1-858c547685-2p5vx  1/1     Running   0           27m
deployment-1-858c547685-ckgxz  1/1     Running   0           27m
httpd-frontend-65c89bbdc5-5r6jb 1/1     Running   0           20m
httpd-frontend-65c89bbdc5-k4sxx 1/1     Running   0           20m
httpd-frontend-65c89bbdc5-ms85p 1/1     Running   0           20m
my-nginx      1/1     Running   0           81m
my-nginx-rs-d78qb 1/1     Running   0           74m

```

18- Create a Simple Web Application:

* Use a Dockerfile to create a simple web application (e.g., an Nginx server serving an HTML page).

* Build the Docker image and push it to DockerHub your private Account.

docker build -t docker_imageweb .

docker run -d -p 8080:80 docker_imageweb

docker login

```
docker push tasneemsherif/docker_imageweb:latest
```

```
task_kube > web > Dockerfile
1 FROM nginx
2 COPY index.html /usr/share/nginx/html/index.html
3 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 15
tasneem@DESKTOP-0VT5601:~/task_kube/web$ docker build -t Docker_imageweb .
[+] Building 0.0s (0/0)
ERROR: failed to build: invalid tag "Docker_imageweb": repository name must be lowercase
tasneem@DESKTOP-0VT5601:~/task_kube/web$ docker build -t docker_imageweb .
[+] Building 18.8s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 98B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 31B
=> CACHED [1/2] FROM docker.io/library/nginx:latest
=> [2/2] COPY index.html /usr/share/nginx/html/index.html
=> exporting to image
=> => exporting layers
=> => writing image sha256:3f1744f9efca605ec3a3032b867e2f5d3d437272ac7a826c5be4c0020a1cecd2
=> naming to docker.io/library/docker_imageweb
tasneem@DESKTOP-0VT5601:~/task_kube/web$ docker run -d -p 8080:80 docker_imageweb
0549b59dc6222db0331613efeb4e54690c73bbf4fd178efb36cb19474ae8e8c
tasneem@DESKTOP-0VT5601:~/task_kube/web$
```



Hello Kubernetes & Docker!

Created by Tasneem ٠٧٣٤

```
tasneem@DESKTOP-0VT5601:~/task_Kube/web$ docker login

USING WEB-BASED LOGIN

Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: NVQQ-VBFN
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...

WARNING! Your credentials are stored unencrypted in '/home/tasneem/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded

tasneem@DESKTOP-0VT5601:~/task_Kube/web$ docker tag docker_imageweb tasneemsherif/docker_imageweb:latest
tasneem@DESKTOP-0VT5601:~/task_Kube/web$ docker push tasneemsherif/docker_imageweb:latest
The push refers to repository [docker.io/tasneemsherif/docker_imageweb]
5045ed8967f3: Pushed
f17478b6e8f3: Pushed
0662742b23b2: Pushed
5c91a024d899: Pushed
6b1b97dc9285: Pushed
a6b19c3d00b1: Pushed
30837a0774b9: Pushed
7cc7fe68eff6: Pushed
latest: digest: sha256:4d0bf59a522023b33ff4f5120251bebdbf680698444f06ff7d43351fea50dbcd size: 1985
```

19- Create a Deployment Using This Image:

* Deploy the Docker image from DockerHub to Kubernetes with a Deployment that has 3 replicas.

In Yaml file:

apiVersion: apps/v1

kind: Deployment

metadata:

name: web-deploy

spec:

replicas: 3

selector:

matchLabels:

name: web-pod

template:

metadata:

labels:

name: web-pod

spec:

containers:

- name: web-container

image: tasneemsherif/docker_imageweb

ports:

- containerPort: 80

Cmd:

kubectl apply -f web-deploy.yaml

kubectl get pods

```
task_Kube > web > ! web-deploy.yaml
9   spec:
19  template:
21    metadata:
23      labels:
26
27    spec:
28
29      containers:
30
31      - name: web-container
32
33        image: tasneemsherif/docker_imageweb
34        ports:
35          - containerPort: 80
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 15

- tasneem@DESKTOP-0VT5601:~/task_Kube/web\$ kubectl apply -f web-deploy.yaml
deployment.apps/web-deploy created
- tasneem@DESKTOP-0VT5601:~/task_Kube/web\$ kubectl get deploy

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment-1	2/2	2	2	12h
httpd-frontend	3/3	3	3	12h
web-deploy	0/3	3	0	3m32s

- tasneem@DESKTOP-0VT5601:~/task_Kube/web\$
- tasneem@DESKTOP-0VT5601:~/task_Kube/web\$ kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
deployment-1-858c547685-2p5vx	1/1	Running	3 (50m ago)	12h
deployment-1-858c547685-ckgxz	1/1	Running	3 (50m ago)	12h
httpd-frontend-65c89bbdc5-5r6jb	1/1	Running	3 (46m ago)	12h
httpd-frontend-65c89bbdc5-k4sxw	1/1	Running	3 (46m ago)	12h
httpd-frontend-65c89bbdc5-ms85p	1/1	Running	3 (46m ago)	12h
my-nginx	1/1	Running	0	13h
my-nginx-rs-d78qb	1/1	Running	0	13h
web-deploy-7dcdb65bff-2xhmw	0/1	ContainerCreating	0	5m52s
web-deploy-7dcdb65bff-dqfct	0/1	ContainerCreating	0	5m53s
web-deploy-7dcdb65bff-k6nm4	0/1	ContainerCreating	0	5m52s

- tasneem@DESKTOP-0VT5601:~/task_Kube/web\$

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 15

- tasneem@DESKTOP-0VT5601:~/task_Kube/web\$ kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
deployment-1-858c547685-2p5vx	1/1	Running	3 (58m ago)	12h
deployment-1-858c547685-ckgxz	1/1	Running	3 (58m ago)	12h
httpd-frontend-65c89bbdc5-5r6jb	1/1	Running	3 (54m ago)	12h
httpd-frontend-65c89bbdc5-k4sxw	1/1	Running	3 (53m ago)	12h
httpd-frontend-65c89bbdc5-ms85p	1/1	Running	3 (53m ago)	12h
my-nginx	1/1	Running	0	13h
my-nginx-rs-d78qb	1/1	Running	0	13h
web-deploy-7dcdb65bff-2xhmw	1/1	Running	0	13m
web-deploy-7dcdb65bff-dqfct	1/1	Running	0	13m
web-deploy-7dcdb65bff-k6nm4	1/1	Running	0	13m

- tasneem@DESKTOP-0VT5601:~/task_Kube/web\$