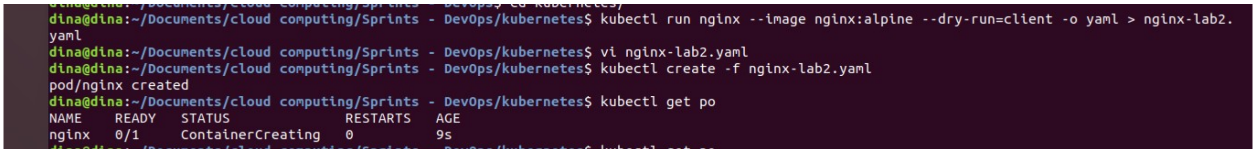
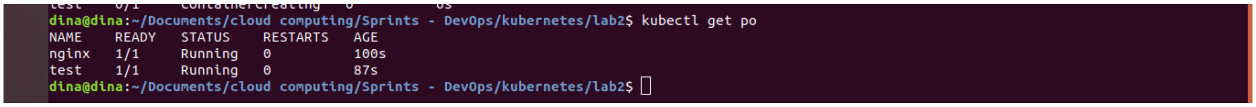
1- Deploy a pod named `nginx-pod` using the `nginx:alpine` image with the labels set to `tier=backend`.



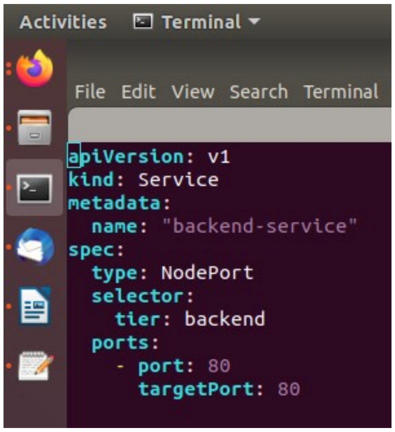
```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes$ kubectl run nginx --image nginx:alpine --dry-run=client -o yaml > nginx-lab2.
yaml
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes$ vi nginx-lab2.yaml
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes$ kubectl create -f nginx-lab2.yaml
pod/nginx created
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes$ kubectl get po
NAME    READY   STATUS              RESTARTS   AGE
nginx   0/1     ContainerCreating   0          9s
```

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



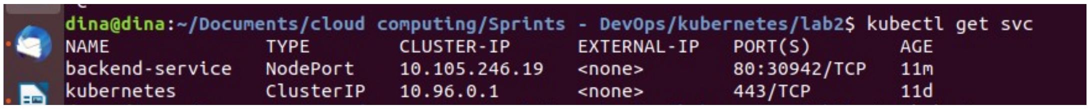
```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get po
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          100s
test    1/1     Running   0          87s
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$
```

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

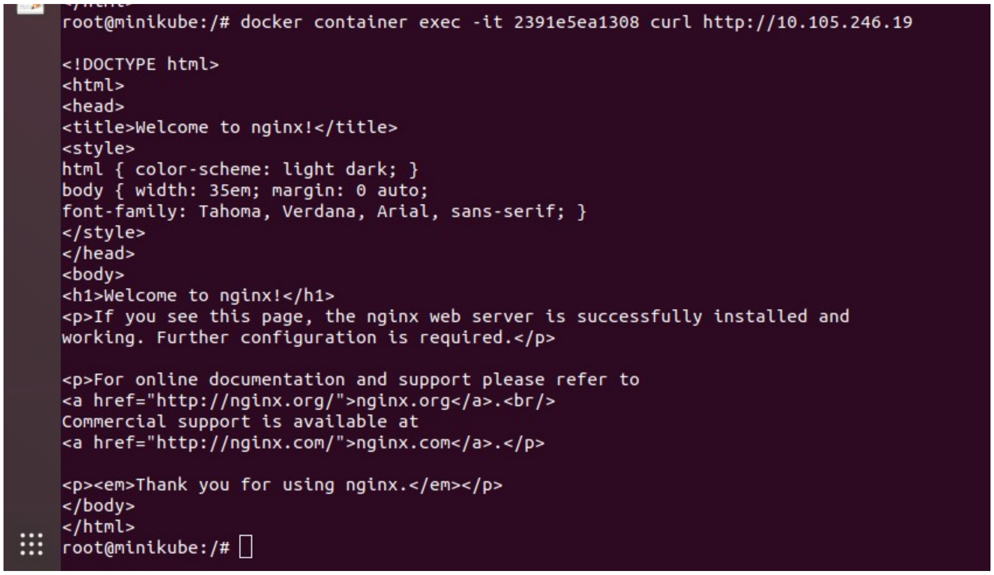


```
apiVersion: v1
kind: Service
metadata:
  name: "backend-service"
spec:
  type: NodePort
  selector:
    tier: backend
  ports:
    - port: 80
      targetPort: 80
```

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

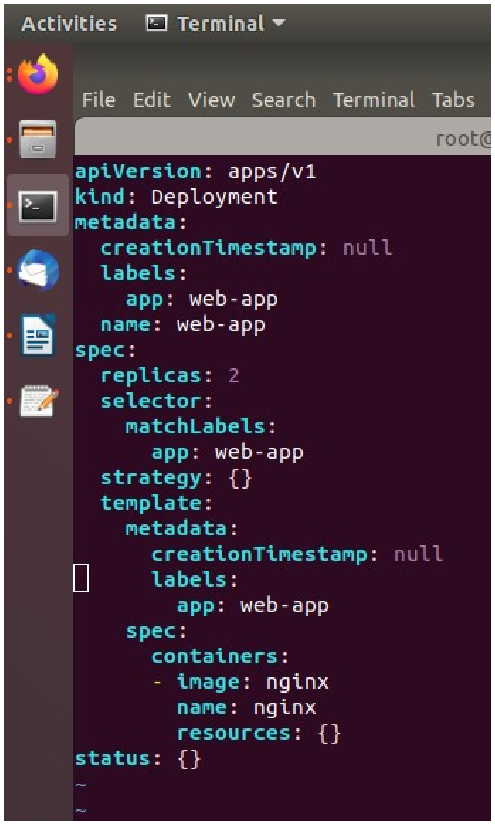


```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get svc
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
backend-service   NodePort    10.105.246.19   <none>        80:30942/TCP   11m
kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP        11d
```

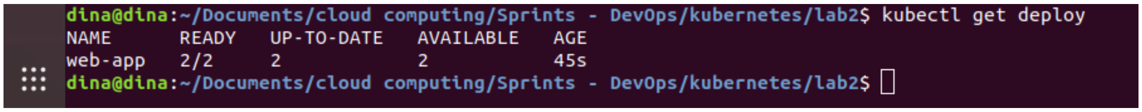


```
root@minikube:/# docker container exec -it 2391e5ea1308 curl http://10.105.246.19

<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@minikube:/#
```

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

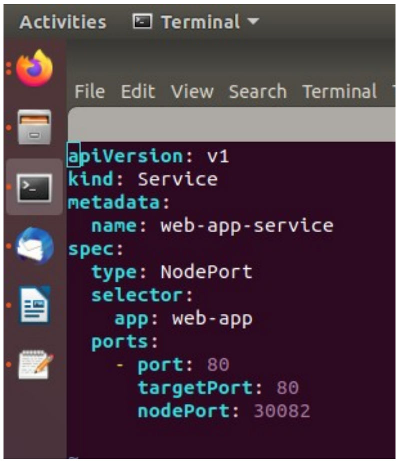


```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: web-app
  name: web-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-app
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: web-app
    spec:
      containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
~
~
```



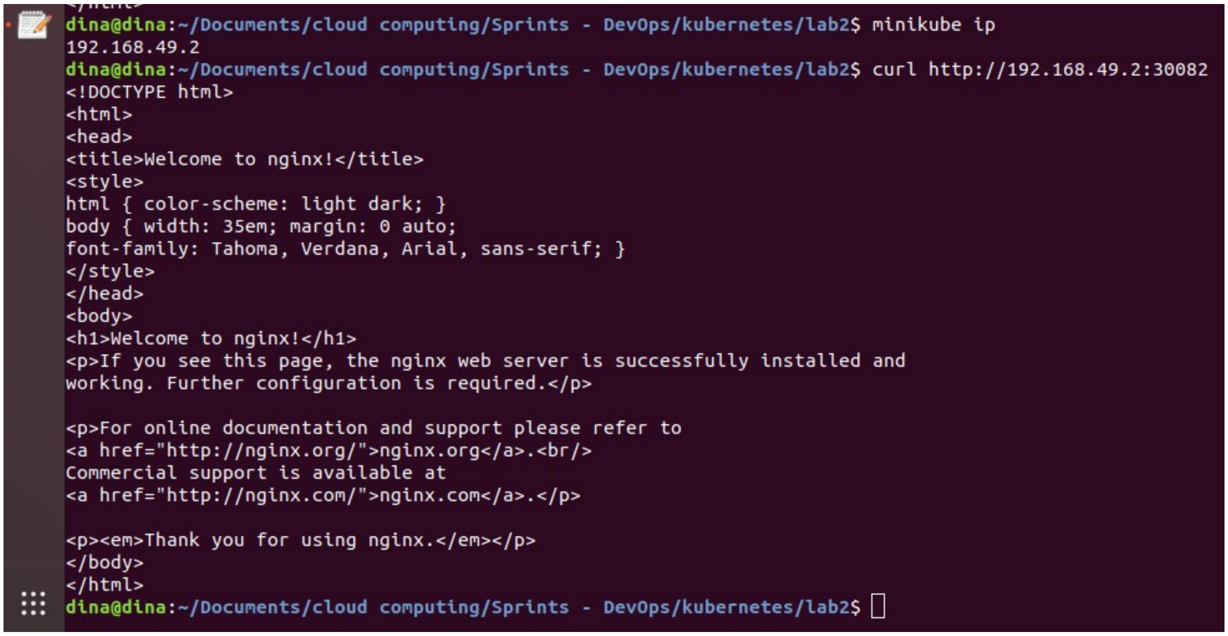
```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
web-app   2/2     2            2           45s
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

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



```
apiVersion: v1
kind: Service
metadata:
  name: web-app-service
spec:
  type: NodePort
  selector:
    app: web-app
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30082
```

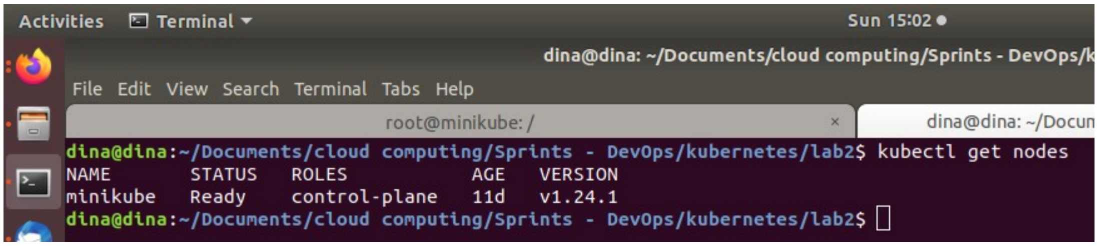
7- access the web app from the node



```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ minikube ip
192.168.49.2
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ curl http://192.168.49.2:30082
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

8- How many Nodes exist on the system?

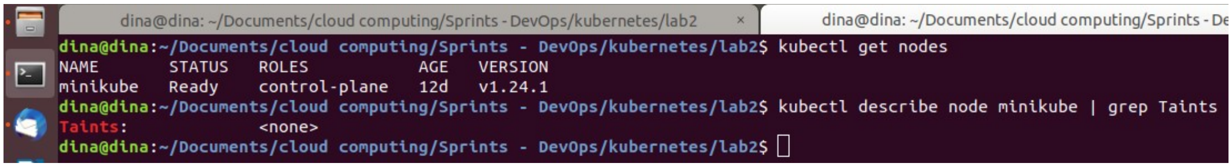
There is 1 node.


Activities
Terminal
Sun 15:02
dina@dina: ~/Documents/cloud computing/Sprints - DevOps/k
File Edit View Search Terminal Tabs Help
root@minikube: /
dina@dina: ~/Docun

```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   11d   v1.24.1
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

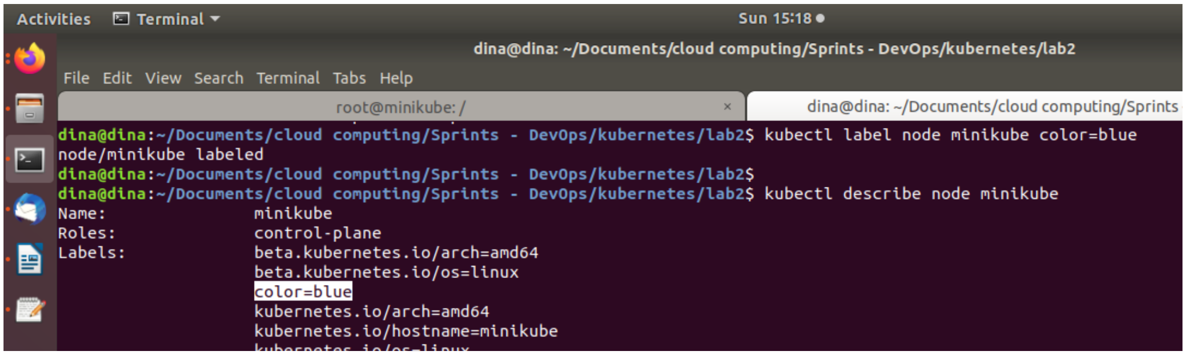
9- Do you see any taints on `master` ?

No taints on the master node.


dina@dina: ~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2
dina@dina: ~/Documents/cloud computing/Sprints - De

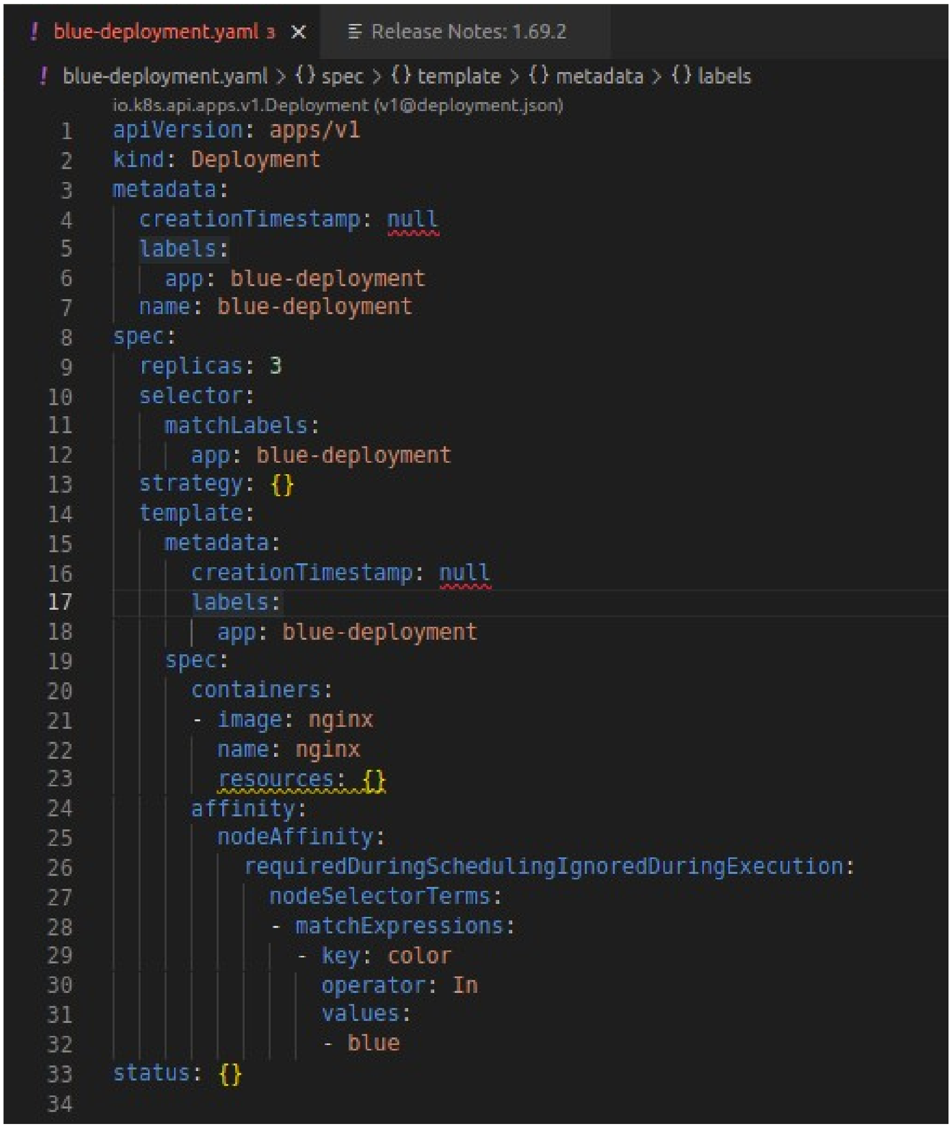
```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   12d   v1.24.1
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl describe node minikube | grep Taints
Taints:             <none>
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

10- Apply a label `color=blue` to the master node


Activities
Terminal
Sun 15:18
dina@dina: ~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2
File Edit View Search Terminal Tabs Help
root@minikube: /
dina@dina: ~/Documents/cloud computing/Sprints

```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl label node minikube color=blue
node/minikube labeled
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl describe node minikube
Name:               minikube
Roles:              control-plane
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    color=blue
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=minikube
                    kubernetes.io/os=linux
```

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


! blue-deployment.yaml 3 ×
Release Notes: 1.69.2
! blue-deployment.yaml > {} spec > {} template > {} metadata > {} labels
io.k8s.api.apps.v1.Deployment (v1@deployment.json)

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: blue-deployment
  name: blue-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: blue-deployment
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: blue-deployment
    spec:
      containers:
      - image: nginx
        name: nginx
        resources: {}
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: color
                operator: In
                values:
                - blue
status: {}

```

12- How many `DaemonSets` are created in the cluster in all namespaces?
Only 1 Deamonset on kube-system name space named kube-proxy.

13- what DaemonSets exist on the kube-system namespace?
Only 1 Deamonset on kube-system name space named kube-proxy.

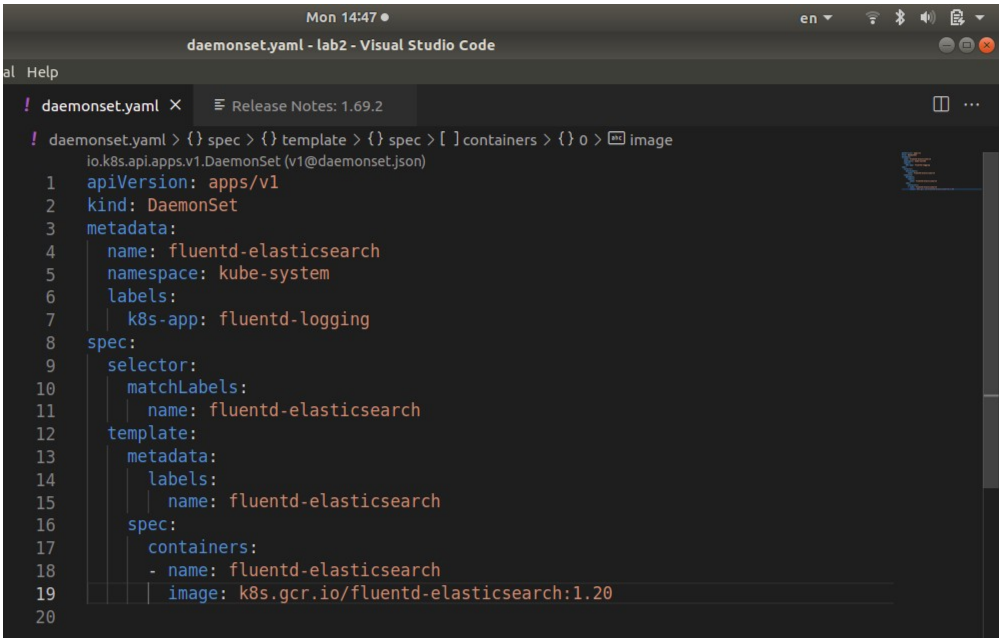
14- What is the image used by the POD deployed by the `kube-proxy` DaemonSet.
k8s.gcr.io/kube-proxy:v1.24.1

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

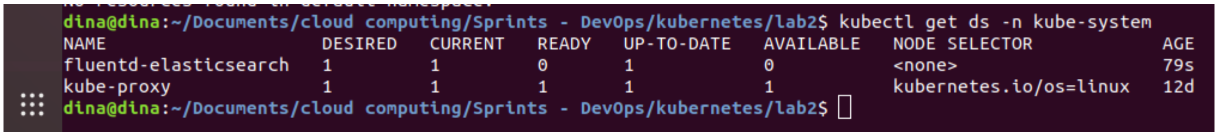
Name: elasticsearch

Namespace: kube-system

Image: k8s.gcr.io/fluentd-elasticsearch:1.20

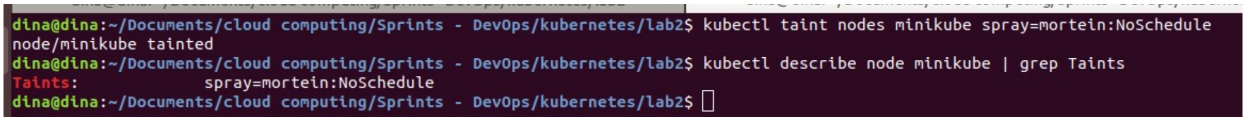

Mon 14:47 ●
en
daemonset.yaml - lab2 - Visual Studio Code
al Help
! daemonset.yaml ×
Release Notes: 1.69.2
! daemonset.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > image
io.k8s.api.apps.v1.DaemonSet (v1@daemonset.json)

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
      - name: fluentd-elasticsearch
        image: k8s.gcr.io/fluentd-elasticsearch:1.20

```



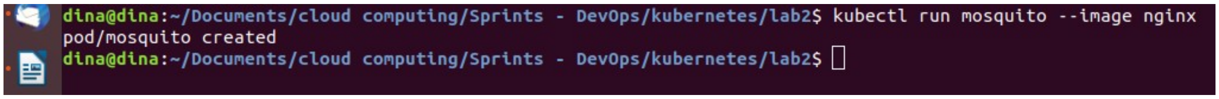
```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get ds -n kube-system
NAME                    DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
fluentd-elasticsearch   1         1         0       1            0           <none>                   79s
kube-proxy              1         1         1       1            1           kubernetes.io/os=linux   12d
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

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



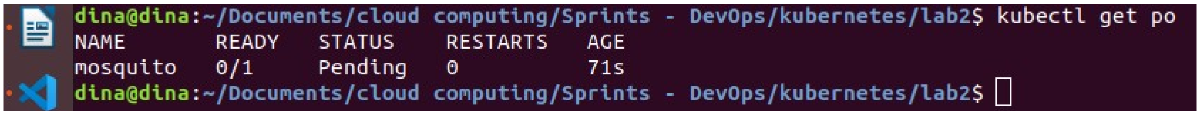
```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl taint nodes minikube spray=mortein:NoSchedule
node/minikube tainted
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl describe node minikube | grep Taints
Taints:             spray=mortein:NoSchedule
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

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



```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl run mosquito --image nginx
pod/mosquito created
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

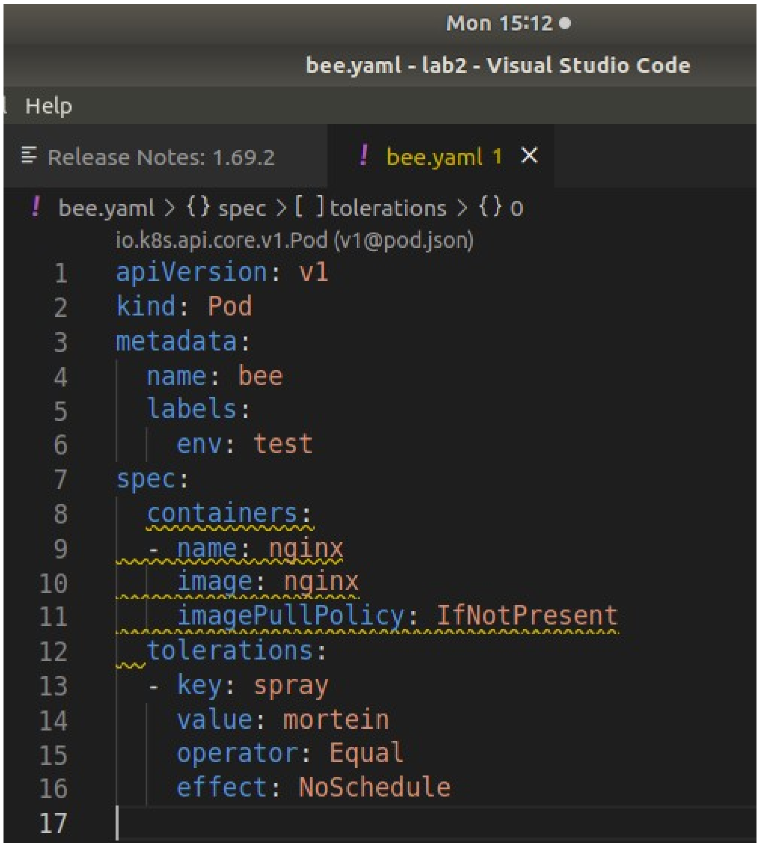
18- What is the state of the `mosquito` POD?



```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get po
NAME       READY   STATUS    RESTARTS   AGE
mosquito   0/1     Pending   0          71s
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

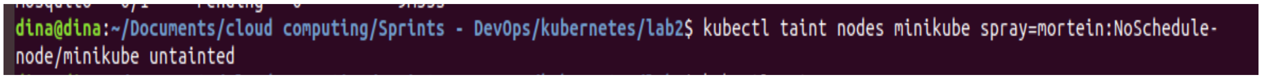
19- Create another pod named `bee` with the NGINX image, which has a toleration set to the taint `Mortein`

- Image name: nginx
- Key: spray
- Value: mortein
- Effect: NoSchedule
- Status: Running


Mon 15:12 ●
bee.yaml - lab2 - Visual Studio Code
Help
Release Notes: 1.69.2
! bee.yaml 1 ×
! bee.yaml > {} spec > [ ] tolerations > {} 0
io.k8s.api.core.v1.Pod (v1@pod.json)

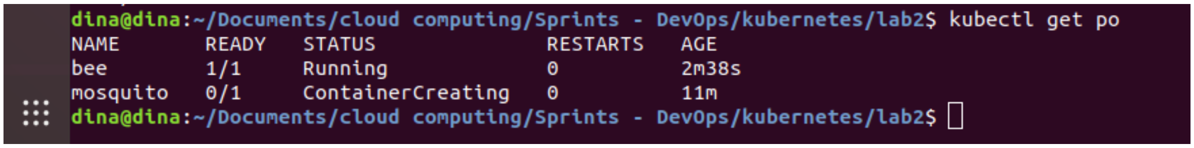
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: bee
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  tolerations:
  - key: spray
    value: mortein
    operator: Equal
    effect: NoSchedule

```

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



```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl taint nodes minikube spray=mortein:NoSchedule-
node/minikube untainted
```

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



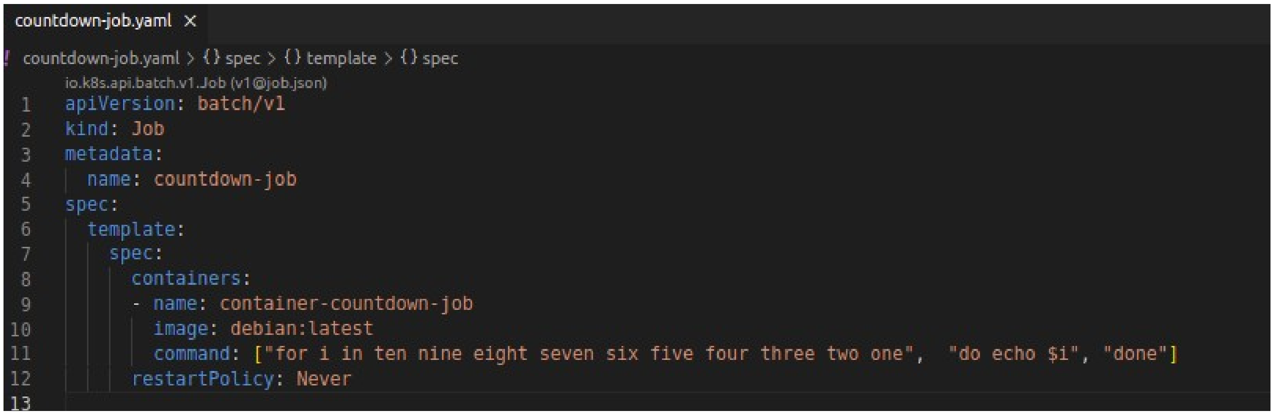
```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl get po
NAME       READY   STATUS              RESTARTS   AGE
bee        1/1     Running             0          2m38s
mosquito   0/1     ContainerCreating   0          11m
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```

22- Create a job countdown-job.

The container should be named as container-countdown-job

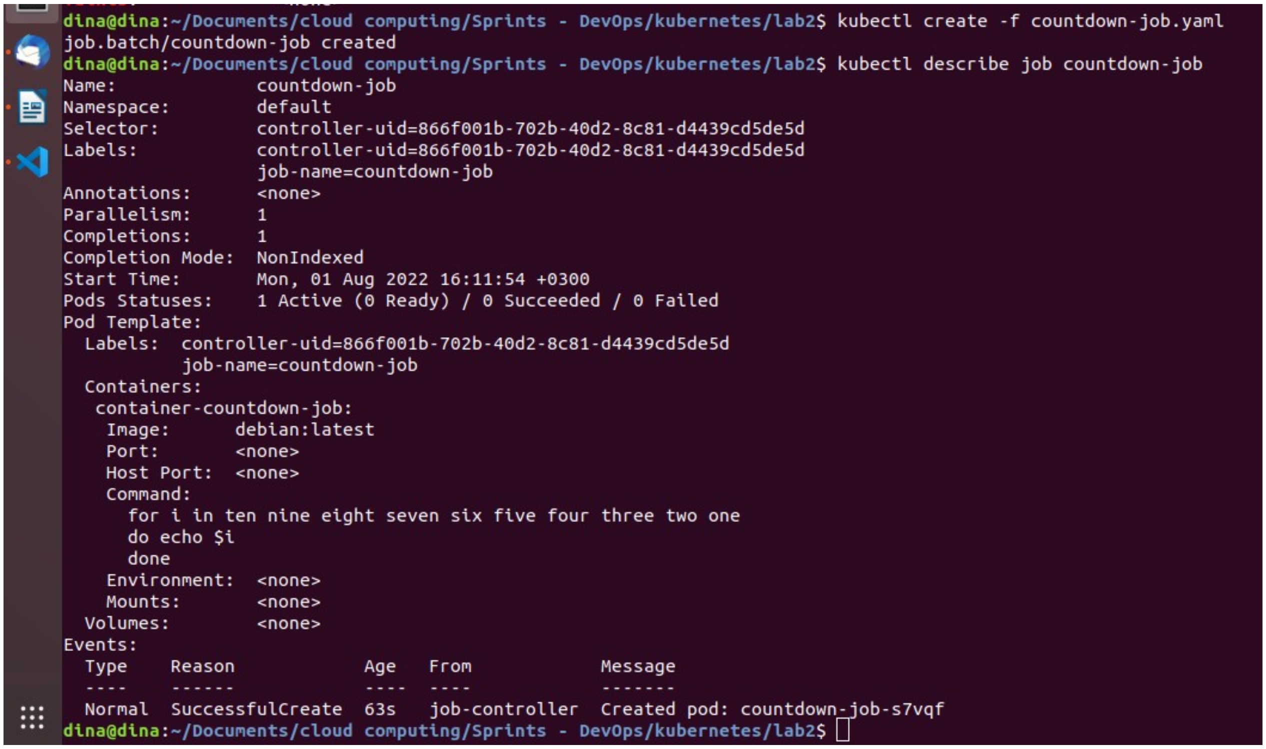
Use image debian:latest, and restart policy should be Never.

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



countdown-job.yaml ×
countdown-job.yaml > {} spec > {} template > {} spec
io.k8s.api.batch.v1.Job (v1@job.json)

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: countdown-job
spec:
  template:
    spec:
      containers:
      - name: container-countdown-job
        image: debian:latest
        command: ["for i in ten nine eight seven six five four three two one",  "do echo $i", "done"]
      restartPolicy: Never

```



```
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl create -f countdown-job.yaml
job.batch/countdown-job created
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ kubectl describe job countdown-job
Name:             countdown-job
Namespace:        default
Selector:         controller-uid=866f001b-702b-40d2-8c81-d4439cd5de5d
Labels:           controller-uid=866f001b-702b-40d2-8c81-d4439cd5de5d
                  job-name=countdown-job
Annotations:      <none>
Parallelism:      1
Completions:      1
Completion Mode:  NonIndexed
Start Time:       Mon, 01 Aug 2022 16:11:54 +0300
Pods Statuses:    1 Active (0 Ready) / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  controller-uid=866f001b-702b-40d2-8c81-d4439cd5de5d
           job-name=countdown-job
  Containers:
   container-countdown-job:
    Image:      debian:latest
    Port:       <none>
    Host Port:  <none>
    Command:
      for i in ten nine eight seven six five four three two one
      do echo $i
      done
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age   From            Message
  ----    ------            ----  ----            -------
  Normal  SuccessfulCreate  63s   job-controller  Created pod: countdown-job-s7vqf
dina@dina:~/Documents/cloud computing/Sprints - DevOps/kubernetes/lab2$ 
```