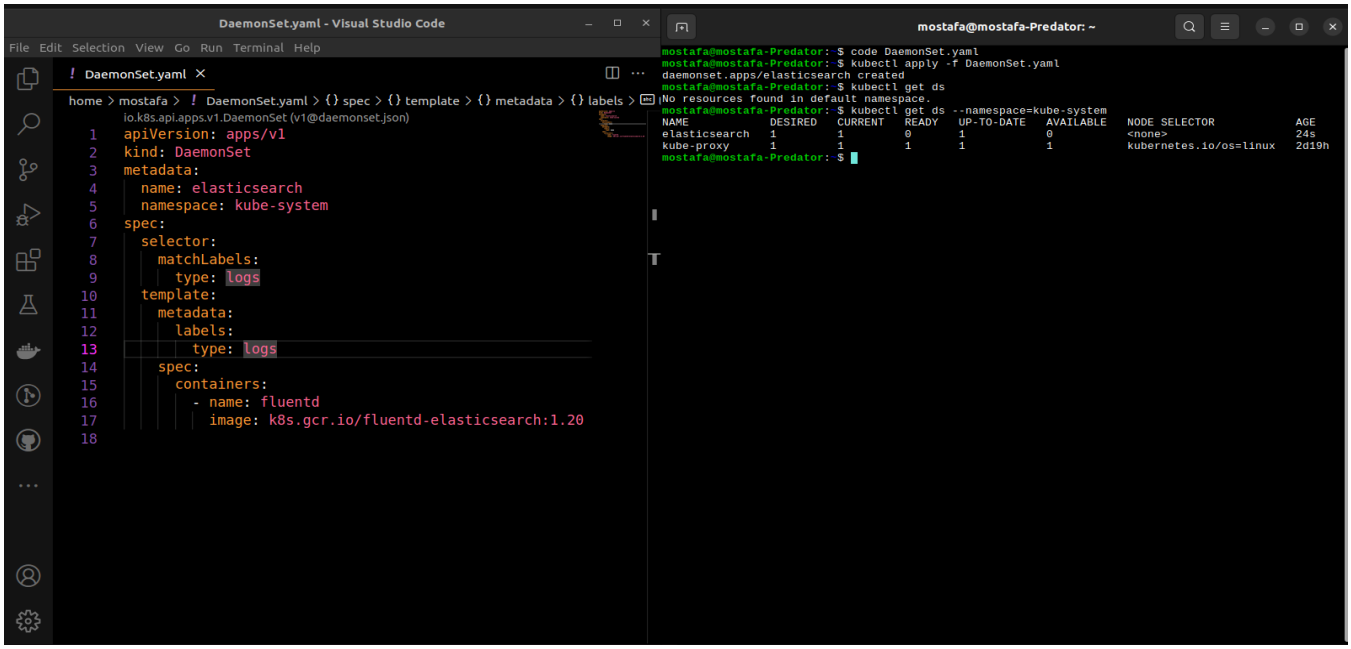
Note: Extract url of service

minikube service we-app-service --url

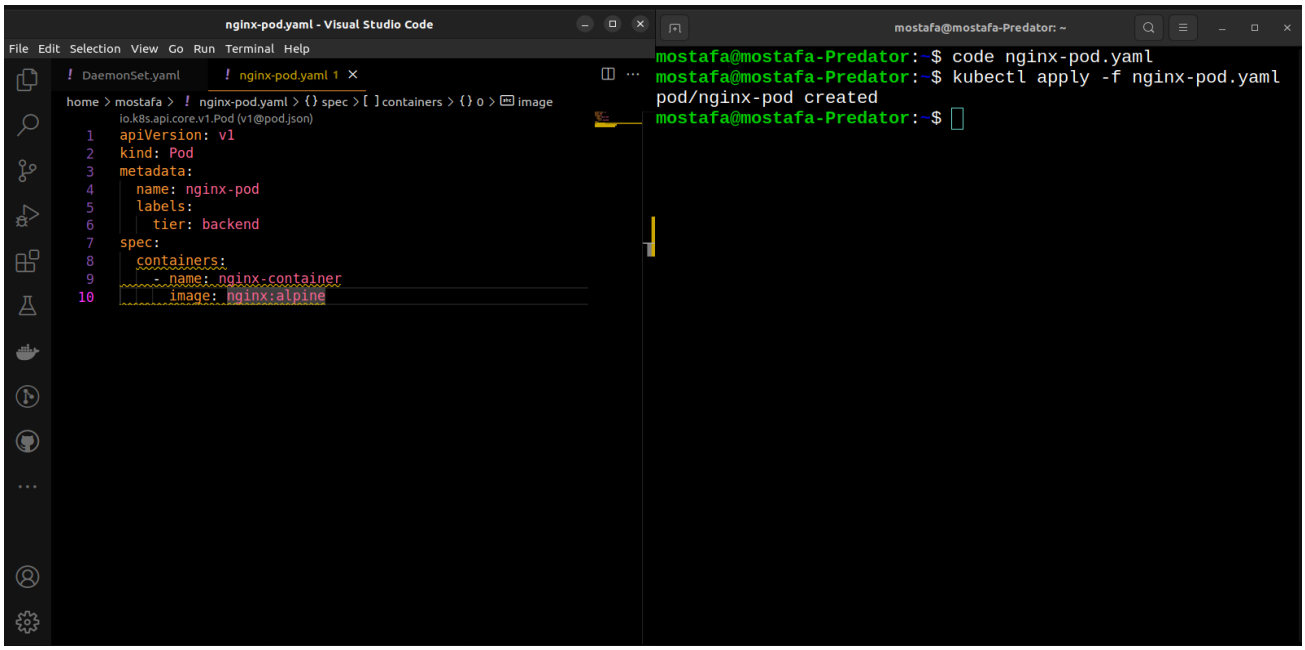
Q1,2,3

```
mostafa@mostafa-Predator:~$ kubectl get ds -A
NAMESPACE     NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   2d19h
mostafa@mostafa-Predator:~$ kubectl get ds --namespace=kube-system
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   2d19h
mostafa@mostafa-Predator:~$ kubectl describe ds -A | grep image -i
    Image:      registry.k8s.io/kube-proxy:v1.26.3
mostafa@mostafa-Predator:~$
```

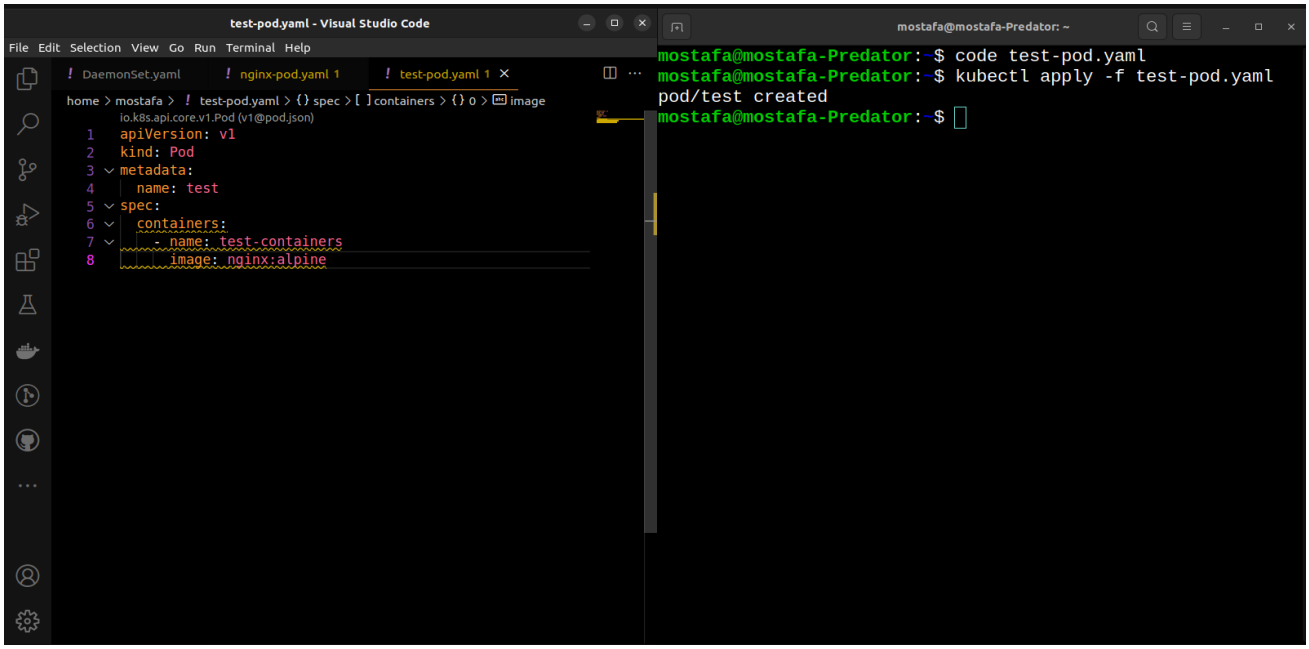
Q4



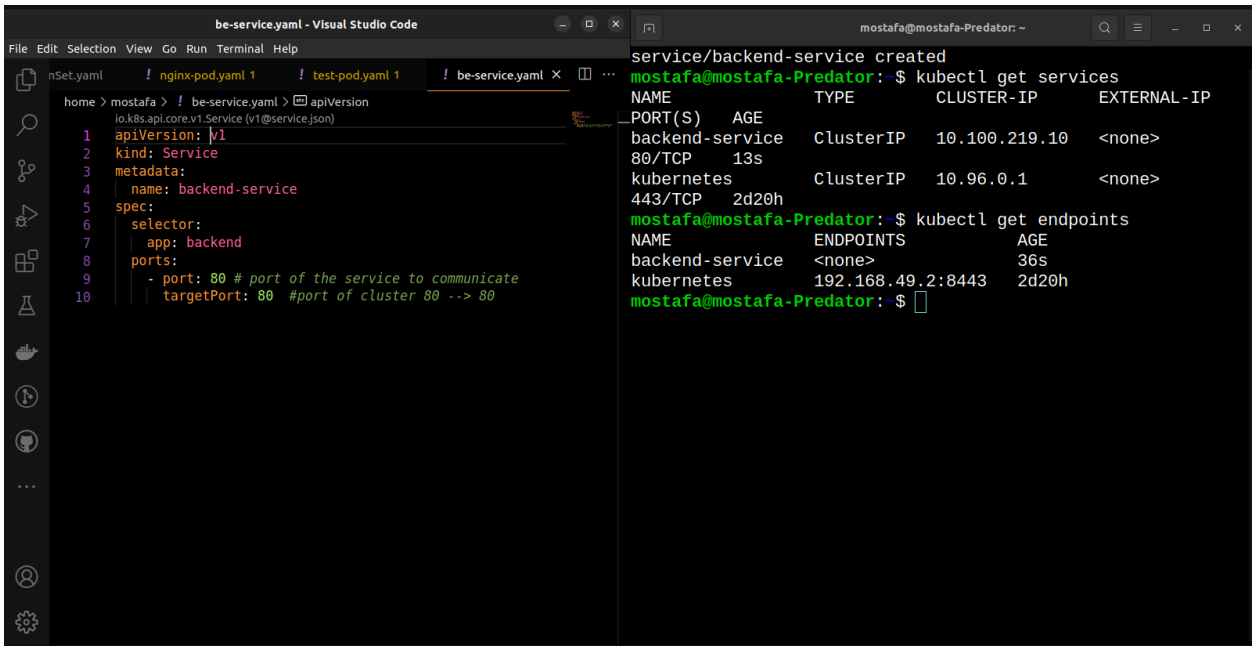
Q5


nginx-pod.yaml - Visual Studio Code
File Edit Selection View Go Run Terminal Help
DaemonSet.yaml
nginx-pod.yaml 1
home > mostafa > nginx-pod.yaml > {} spec > [ ] containers > {} 0 > image
io.k8s.api.core.v1.Pod (v1@pod.json)
apiVersion: v1
kind: Pod
metadata:
name: nginx-pod
labels:
tier: backend
spec:
containers:
- name: nginx-container
image: nginx:alpine
mostafa@mostafa-Predator: ~
mostafa@mostafa-Predator:~$ code nginx-pod.yaml
mostafa@mostafa-Predator:~$ kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
mostafa@mostafa-Predator:~$

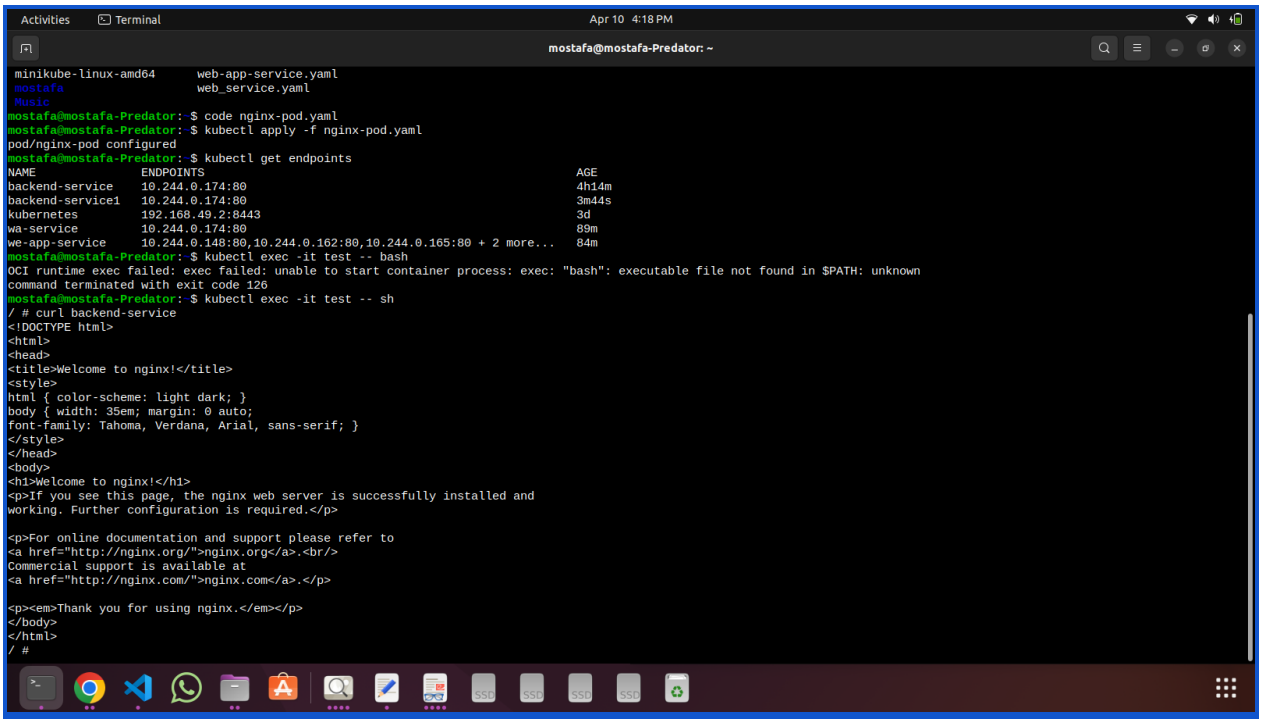
Q6


test-pod.yaml - Visual Studio Code
File Edit Selection View Go Run Terminal Help
DaemonSet.yaml
nginx-pod.yaml 1
test-pod.yaml 1
home > mostafa > test-pod.yaml > {} spec > [ ] containers > {} 0 > image
io.k8s.api.core.v1.Pod (v1@pod.json)
apiVersion: v1
kind: Pod
metadata:
name: test
spec:
containers:
- name: test-containers
image: nginx:alpine
mostafa@mostafa-Predator: ~
mostafa@mostafa-Predator:~$ code test-pod.yaml
mostafa@mostafa-Predator:~$ kubectl apply -f test-pod.yaml
pod/test created
mostafa@mostafa-Predator:~$

Q7


be-service.yaml - Visual Studio Code
File Edit Selection View Go Run Terminal Help
nginx-pod.yaml 1
test-pod.yaml 1
be-service.yaml
home > mostafa > be-service.yaml > apiVersion
io.k8s.api.core.v1.Service (v1@service.json)
1 apiVersion: v1
2 kind: Service
3 metadata:
4 name: backend-service
5 spec:
6 selector:
7 app: backend
8 ports:
9 - port: 80 # port of the service to communicate
10 targetPort: 80 #port of cluster 80 --> 80
mostafa@mostafa-Predator: ~
service/backend-service created
mostafa@mostafa-Predator:~$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP
PORT(S) AGE
backend-service ClusterIP 10.100.219.10 <none>
80/TCP 13s
kubernetes ClusterIP 10.96.0.1 <none>
443/TCP 2d20h
mostafa@mostafa-Predator:~$ kubectl get endpoints
NAME ENDPOINTS AGE
backend-service <none> 36s
kubernetes 192.168.49.2:8443 2d20h
mostafa@mostafa-Predator:~$

Q8

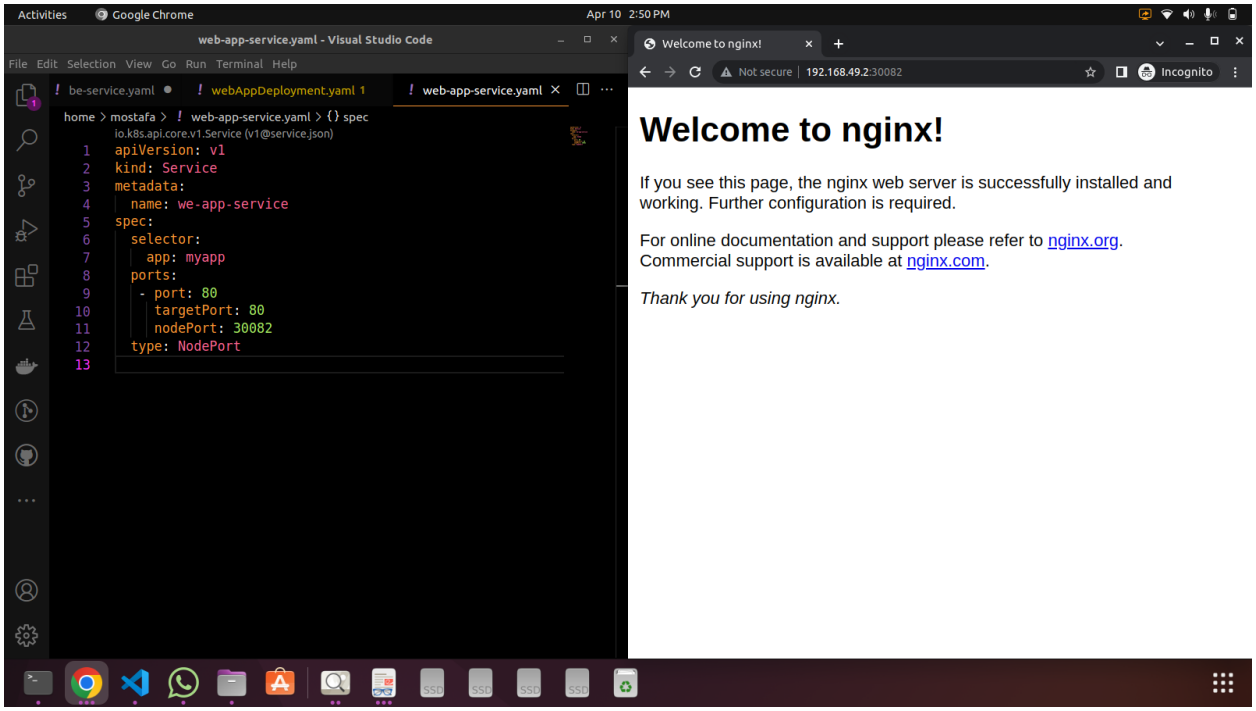

Activities
Terminal
Apr 10 4:18 PM
mostafa@mostafa-Predator: ~
minikube-linux-amd64 web-app-service.yaml
mostafa web_service.yaml
Music
mostafa@mostafa-Predator:~$ code nginx-pod.yaml
mostafa@mostafa-Predator:~$ kubectl apply -f nginx-pod.yaml
pod/nginx-pod configured
mostafa@mostafa-Predator:~$ kubectl get endpoints
NAME ENDPOINTS AGE
backend-service 10.244.0.174:80 4h14m
backend-service1 10.244.0.174:80 3m44s
kubernetes 192.168.49.2:8443 3d
wa-service 10.244.0.174:80 89m
we-app-service 10.244.0.148:80,10.244.0.162:80,10.244.0.165:80 + 2 more... 84m
mostafa@mostafa-Predator:~$ kubectl exec -it test -- bash
OCI runtime exec failed: exec failed: unable to start container process: exec: "bash": executable file not found in $PATH: unknown
command terminated with exit code 126
mostafa@mostafa-Predator:~$ kubectl exec -it test -- sh
/ # curl backend-service
<!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>
If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.
For online documentation and support please refer to
nginx.org.
Commercial support is available at
nginx.com.
Thank you for using nginx.
</body>
</html>
/ #

Q9

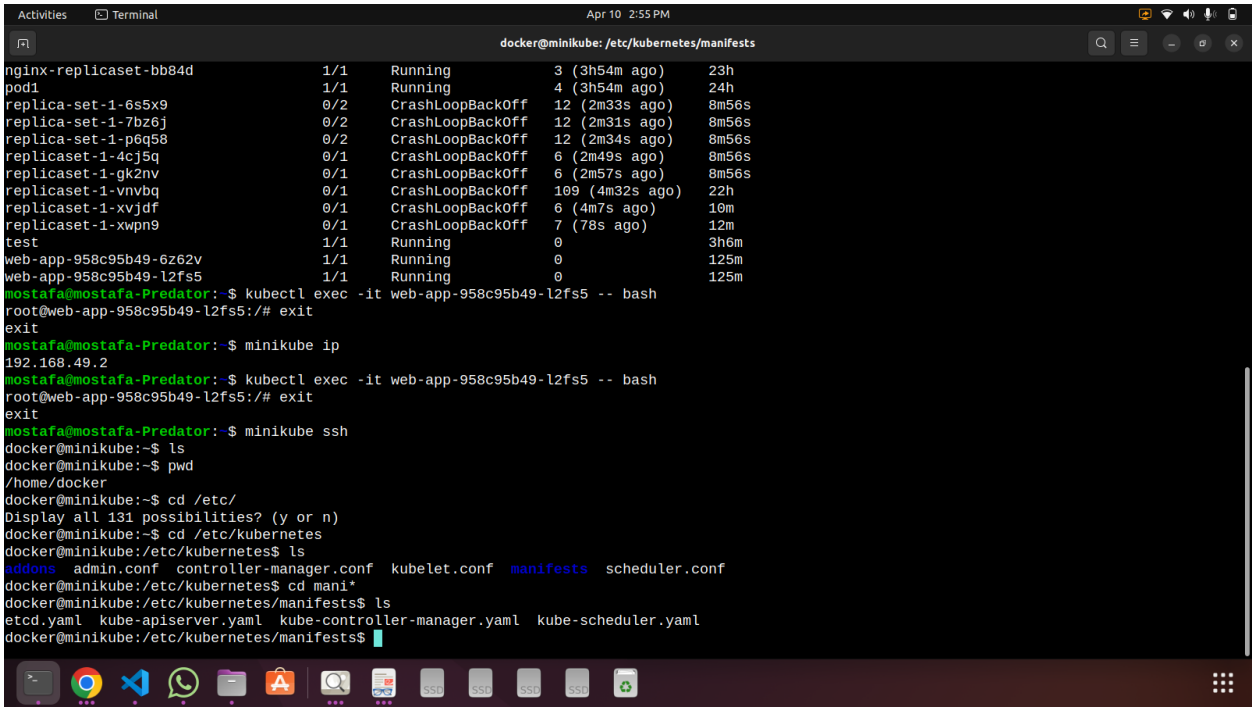

io.k8s.api.apps.v1.Deployment (v1@deployment.json)
apiVersion: apps/v1
kind: Deployment
metadata:
name: web-app
spec:
replicas: 2
selector:
matchLabels:
app: myapp
template:
metadata:
labels:
app: myapp
spec:
containers:
- name: nginx-cont
image: nginx

Q10

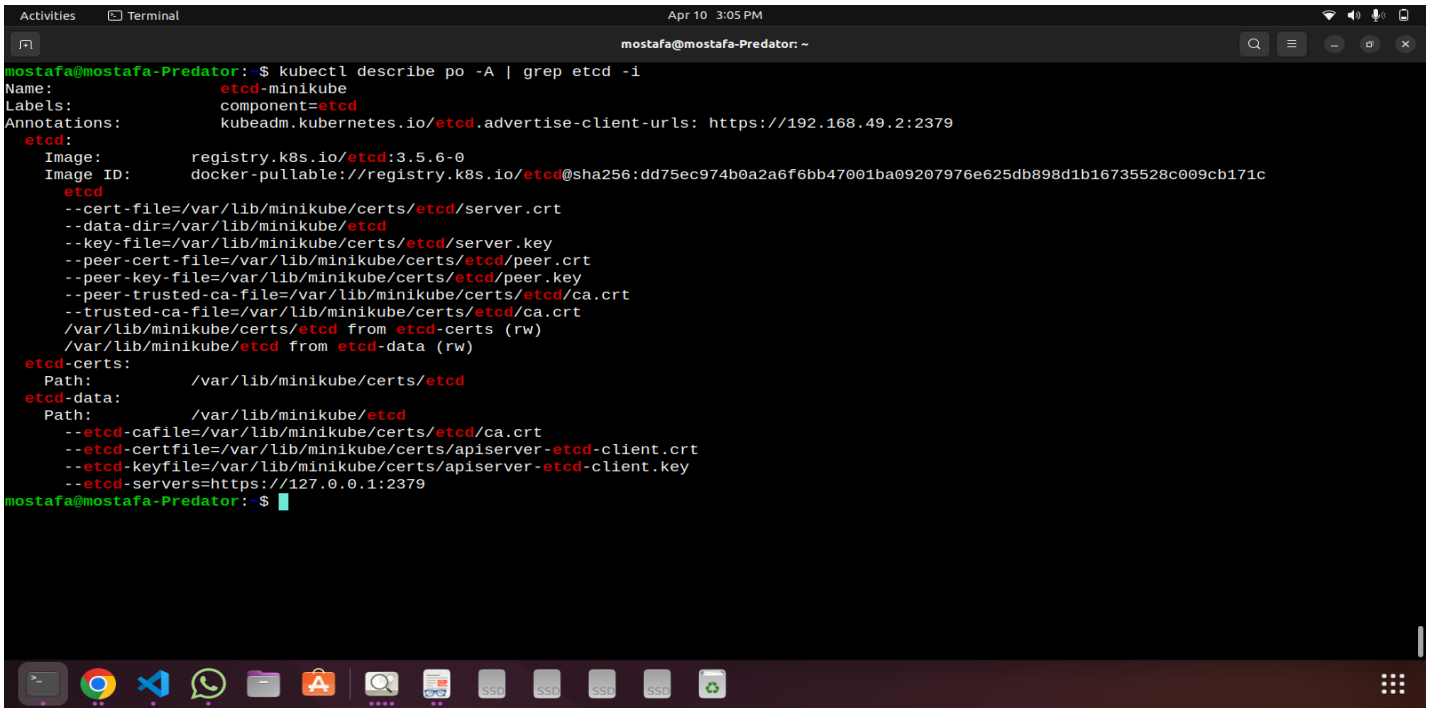
Q11


Activities
Google Chrome
Apr 10 2:50 PM
web-app-service.yaml - Visual Studio Code
File Edit Selection View Go Run Terminal Help
be-service.yaml
webAppDeployment.yaml 1
web-app-service.yaml
home > mostafa > web-app-service.yaml > {} spec
io.k8s.api.core.v1.Service (v1@service.json)
apiVersion: v1
kind: Service
metadata:
name: we-app-service
spec:
selector:
app: myapp
ports:
- port: 80
targetPort: 80
nodePort: 30082
type: NodePort
Welcome to nginx!
Not secure | 192.168.49.2:30082
Incognito
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.

Q12


Activities
Terminal
Apr 10 2:55 PM
docker@minikube: /etc/kubernetes/manifests
nginx-replicaset-bb84d 1/1 Running 3 (3h54m ago) 23h
pod1 1/1 Running 4 (3h54m ago) 24h
replica-set-1-6s5x9 0/2 CrashLoopBackOff 12 (2m33s ago) 8m56s
replica-set-1-7bz6j 0/2 CrashLoopBackOff 12 (2m31s ago) 8m56s
replica-set-1-p6q58 0/2 CrashLoopBackOff 12 (2m34s ago) 8m56s
replicaset-1-4cj5q 0/1 CrashLoopBackOff 6 (2m49s ago) 8m56s
replicaset-1-gk2nv 0/1 CrashLoopBackOff 6 (2m57s ago) 8m56s
replicaset-1-vnvbq 0/1 CrashLoopBackOff 109 (4m32s ago) 22h
replicaset-1-xvjdf 0/1 CrashLoopBackOff 6 (4m7s ago) 10m
replicaset-1-xwpn9 0/1 CrashLoopBackOff 7 (78s ago) 12m
test 1/1 Running 0 3h6m
web-app-958c95b49-6z62v 1/1 Running 0 125m
web-app-958c95b49-l2fs5 1/1 Running 0 125m
mostafa@mostafa-Predator:~$ kubectl exec -it web-app-958c95b49-l2fs5 -- bash
root@web-app-958c95b49-l2fs5:/# exit
exit
mostafa@mostafa-Predator:~$ minikube ip
192.168.49.2
mostafa@mostafa-Predator:~$ kubectl exec -it web-app-958c95b49-l2fs5 -- bash
root@web-app-958c95b49-l2fs5:/# exit
exit
mostafa@mostafa-Predator:~$ minikube ssh
docker@minikube:~$ ls
docker@minikube:~$ pwd
/home/docker
docker@minikube:~$ cd /etc/
Display all 131 possibilities? (y or n)
docker@minikube:~$ cd /etc/kubernetes
docker@minikube:/etc/kubernetes$ ls
addons admin.conf controller-manager.conf kubelet.conf manifests scheduler.conf
docker@minikube:/etc/kubernetes$ cd mani*
docker@minikube:/etc/kubernetes/manifests$ ls
etcd.yaml kube-apiserver.yaml kube-controller-manager.yaml kube-scheduler.yaml
docker@minikube:/etc/kubernetes/manifests$

Q13


Activities
Terminal
Apr 10 3:05 PM
mostafa@mostafa-Predator: ~
mostafa@mostafa-Predator:~$ kubectl describe po -A | grep etcd -i
Name: etcd-minikube
Labels: component=etcd
Annotations: kubeadm.kubernetes.io/etcd.advertise-client-urls: https://192.168.49.2:2379
etcd:
Image: registry.k8s.io/etcd:3.5.6-0
Image ID: docker-pullable://registry.k8s.io/etcd@sha256:dd75ec974b0a2a6f6bb47001ba09207976e625db898d1b16735528c009cb171c
etcd
--cert-file=/var/lib/minikube/certs/etcd/server.crt
--data-dir=/var/lib/minikube/etcd
--key-file=/var/lib/minikube/certs/etcd/server.key
--peer-cert-file=/var/lib/minikube/certs/etcd/peer.crt
--peer-key-file=/var/lib/minikube/certs/etcd/peer.key
--peer-trusted-ca-file=/var/lib/minikube/certs/etcd/ca.crt
--trusted-ca-file=/var/lib/minikube/certs/etcd/ca.crt
/var/lib/minikube/certs/etcd from etcd-certs (rw)
/var/lib/minikube/etcd from etcd-data (rw)
etcd-certs:
Path: /var/lib/minikube/certs/etcd
etcd-data:
Path: /var/lib/minikube/etcd
--etcd-cafile=/var/lib/minikube/certs/etcd/ca.crt
--etcd-certfile=/var/lib/minikube/certs/apiserver-etcd-client.crt
--etcd-keyfile=/var/lib/minikube/certs/apiserver-etcd-client.key
--etcd-servers=https://127.0.0.1:2379
mostafa@mostafa-Predator:~$