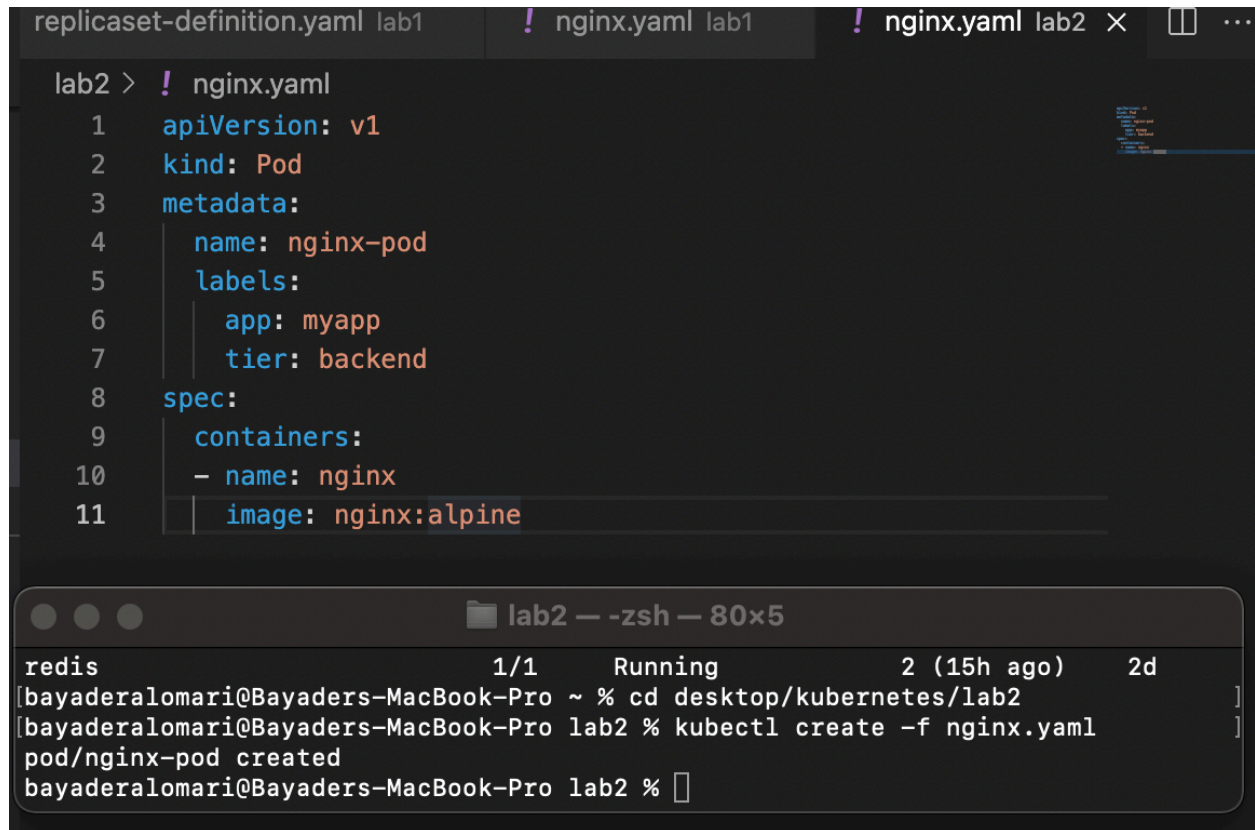


1- Deploy a pod named `nginx-pod` using the `nginx:alpine` image with the labels set to `tier=backend`.

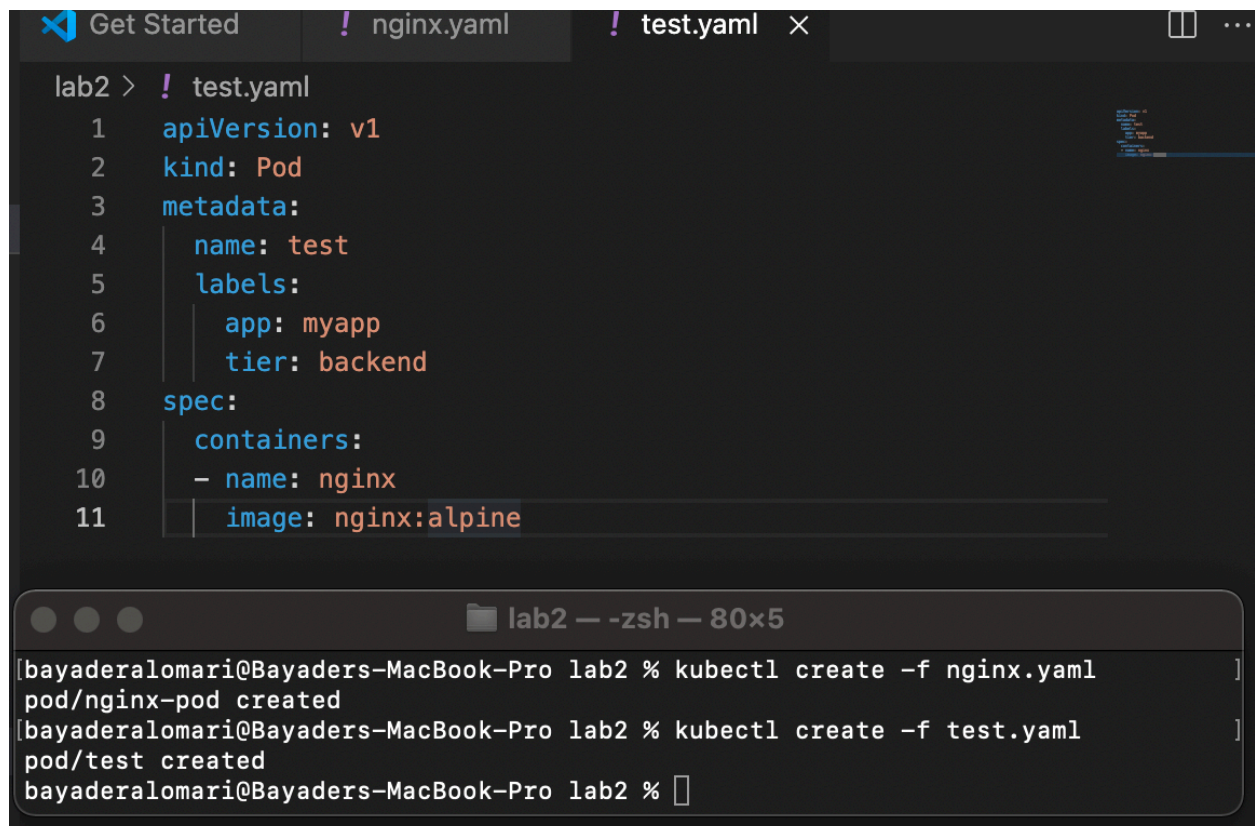


The screenshot shows a code editor with a file named `nginx.yaml` open. The file contains a Kubernetes Pod definition. Below the editor, a terminal window shows the command `kubectl create -f nginx.yaml` being executed, resulting in the pod `nginx-pod` being created.

```
replicaset-definition.yaml lab1 | ! nginx.yaml lab1 | ! nginx.yaml lab2 X ...
lab2 > ! nginx.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx-pod
5    labels:
6      app: myapp
7      tier: backend
8  spec:
9    containers:
10   - name: nginx
11     image: nginx:alpine
```

```
lab2 -- -zsh -- 80x5
redis 1/1 Running 2 (15h ago) 2d
[bayaderalomari@Bayaders-MacBook-Pro ~ % cd desktop/kubernetes/lab2
[bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl create -f nginx.yaml
pod/nginx-pod created
bayaderalomari@Bayaders-MacBook-Pro lab2 % ]
```

2- Deploy a test pod using the `nginx:alpine` image.



The screenshot shows a code editor with a file named `test.yaml` open. The file contains a Kubernetes Pod definition. Below the editor, a terminal window shows the command `kubectl create -f test.yaml` being executed, resulting in the pod `test` being created.

```
Get Started | ! nginx.yaml | ! test.yaml X ...
lab2 > ! test.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: test
5    labels:
6      app: myapp
7      tier: backend
8  spec:
9    containers:
10   - name: nginx
11     image: nginx:alpine
```

```
lab2 -- -zsh -- 80x5
[bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl create -f nginx.yaml
pod/nginx-pod created
[bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl create -f test.yaml
pod/test created
bayaderalomari@Bayaders-MacBook-Pro lab2 % ]
```

3- Create a service `backend-service` to expose the backend application within the cluster on port 80.

```
lab2 > ! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: backend-service
5  spec:
6    selector:
7      tier: backend
8    ports:
9      - protocol: TCP
10      port: 80
11

o struct field ServiceSpec.spec.selector of type string
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl create -f service.yaml --vali
date=false
service/backend-service created
bayaderalomari@Bayaders-MacBook-Pro lab2 %
```

4- try to curl the backend-service from the test pod. What is the response?

```
172.17.0.17   minikube   <none>           <none>
web-app-67579dd966-svdr8      1/1      Running          0           9m
172.17.0.16   minikube   <none>           <none>
bayaderalomari@Bayaders-MacBook-Pro lab2 % curl http://172.17.0.15:80
```

5- Create a deployment named `web-app` using the image `nginx` with 2 replicas

```
lab2 > ! deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: web-app
5    labels:
6      app: nginx
7  spec:
8    replicas: 2
9    selector:
10     matchLabels:
11       app: nginx
12    template:
13     metadata:
14       labels:
15         app: nginx
16     spec:
17       containers:
18         - name: nginx
19           image: nginx:1.14.2

redis        1/1      Running          2 (16h ago)    2d1h
test         1/1      Running          0              31m
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl apply -f deployment.yaml
deployment.apps/web-app created
bayaderalomari@Bayaders-MacBook-Pro lab2 %
```

6- Expose the `web-app` as service `web-app-service` application on port 80 and nodeport 30082 on the nodes on the cluster

```
lab2 > ! web-app-service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: web-app-service
5  spec:
6    selector:
7      app: nginx
8    type: NodePort
9    ports:
10     - targetPort: 80
11       port: 80
12       nodePort: 30082
```

```
lab2 — zsh — 82x5
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl apply -f web-app-service.yaml
service/web-app-service created
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl apply -f web-app-service.yaml
service/web-app-service unchanged
bayaderalomari@Bayaders-MacBook-Pro lab2 %
```

7- access the web app from the node

```
minikube ready control-plane,master 40s/v1.25.3 192.168.49.2 <none> Ubuntu 20.04.2 LTS
bayaderalomari@Bayaders-MacBook-Pro ~ % curl 192.168.49.2:80
```

8- How many Nodes exist on the system?

```
minikube    Ready    control-plane,master    2d2h    v1.23.3
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl get no
NAME        STATUS    ROLES    AGE    VERSION
minikube    Ready    control-plane,master    2d2h    v1.23.3
bayaderalomari@Bayaders-MacBook-Pro lab2 %
```

9- Do you see any taints on master ?

```
CreationTimestamp: Sat, 05 Mar 2022 13:10:22 +0300
Taints:           <none>
Unschedulable:    false
Lease:            volumes.kubernetes.io/controller-managed-attach-detach: true
HolderIdentity:    minikube
```



```

bayaderalomari@Bayaders-MacBook-Pro ~ % kubectl label nodes minikube color=blue
node/minikube labeled
bayaderalomari@Bayaders-MacBook-Pro ~ % kubectl get no
NAME        STATUS    ROLES                  AGE      VERSION
minikube    Ready     control-plane,master   2d21h    v1.23.3
bayaderalomari@Bayaders-MacBook-Pro ~ % kubectl get nodes --show-labels
NAME        STATUS    ROLES                  AGE      VERSION   LABELS
minikube    Ready     control-plane,master   2d21h    v1.23.3    beta.kubernetes.io/arch=arm64,kubernetes.io/hostname=minikube,kubernetes.io/commit=362d5fdc0a3db3e389b3d3f1034e8023e72bd3a7,minikube.k8s.io/name=minikube,kubernetes.io/os=linux,color=blue

```

10- Apply  
a label

color=blue

to the master node

```

lab2 > ! blue-deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: blue
5    labels:
6      app: nginx
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: nginx
12    template:
13     metadata:
14       labels:
15         app: nginx
16     spec:
17       affinity:
18         podAntiAffinity:
19           requiredDuringSchedulingIgnoredDuringExecution:
20             - labelSelector:
21                 matchExpressions:
22                   - key: color
23                     operator: In
24                     values:
25                       - blue
26             topologyKey: "kubernetes.io/hostname"
27     containers:
28       - name: nginx
29         image: nginx:1.14.2

```

NOMINATED NODE	READINESS GATES						
blue-96d54cb5d-kfzpz	1/1	Running	0	35s	172.17.0.18	minikube	
<none>	<none>						
blue-96d54cb5d-skppk	1/1	Running	0	35s	172.17.0.20	minikube	
<none>	<none>						
blue-96d54cb5d-zv2gh	1/1	Running	0	35s	172.17.0.19	minikube	
<none>	<none>						

11- Create a new deployment named `blue` with the `nginx` image and 3 replicas  
Set Node Affinity to the deployment to place the pods on `master` only  
NodeAffinity: requiredDuringSchedulingIgnoredDuringExecution  
Key: `color`  
values: `blue`

12- How many `DaemonSets` are created in the cluster in all namespaces?

```
error: the server doesn't have a resource type "daemonset"
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl get daemonset --all-namespaces
error: the server doesn't have a resource type "daemonset"
bayaderalomari@Bayaders-MacBook-Pro lab2 %
```

13- what `DaemonSets` exist on the `kube-system` namespace?

```
error: the server doesn't have a resource type "daemonset"
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl get daemonset -n kube-system
error: the server doesn't have a resource type "daemonset"
bayaderalomari@Bayaders-MacBook-Pro lab2 %
```

14- What is the image used by the POD deployed by the `kube-proxy` `DaemonSet`

```
kube-proxy:
  Container ID:  docker://b6812f8007c5fce3396781ca3a3059fb37e9306ce88d573c75278958f9915ce3
  Image:         k8s.gcr.io/kube-proxy:v1.23.3
  Image ID:     docker-pullable://k8s.gcr.io/kube-proxy@sha256:def87f007b49d50693aed83d4703d0e56c69ae286154b1c7a20cd1b3a320cf7c
```

15- Deploy a `DaemonSet` for `FluentD` Logging. Use the given specifications.

Name: `elasticsearch`

Namespace: `kube-system`

Image: [k8s.gcr.io/fluentd-elasticsearch:1.20](https://k8s.gcr.io/fluentd-elasticsearch:1.20)

16- Create a taint on `node01` with key of `spray`, value of `mortein` and effect of `NoSchedule`

17- Create a new pod named `mosquito` with the `NGINX` image

18- What is the state of the `mosquito` POD?

19- Create another pod named `bee` with the `NGINX` image, which has a toleration set to the taint `Mortein`

Image name: `nginx`

Key: `spray`

Value: `mortein`

Effect: `NoSchedule`

Status: `Running`

```

1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: bee
5    labels:
6      app: myapp
7      tier: backend
8  spec:
9    containers:
10     - name: nginx
11       image: nginx:alpine
12    tolerations:
13     - key: spray
14       Value: mortein
15     effect: NoSchedule

```

```

lab2 - zsh - 109x5
field "Value" in io.k8s.api.core.v1.Toleration; if you choose to ignore these errors, turn validation off with --validate=false
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl apply -f bee.yaml --validate=false
pod/bee created
bayaderalomari@Bayaders-MacBook-Pro lab2 %

```

```

22     effect: NoSchedule
23     containers:
24       - name: fluentd-elasticsearch
25       image: k8s.gcr.io/fluentd-elasticsearch:1.20

```

```

lab2 - zsh - 109x6
Events: <none>
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl apply -f daemonset.yaml --validate=false
daemonset.apps/fluentd-elasticsearch created
bayaderalomari@Bayaders-MacBook-Pro lab2 %

```

20- Remove the taint on master/controlplane, which currently has the taint effect of NoSchedule

```

CreationTimestamp: Tue, 08 Mar 2022 11:56:02 +0300
Taints: <none>
Unschedulable: false

```

21- What is the state of the pod `mosquito` now and Which node is the POD `mosquito` on?

S								
bee	1/1	Running	0	5m40s	10.244.0.4	multinode-demo	<none>	<none>
mosquito	1/1	Running	0	12m	10.244.0.3	multinode-demo	<none>	<none>

22- Create a job countdown-job.

The container should be named as container-countdown-job

Use image `debian:latest`, and restart policy should be `Never`.

Use command `for i in ten nine eight seven six five four three two one ; do echo $i ; done`

```
1  apiVersion: batch/v1
2  kind: Job
3  metadata:
4    name: countdown-job
5  spec:
6    template:
7      spec:
8        containers:
9          - name: container-countdown-job
10            image: debian:latest
11            command: [for i in ten nine eight seven six five four three two one ; do echo $i ; done]
12            restartPolicy: Never
```

lab2 — -zsh — 109x5

bee	1/1	Running	0	5m40s	10.244.0.4	multinode-demo	<none>	<none>
mosquito	1/1	Running	0	12m	10.244.0.3	multinode-demo	<none>	<none>

```
bayaderalomari@Bayaders-MacBook-Pro lab2 % kubectl apply -f countdown.yaml --validate=false
job.batch/countdown-job created
bayaderalomari@Bayaders-MacBook-Pro lab2 %
```