

1- Install k8s cluster (minikube) (optional you can use <https://www.katacoda.com/courses/kubernetes/playground>) => done

2- Create a pod with the name redis and with the image redis.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  bash + - [ ] [ ] ... ^ X

• [mariam@localhost Lab1]$ kubectl run redis --image=redis
pod/redis created
• [mariam@localhost Lab1]$ kubectl get pods
NAME      READY   STATUS              RESTARTS   AGE
redis     0/1     ContainerCreating   0          9s
○ [mariam@localhost Lab1]$
```

3- Create a pod with the name nginx and with the image “nginx123”
Use a pod-definition YAML file.

```
Welcome  ! nginx123.yml x [ ] ...

! nginx123.yml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: yml-nginx-pod
5  spec:
6    containers:
7    - name: nginx
8      image: nginx:123

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  bash + - [ ] [ ] ... ^ X

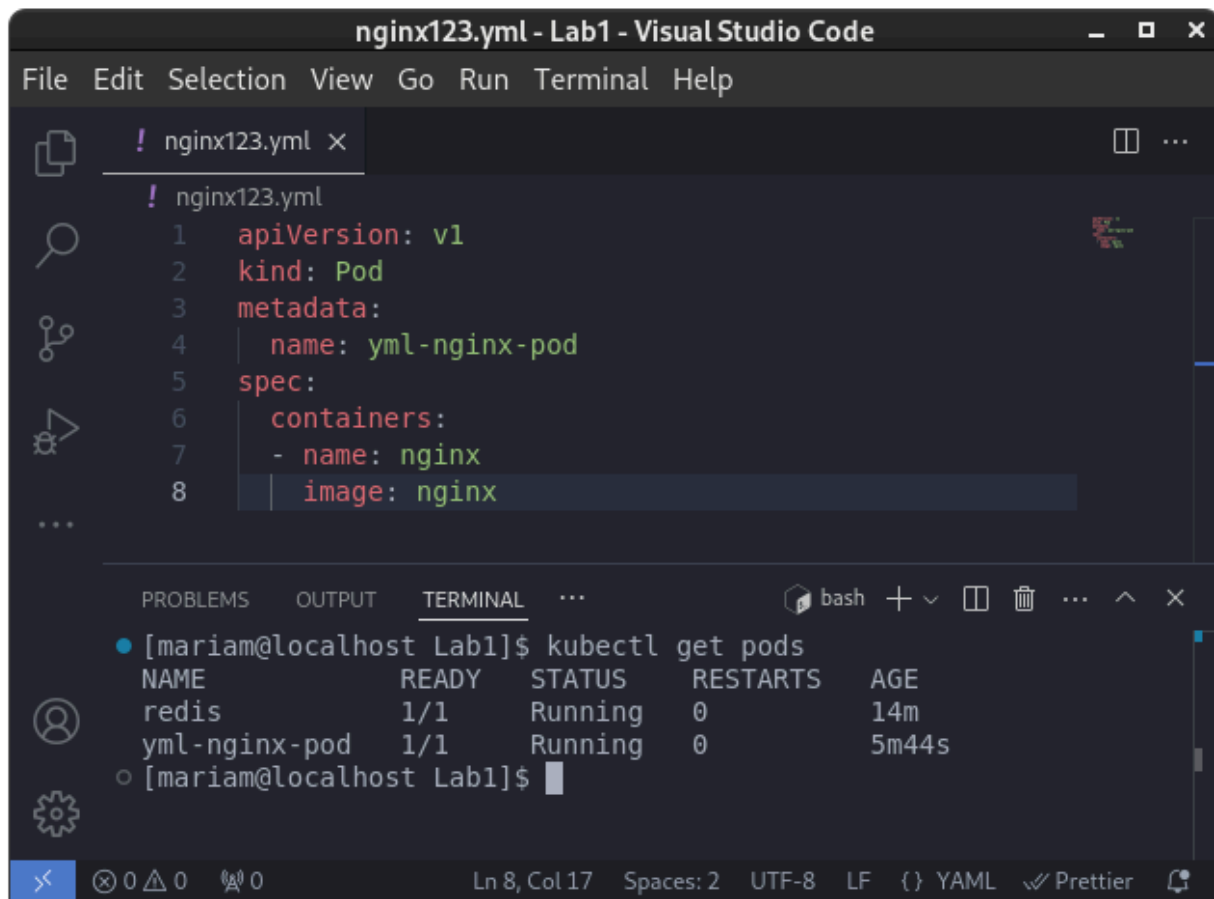
• [mariam@localhost Lab1]$ kubectl apply -f nginx123.yml
pod/yml-nginx-pod created
○ [mariam@localhost Lab1]$
```

4- What is the nginx pod status?

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  bash + - [ ] [ ] ... ^ X

• [mariam@localhost Lab1]$ kubectl get pods
NAME            READY   STATUS              RESTARTS   AGE
redis           1/1     Running             0          11m
yml-nginx-pod   0/1     ImagePullBackOff    0          2m35s
○ [mariam@localhost Lab1]$
```

5- Change the nginx pod image to “nginx” check the status again



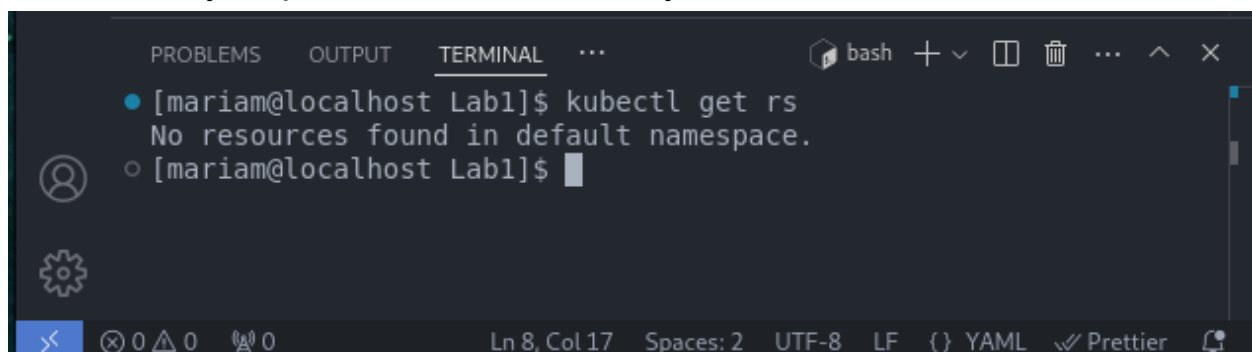
The screenshot shows the Visual Studio Code interface. The editor window displays a file named `nginx123.yml` with the following content:

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: yml-nginx-pod
5 spec:
6   containers:
7   - name: nginx
8     image: nginx
```

The terminal window at the bottom shows the command `kubectl get pods` being executed. The output is as follows:

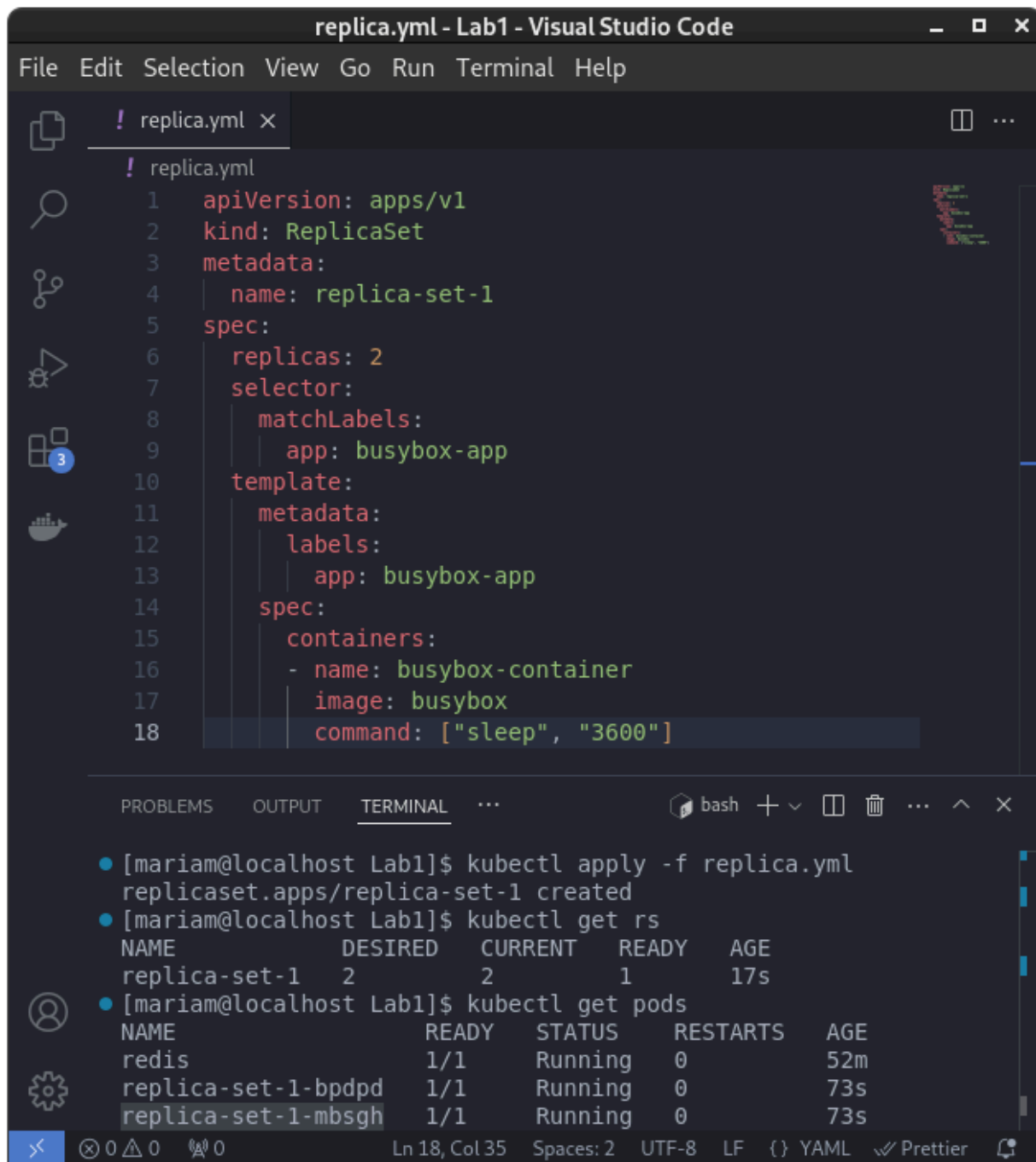
| NAME | READY | STATUS | RESTARTS | AGE |
|---------------|-------|---------|----------|-------|
| redis | 1/1 | Running | 0 | 14m |
| yml-nginx-pod | 1/1 | Running | 0 | 5m44s |

6- How many ReplicaSets exist on the system?



The screenshot shows the terminal window in Visual Studio Code. The command `kubectl get rs` has been executed, resulting in the message: "No resources found in default namespace."

7- create a ReplicaSet with
name= replica-set-1
image= busybox
replicas= 2



The image shows a Visual Studio Code window titled "replica.yml - Lab1 - Visual Studio Code". The editor displays a YAML file named "replica.yml" with the following content:

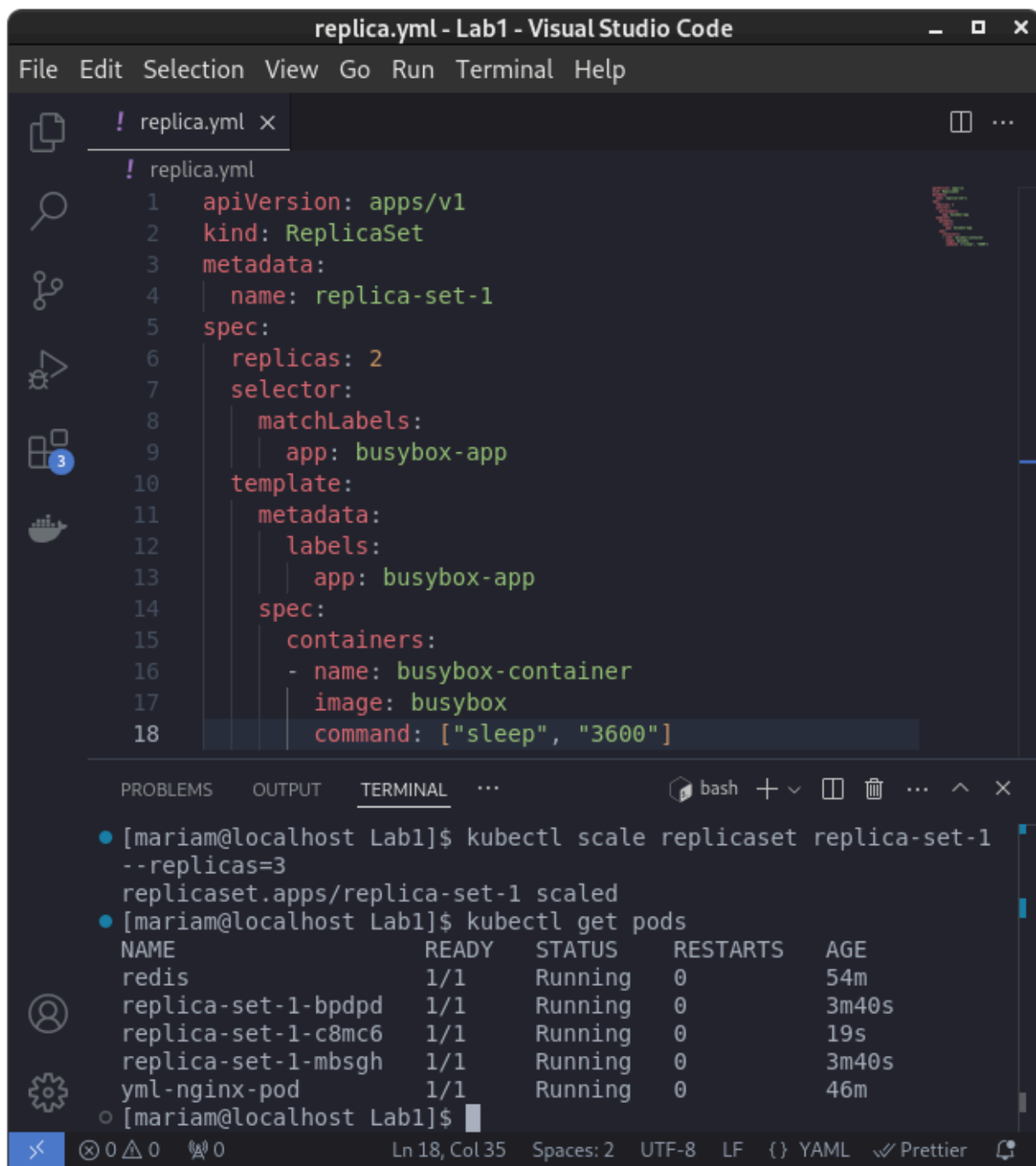
```
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: replica-set-1
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: busybox-app
10   template:
11     metadata:
12       labels:
13         app: busybox-app
14     spec:
15       containers:
16       - name: busybox-container
17         image: busybox
18         command: ["sleep", "3600"]
```

Below the editor, the TERMINAL panel shows the execution of the following commands and their output:

```
[mariam@localhost Lab1]$ kubectl apply -f replica.yml
replicaset.apps/replica-set-1 created
[mariam@localhost Lab1]$ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
replica-set-1  2         2         1       17s
[mariam@localhost Lab1]$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
redis                               1/1     Running   0           52m
replica-set-1-bpdpd                 1/1     Running   0           73s
replica-set-1-mbsgh                 1/1     Running   0           73s
```

The status bar at the bottom indicates the file is at Line 18, Column 35, with 2 spaces, UTF-8 encoding, LF line endings, and is a YAML file with Prettier formatting.

8- Scale the ReplicaSet replica-set-1 to 3 PODs.



The screenshot shows the Visual Studio Code interface with a file named `replica.yml` open. The file contains a Kubernetes ReplicaSet configuration for `replica-set-1` with 2 replicas. The terminal window shows the command `kubectl scale replicaset replica-set-1 --replicas=3` being executed, which successfully scales the ReplicaSet. Subsequently, the command `kubectl get pods` is run, displaying a table of pods. The table shows that the `replica-set-1` pods are in a `READY` state, with 3 pods ready out of 3 total.

```
replica.yml - Lab1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

! replica.yml x
! replica.yml
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: replica-set-1
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: busybox-app
10   template:
11     metadata:
12       labels:
13         app: busybox-app
14     spec:
15       containers:
16       - name: busybox-container
17         image: busybox
18         command: ["sleep", "3600"]

PROBLEMS OUTPUT TERMINAL ...
bash + v [ ] [ ] ... ^ x

• [mariam@localhost Lab1]$ kubectl scale replicaset replica-set-1 --replicas=3
replicaset.apps/replica-set-1 scaled
• [mariam@localhost Lab1]$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
redis                1/1     Running   0           54m
replica-set-1-bpdpd  1/1     Running   0           3m40s
replica-set-1-c8mc6  1/1     Running   0           19s
replica-set-1-mbsgh  1/1     Running   0           3m40s
yml-nginx-pod       1/1     Running   0           46m
○ [mariam@localhost Lab1]$
```

Ln 18, Col 35 Spaces: 2 UTF-8 LF {} YAML ✓ Prettier

9- How many PODs are READY in the replica-set-1?
= 3 PODS are ready

10- Delete any one of the 3 PODs then check How many PODs exist now?

```
PROBLEMS OUTPUT TERMINAL ... bash + - [ ] [ ] ... ^ X
• [mariam@localhost Lab1]$ kubectl delete pod replica-set-1-bpdpd
pod "replica-set-1-bpdpd" deleted
• [mariam@localhost Lab1]$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
redis               1/1     Running   0           57m
replica-set-1-c8mc6 1/1     Running   0          2m52s
replica-set-1-mbsgh 1/1     Running   0          6m13s
replica-set-1-vpf9t 1/1     Running   0          48s
yaml-nginx-pod      1/1     Running   0          49m
○ [mariam@localhost Lab1]$
```

Why are there still 3 PODs, even after you deleted one?

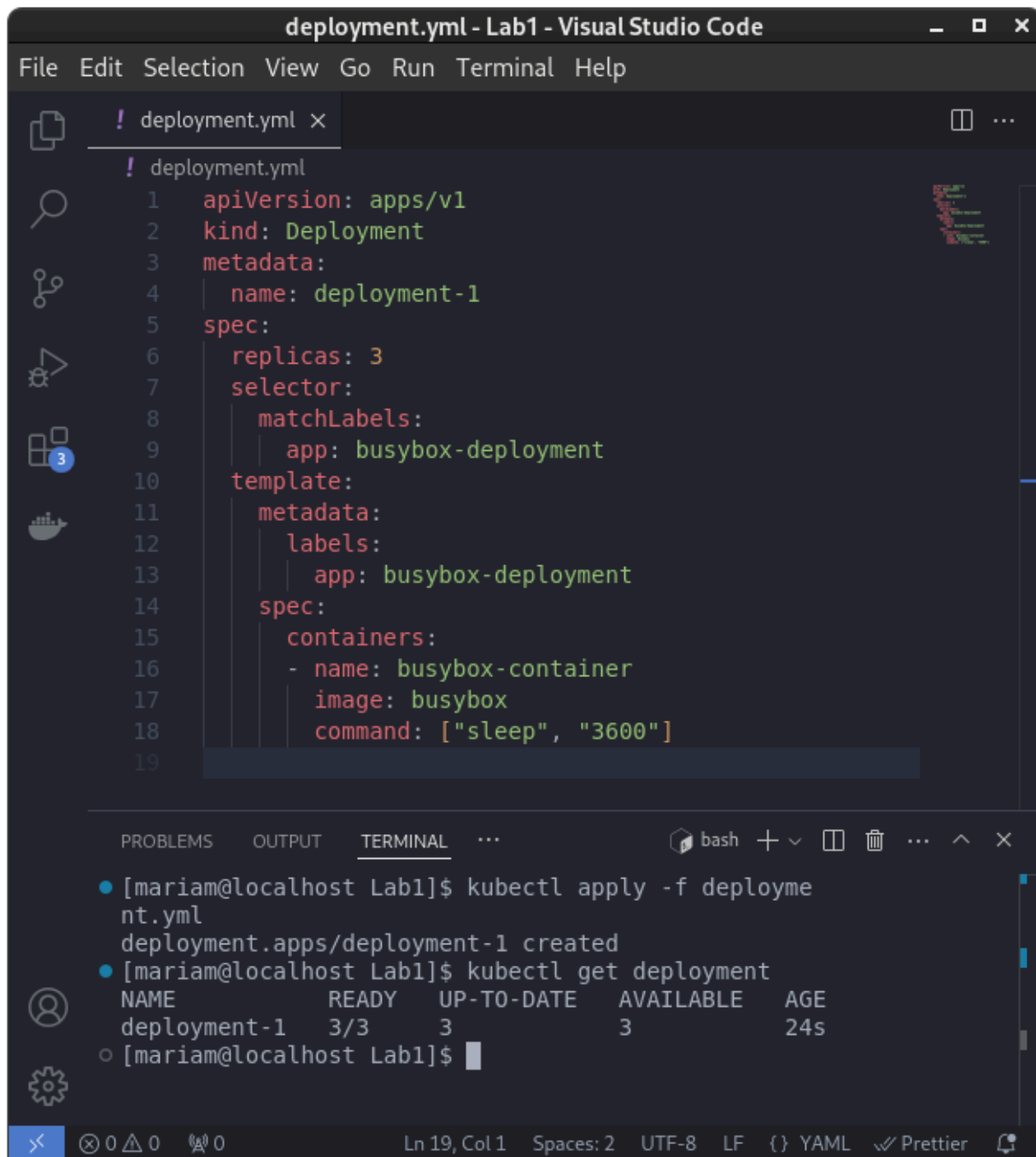
=> because the replicaset creates 3 pods automatically, if one is deleted it will create alternative one to match the value of replicas in yaml file.

If two or even the 3 pods got deleted, the replicaset will recreate 3 other pods to match replicas value.

11- How many Deployments and ReplicaSets exist on the system?

```
PROBLEMS OUTPUT TERMINAL ... bash + - [ ] [ ] ... ^ X
• [mariam@localhost Lab1]$ kubectl get deployment
No resources found in default namespace.
• [mariam@localhost Lab1]$ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
replica-set-1       3         3         3       14m
○ [mariam@localhost Lab1]$
```

12- create a Deployment with
name= deployment-1
image= busybox
replicas= 3



The screenshot shows the Visual Studio Code editor with a file named `deployment.yml` open. The file contains a Kubernetes Deployment manifest. Below the editor, the terminal shows the commands used to apply the deployment and check its status.

```
deployment.yml - Lab1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

! deployment.yml x
! deployment.yml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: deployment-1
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: busybox-deployment
10   template:
11     metadata:
12       labels:
13         app: busybox-deployment
14     spec:
15       containers:
16       - name: busybox-container
17         image: busybox
18         command: ["sleep", "3600"]
19

PROBLEMS OUTPUT TERMINAL ...
bash + v [ ] [ ] ... ^ x

• [mariam@localhost Lab1]$ kubectl apply -f deployment.yml
deployment.apps/deployment-1 created
• [mariam@localhost Lab1]$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1  3/3     3            3           24s
○ [mariam@localhost Lab1]$
```

Ln 19, Col 1 Spaces: 2 UTF-8 LF {} YAML ✓ Prettier

13- How many Deployments and ReplicaSets exist on the system now?

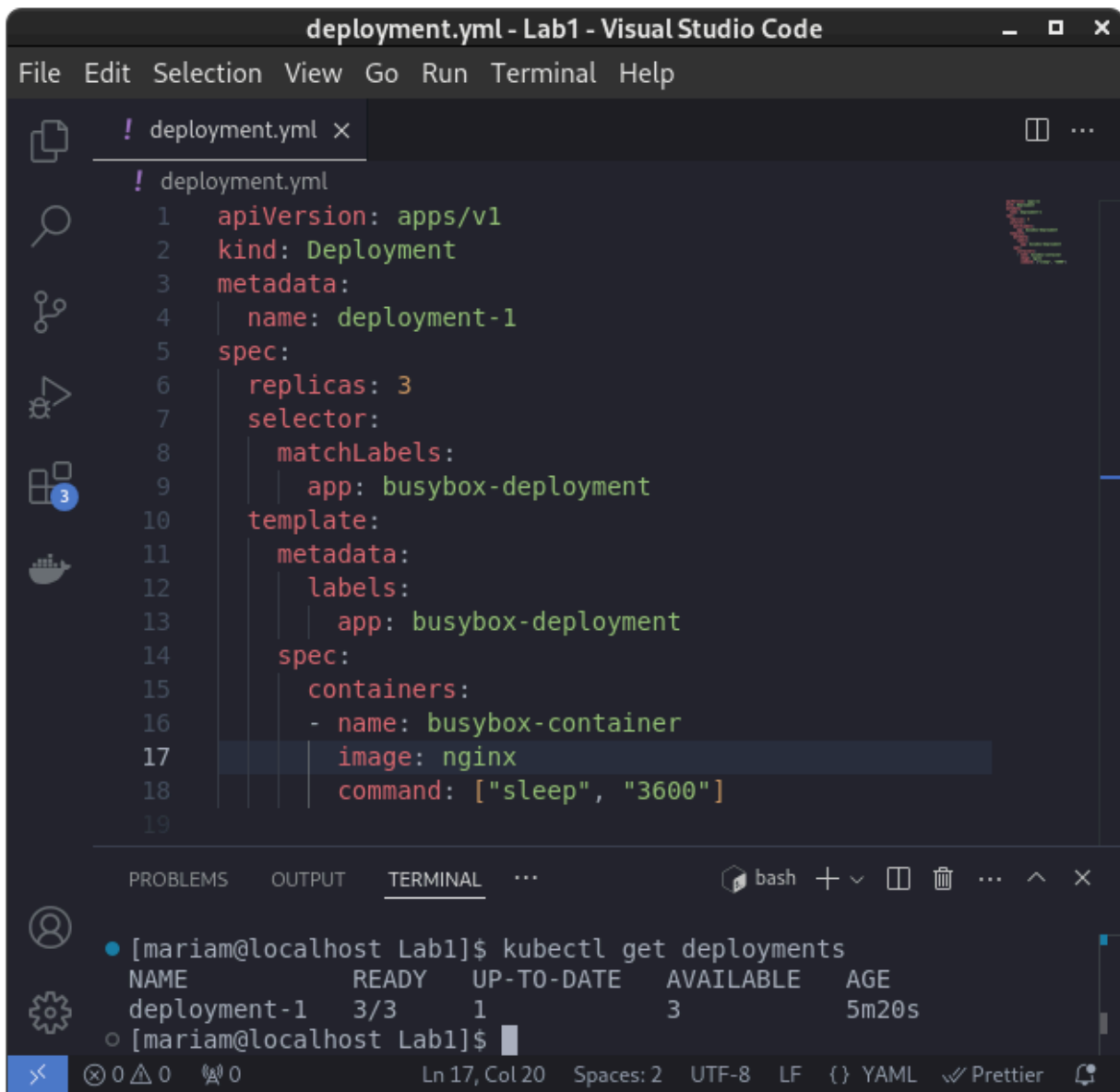
```
PROBLEMS OUTPUT TERMINAL ... bash + - [ ] [X] ... ^ X
• [mariam@localhost Lab1]$ kubectl get deployments,replicasets
NAME                                READY  UP-TO-DATE  AVAILABLE
AGE
deployment.apps/deployment-1        3/3    3            3
2m3s

NAME                                DESIRED  CURRENT  REA
DY    AGE
replicaset.apps/deployment-1-b7856cdc6  3        3        3
2m3s
replicaset.apps/replica-set-1          3        3        3
21m
○ [mariam@localhost Lab1]$
```

14- How many pods are ready with the deployment-1?

= 3

15- Update deployment-1 image to nginx then check the ready pods again



The screenshot shows the Visual Studio Code editor with a file named `deployment.yaml` open. The file contains a Kubernetes Deployment configuration for `deployment-1`. The `image` field in the `containers` section is currently set to `nginx` and is highlighted. Below the editor, the terminal window shows the command `kubectl get deployments` being executed, resulting in a table of deployment status.

```
deployment.yaml - Lab1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

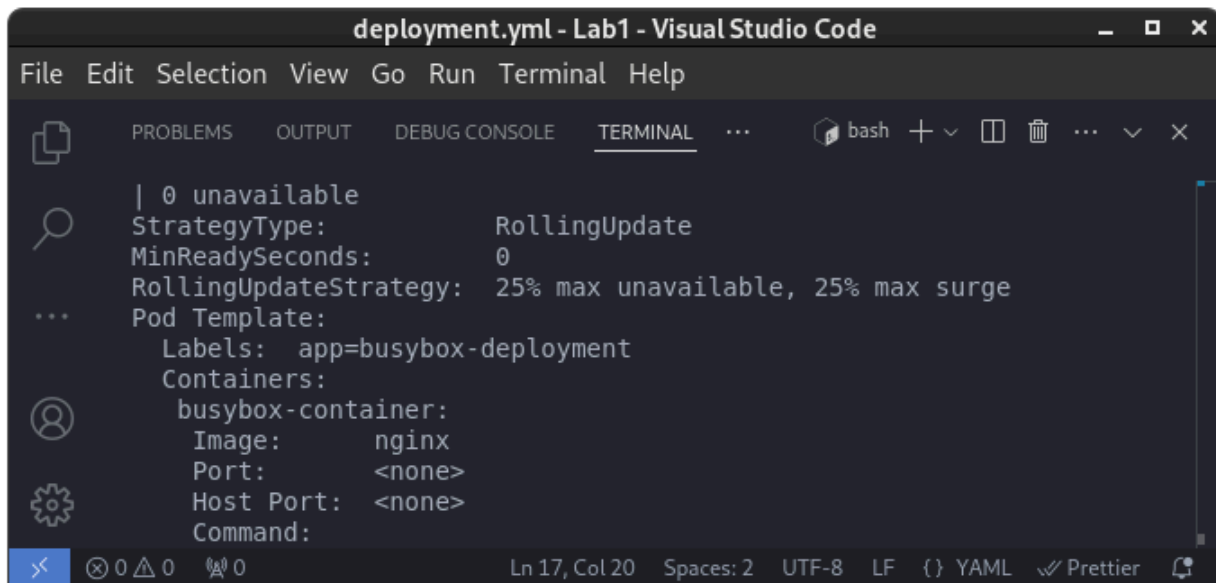
! deployment.yaml x
! deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: deployment-1
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: busybox-deployment
10   template:
11     metadata:
12       labels:
13         app: busybox-deployment
14     spec:
15       containers:
16       - name: busybox-container
17         image: nginx
18         command: ["sleep", "3600"]
19

PROBLEMS OUTPUT TERMINAL ...
bash + - [ ] [ ] ... ^ x

• [mariam@localhost Lab1]$ kubectl get deployments
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
deployment-1  3/3    1           3          5m20s
○ [mariam@localhost Lab1]$
```

Ln 17, Col 20 Spaces: 2 UTF-8 LF {} YAML ✓ Prettier

16- Run `kubectl describe deployment deployment-1` and check events
What is the deployment strategy used to upgrade the deployment-1?
=> RollingUpdate, and the image is nginx



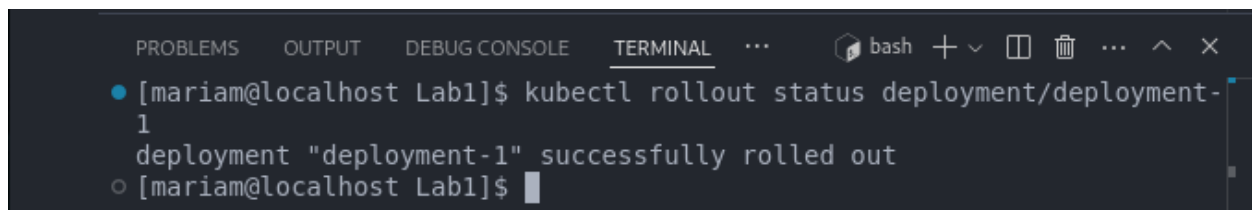
The screenshot shows the Visual Studio Code editor with the file `deployment.yml - Lab1` open. The `TERMINAL` tab is active, displaying the output of `kubectl describe deployment deployment-1`. The output shows the deployment strategy is `RollingUpdate` and the image is `nginx`.

```
deployment.yml - Lab1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

| 0 unavailable
StrategyType:      RollingUpdate
MinReadySeconds:   0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=busybox-deployment
  Containers:
    busybox-container:
      Image:      nginx
      Port:       <none>
      Host Port:  <none>
      Command:

Ln 17, Col 20  Spaces: 2  UTF-8  LF  {}  YAML  ✓ Prettier
```

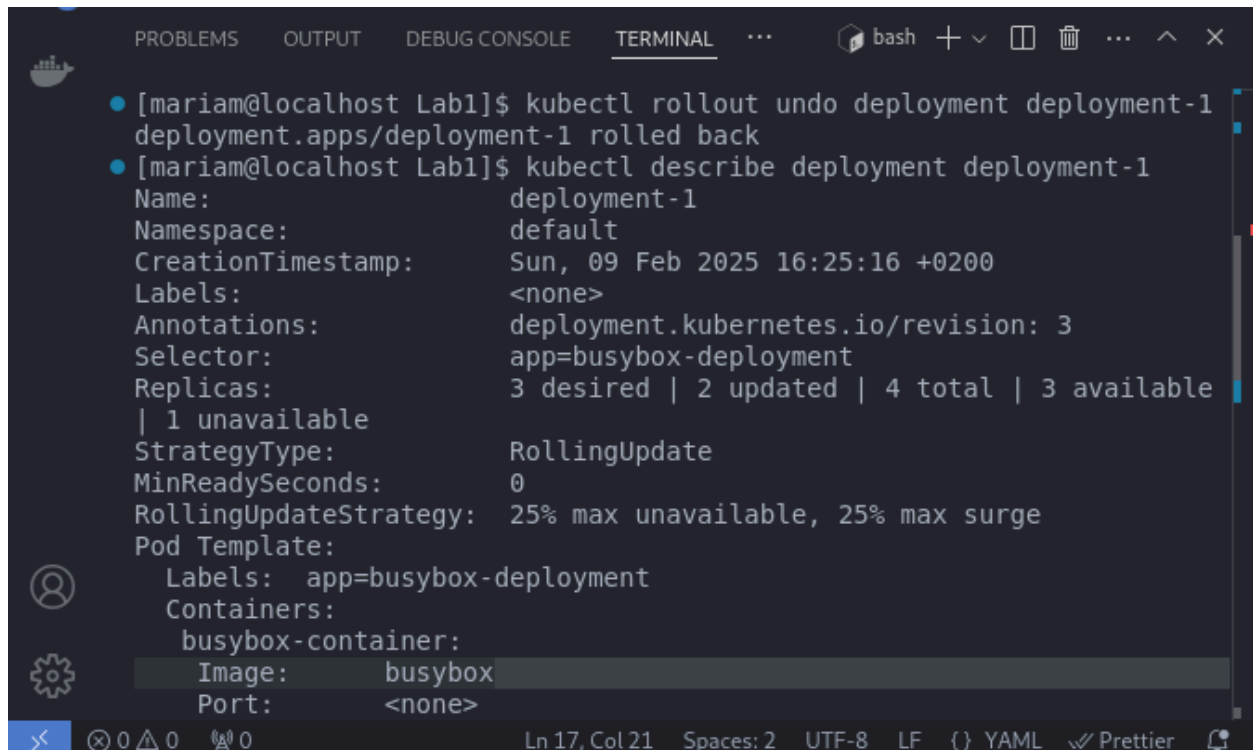
17- Rollback the deployment-1



The screenshot shows a terminal window with the command `kubectl rollout status deployment/deployment-1` being executed. The output indicates that the deployment was successfully rolled out.

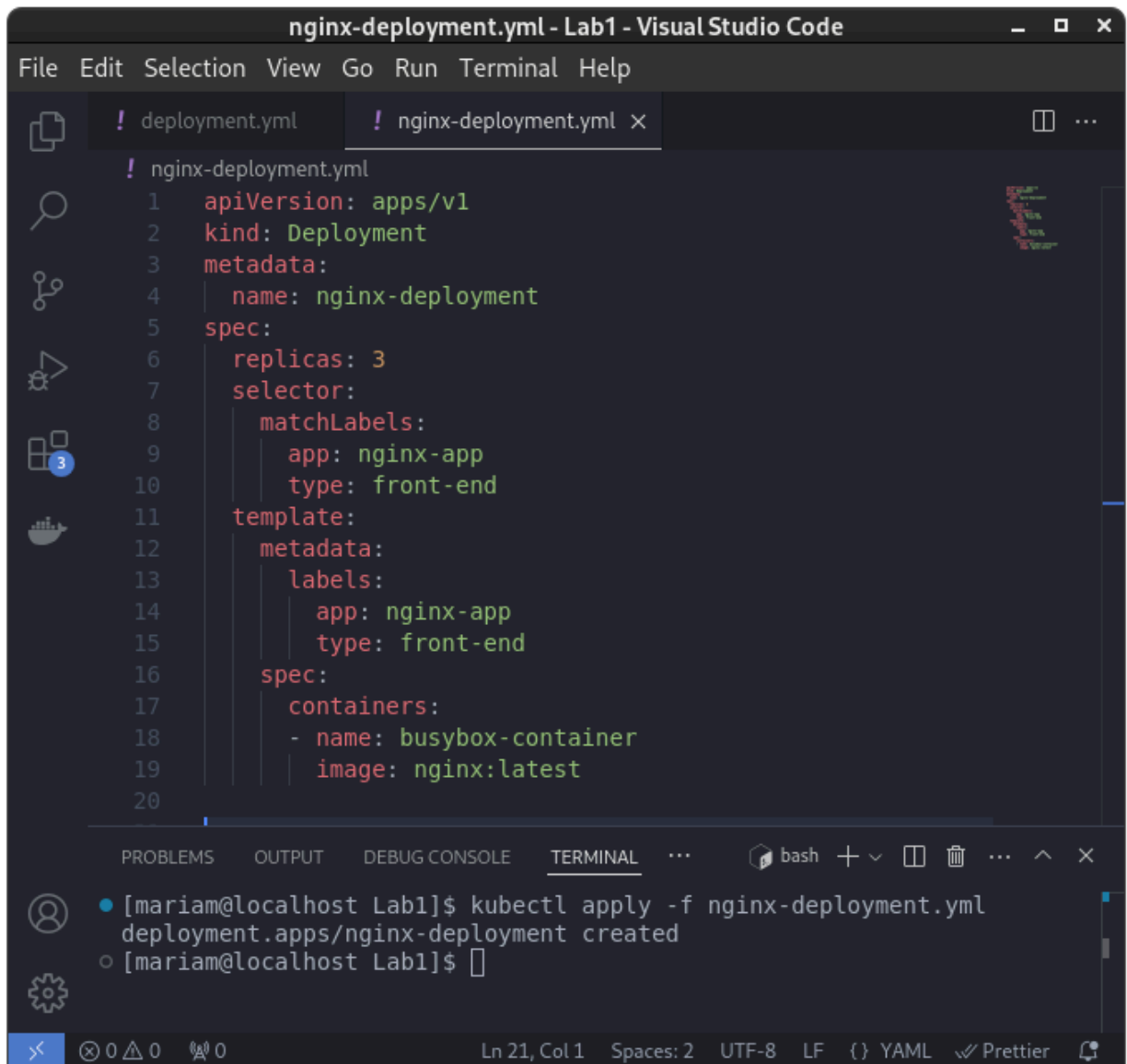
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... bash + - [ ] [ ] ... ^ x
• [mariam@localhost Lab1]$ kubectl rollout status deployment/deployment-1
deployment "deployment-1" successfully rolled out
○ [mariam@localhost Lab1]$
```

What is the used image with the deployment-1?



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... bash + - [ ] [X] ... ^ X
• [mariam@localhost Lab1]$ kubectl rollout undo deployment deployment-1
deployment.apps/deployment-1 rolled back
• [mariam@localhost Lab1]$ kubectl describe deployment deployment-1
Name: deployment-1
Namespace: default
CreationTimestamp: Sun, 09 Feb 2025 16:25:16 +0200
Labels: <none>
Annotations: deployment.kubernetes.io/revision: 3
Selector: app=busybox-deployment
Replicas: 3 desired | 2 updated | 4 total | 3 available
| 1 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=busybox-deployment
  Containers:
    busybox-container:
      Image: busybox
      Port: <none>
```

18- Create a deployment using nginx image with latest tag only and remember to mention tag i.e nginx:latest and name it as nginx-deployment. App labels should be app: nginx-app and type: front-end. The container should be named as nginx-container; also make sure replica counts are 3.



The screenshot shows the Visual Studio Code editor with a file named `nginx-deployment.yml` open. The file contains a Kubernetes Deployment manifest. The manifest specifies 3 replicas, labels `app: nginx-app` and `type: front-end`, and a container named `nginx-container` using the `nginx:latest` image. Below the editor, the terminal shows the command `kubectl apply -f nginx-deployment.yml` being executed, resulting in the deployment being created.

```
nginx-deployment.yml - Lab1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help

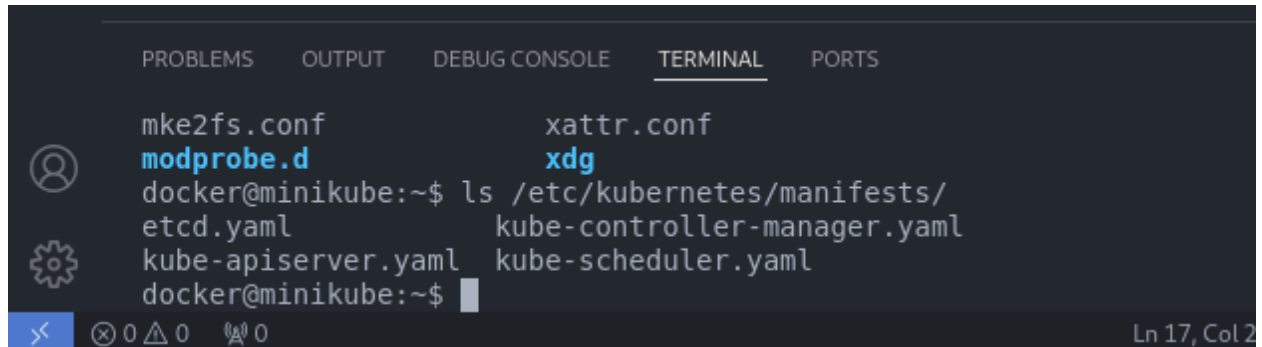
! deployment.yml ! nginx-deployment.yml x

! nginx-deployment.yml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx-deployment
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: nginx-app
10       type: front-end
11  template:
12    metadata:
13      labels:
14        app: nginx-app
15        type: front-end
16    spec:
17      containers:
18      - name: nginx-container
19        image: nginx:latest
20

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... bash + - [x] [x] ... ^ x
• [mariam@localhost Lab1]$ kubectl apply -f nginx-deployment.yml
  deployment.apps/nginx-deployment created
○ [mariam@localhost Lab1]$
```

Ln 21, Col 1 Spaces: 2 UTF-8 LF {} YAML ✓ Prettier

19- How many static pods exist in this cluster?



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

mke2fs.conf      xattr.conf
modprobe.d       xdg
docker@minikube:~$ ls /etc/kubernetes/manifests/
etcd.yaml        kube-controller-manager.yaml
kube-apiserver.yaml kube-scheduler.yaml
docker@minikube:~$
```

Ln 17, Col 2

20- create a static pod nginx

First => minikube ssh

Then => vi /etc/kubernetes/manifests/my-static-pod.yml



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash +

[mariam@localhost Lab1]$ kubectl get pods
NAME                                READY  STATUS   RESTARTS   AGE
deployment-1-b7856cdc6-2ll6l        1/1    Running  0          10m
deployment-1-b7856cdc6-8hs8z        1/1    Running  0          10m
deployment-1-b7856cdc6-n8t59        1/1    Running  0          10m
my-pod-minikube                     1/1    Running  0          34s
redis                                1/1    Running  0          114m
replica-set-1-c8mc6                 1/1    Running  0          59m
replica-set-1-mbsgh                 1/1    Running  1 (2m55s ago) 63m
replica-set-1-vpf9t                 1/1    Running  0          57m
yml-nginx-pod                       1/1    Running  0          105m
[mariam@localhost Lab1]$
```