

KUBERNETES – EXERCISES

- 1) Create a pod called web-server with nginx:1.19.10 image

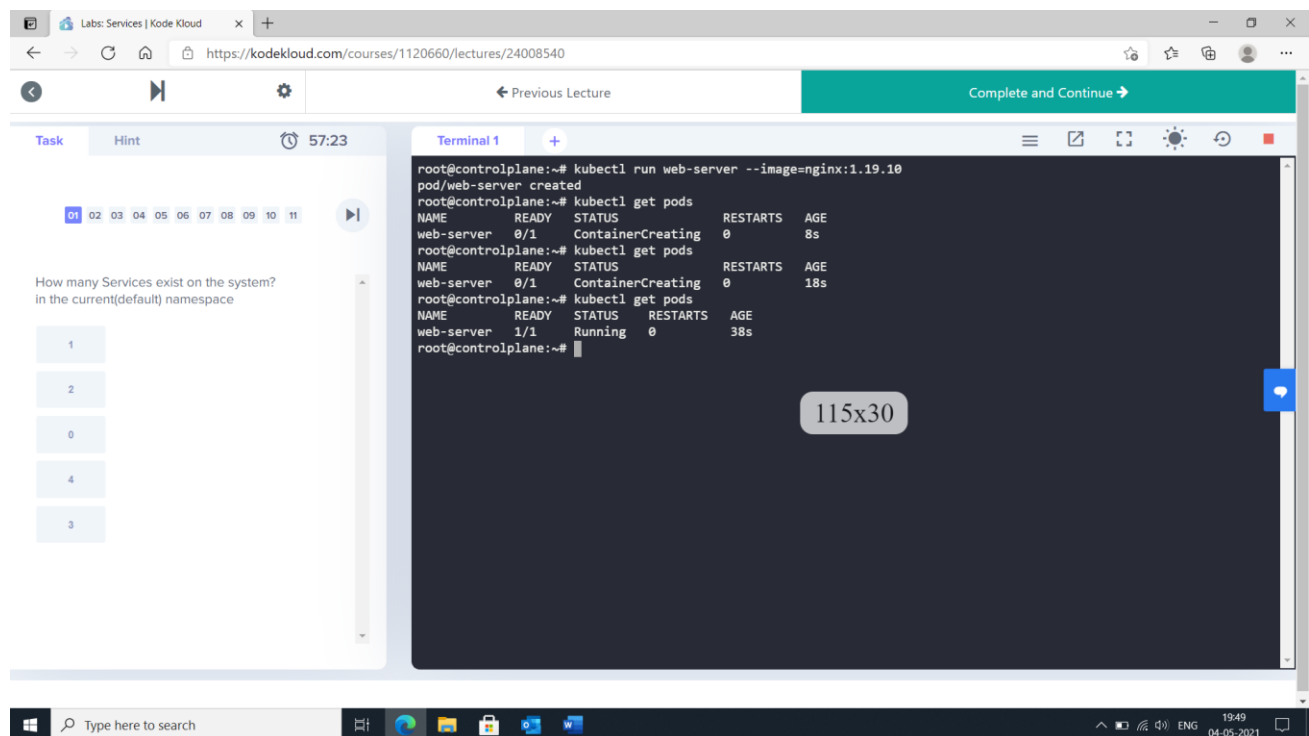
```
kubectl run web-server - --image=nginx:1.19.10
```

```
kubectl get pods
```

(or)

pod-def.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: webserver
spec:
  containers:
  - name: nginx
    image: nginx:1.19.10
```



2) Expose port 80 of the web-server pod to be reachable within cluster

```
kubectl expose pod web-server - --type=ClusterIP - -port=80
```

Task: How many Services exist on the system? in the current(default) namespace

Options: 1, 2, 0, 4, 3

Terminal 1:

```
root@controlplane:~# kubectl run web-server --image=nginx:1.19.10
pod/web-server created
root@controlplane:~# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
web-server    0/1     ContainerCreating   0           8s
root@controlplane:~# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
web-server    0/1     ContainerCreating   0          18s
root@controlplane:~# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
web-server    1/1     Running    0          38s
root@controlplane:~# kubectl expose pod web-server --type=ClusterIP --port=80
service/web-server exposed
root@controlplane:~# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
web-server    1/1     Running    0         2m12s
root@controlplane:~# kubectl describe pods
```

Task: How many Services exist on the system? in the current(default) namespace

Options: 1, 2, 0, 4, 3

Terminal 1:

```
root@controlplane:~# kubectl describe pods
Name:          web-server
Namespace:     default
Priority:       0
Node:          controlplane/10.231.40.6
Start Time:    Tue, 04 May 2021 14:19:03 +0000
Labels:        run=web-server
Annotations:    <none>
Status:        Running
IP:            10.244.0.4
IPs:           IP: 10.244.0.4
Containers:
  web-server:
    Container ID:  docker://8ba8496eb5b0363201b53435c6f9642e4d025833fa1519880dfdb3e4acea8877
    Image:          nginx:1.19.10
    Image ID:       docker-pullable://nginx@sha256:75a55d33ecc73c2a242450a9f1cc858499d468f077ea942867e662c247b5e412
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Tue, 04 May 2021 14:19:32 +0000
    Ready:          True
    Restart Count:   0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-rkjs7 (ro)
Conditions:
  Type            Status
  Initialized      True
  Ready            True
```

The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue" buttons. Below the navigation bar, there is a "Task" section with a "Hint" button and a timer showing 55:24. The task question is "How many Services exist on the system? in the current(default) namespace". There are five radio button options: 1, 2, 0, 4, and 3. To the right of the task is a "Terminal 1" window showing the output of a `kubectl get svc` command. The terminal output shows the following details for the 'web-server' service:

```
Started: Tue, 04 May 2021 14:19:32 +0000
Ready: True
Restart Count: 0
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from default-token-rkjs7 (ro)
Conditions:
  Type      Status
  Initialized   True
  Ready        True
  ContainersReady True
  PodScheduled  True
Volumes:
  default-token-rkjs7:
    Type: Secret (a volume populated by a Secret)
    SecretName: default-token-rkjs7
    Optional: false
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   2m29s default-scheduler Successfully assigned default/web-server to controlplane
  Normal    Pulling     2m26s kubelet        Pulling image "nginx:1.19.10"
  Normal    Pulled      2m    kubelet        Successfully pulled image "nginx:1.19.10" in 26.743199304s
  Normal    Created     119s  kubelet        Created container web-server
  Normal    Started     119s  kubelet        Started container web-server
root@controlplane:~#
```

The screenshot shows the same web browser window as above, but the timer now shows 54:30. The task question remains the same. The "Terminal 1" window now shows the output of a `kubectl describe svc` command for the 'web-server' service. The terminal output shows the following details:

```
root@controlplane:~# kubectl get svc
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1     <none>        443/TCP    8m55s
web-server   ClusterIP   10.105.17.59  <none>        80/TCP     75s
root@controlplane:~# kubectl describe svc
Name:         web-server
Namespace:    default
Labels:       component=apiserver,provider=kubernetes
Annotations:   <none>
Selector:     <none>
Type:         ClusterIP
IP Families:  <none>
IP:           10.96.0.1
IPs:          10.96.0.1
Port:         https 443/TCP
TargetPort:   6443/TCP
Endpoints:    10.231.40.6:6443
Session Affinity: None
Events:       <none>

Name:         web-server
Namespace:    default
Labels:       run=web-server
Annotations:   <none>
Selector:     run=web-server
Type:         ClusterIP
IP Families:  <none>
IP:           10.105.17.59
```

The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue" buttons. Below the navigation bar, there is a "Task" section with a "Hint" tab and a timer showing 54:28. The task question is: "How many Services exist on the system? in the current(default) namespace". There are five radio button options: 1, 2, 0, 4, and 3. To the right of the task is a "Terminal 1" window showing the output of the command `kubectl get services`. The output lists two services: `api` and `web-server`.

```
Labels:      component=apiserver,provider=kubernetes
Annotations: <none>
Selector:    <none>
Type:        ClusterIP
IP Families: <none>
IP:          10.96.0.1
IPs:         10.96.0.1
Port:        https 443/TCP
TargetPort:  6443/TCP
Endpoints:   10.231.40.6:6443
Session Affinity: None
Events:      <none>

Name:        web-server
Namespace:   default
Labels:      run=web-server
Annotations: <none>
Selector:    run=web-server
Type:        ClusterIP
IP Families: <none>
IP:          10.105.17.59
IPs:         10.105.17.59
Port:        <unset> 80/TCP
TargetPort:  80/TCP
Endpoints:   10.244.0.4:80
Session Affinity: None
Events:      <none>
root@controlplane:~#
```

3) Create a single pod with below images:

i) nginx:1.19.10

ii) redis:6.2.2

pod-def.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - name: nginx
      image: nginx:1.19.10
      ports:
        containerPort: 3000
      imagePullPolicy: Always
    - name: redis
      image: nginx:6.2.2
      ports:
        containerPort: 3001
      imagePullPolicy: Always
```

The screenshot shows a web browser window with multiple tabs. The active tab is 'Kubernetes for the Absolute Beginner' from KodeKloud. The URL is <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with 'Previous Lecture' and 'Complete and Continue' buttons. Below the navigation bar, there is a 'Task' section with a 'Hint' button and a timer showing 39:36. The task asks: 'How many Services exist on the system? in the current(default) namespace'. There are five radio button options: 1, 0, 4, 3, and 2. To the right of the task is a 'Terminal 1' window showing a YAML file named 'pod-define.yaml'. The YAML content is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: myfirstpod
  labels:
    app: myfirstpod
spec:
  containers:
  - name: nginx
    image: nginx:1.19.10
    ports:
      - containerPort: 3000
  - name: redis
    image: redis:6.2.0
    ports:
      - containerPort: 3001
```

The terminal window also shows the file size '17L, 372C' and a character count '14,28' and 'A11'.

The screenshot shows the same web browser window as the previous one, but the terminal window now shows the output of the 'kubectl get pods' command. The output is as follows:

```
root@controlplane:~# vim pod-define.yaml
root@controlplane:~# kubectl apply -f pod-define.yaml
pod/myfirstpod unchanged
root@controlplane:~# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
myfirstpod    2/2     Running   0           14m
root@controlplane:~#
```

The task section remains the same, with the timer now showing 39:05.

4) Expose port 80 and 6379 of the above created pod such that the application can be connected from the outside world using node's IP address

pod-def.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: myfirstpod
  labels:
    app: myfirstpod
spec:
  containers:
    - name: nginx
      image: nginx:1.19.10
      ports:
        - containerPort: 3000
      imagePullPolicy: Always
    - name: redis
      image: nginx:6.2.2
      ports:
        - containerPort: 3001
      imagePullPolicy: Always
```


pod-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: myfirstapp-service
  labels:
    name: myservice
spec:
  type: NodePort
  selector:
    name: myfirstpod
  ports:
    - name: nginx
      protocol: TCP
      port: 80
      targetPort: 3000
      nodePort: 30000
    - name: redis
      protocol: TCP
      port: 6379
      targetPort: 3001
      nodePort: 30001
```

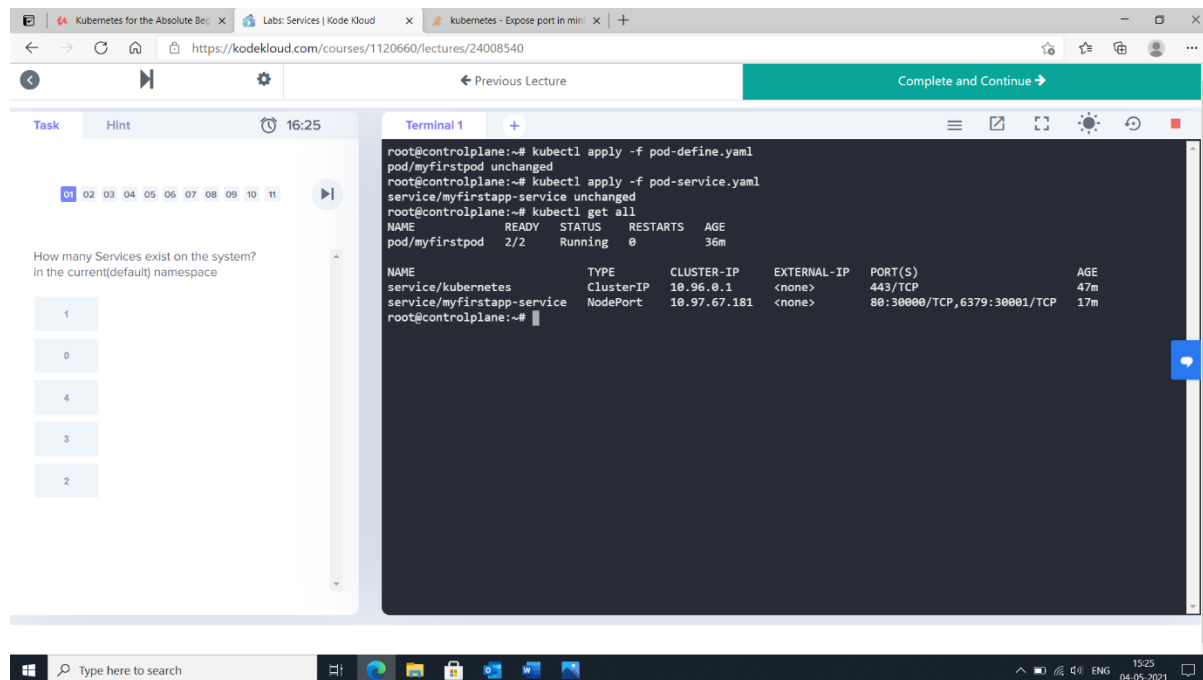
The screenshot shows a web browser window with multiple tabs. The active tab is titled "Kubernetes for the Absolute Beginner" and displays a lecture page from KodeKloud. The URL is <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue" buttons. Below the navigation bar, there is a "Task" section with a hint and a timer showing 33:28. The task asks: "How many Services exist on the system? in the current(default) namespace". There are five radio button options: 1, 0, 4, 3, and 2. To the right of the task is a "Terminal 1" window showing a Kubernetes manifest file named "pod-service.yaml". The manifest defines a Service of type NodePort with two ports: nginx (port 80, targetPort 3000, nodePort 30000) and redis (port 6379, targetPort 3001, nodePort 30001). The terminal output shows the file content and its size (22L, 534C).

```
apiVersion: v1
kind: Service
metadata:
  name: myfirstapp-service
  labels:
    name: myservice
spec:
  type: NodePort
  selector:
    name: myfirstpod
  ports:
    - name: nginx
      protocol: TCP
      port: 80
      targetPort: 3000
      nodePort: 30000
    - name: redis
      protocol: TCP
      port: 6379
      targetPort: 3001
      nodePort: 30001
```

"pod-service.yaml" 22L, 534C

The screenshot shows the same web browser window as the previous one, but the terminal window now displays the output of several Kubernetes commands. The commands and their outputs are as follows:

```
root@controlplane:~# vim pod-service.yaml
root@controlplane:~# kubectl apply -f pod-service.yaml
service/myfirstapp-service created
root@controlplane:~# kubectl get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes           ClusterIP   10.96.0.1      <none>          443/TCP          30m
myfirstapp-service   NodePort    10.97.67.181   <none>          80:30000/TCP,6379:30001/TCP  12s
root@controlplane:~# kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
myfirstpod  2/2     Running   0          19m
root@controlplane:~#
```



```
root@controlplane:~# kubectl apply -f pod-define.yaml
pod/myfirstpod unchanged
root@controlplane:~# kubectl apply -f pod-service.yaml
service/myfirstapp-service unchanged
root@controlplane:~# kubectl get all
NAME                READY   STATUS    RESTARTS   AGE
pod/myfirstpod       2/2     Running   0           36m

NAME                                TYPE               CLUSTER-IP   EXTERNAL-IP   PORT(S)                  AGE
service/kubernetes              ClusterIP          10.96.0.1    <none>         443/TCP                  47m
service/myfirstapp-service       NodePort           10.97.67.181 <none>         80:30000/TCP,6379:30001/TCP 17m
root@controlplane:~#
```

5) Create a deployment web-deploy with nginx:1.19.10 image of 2 replica

deploy-def.yaml

```
apiVersion: apps/v1
kind: Deployment
  metadata:
    name: myfirstdeploy
    labels:
      app: myfirst-app
  spec:
    replicas: 2
    selector:
      matchLabels:
        app: myfirst-app
    template:
      metadata:
        labels:
          app: myfirst-app
      containers:
        - name: nginx
          image: nginx:1.19.10
          ports:
            - containerPort: 80
```

Task: Hint 32:09

How many Services exist on the system? in the current(default) namespace

3
1
0
4
2

```
root@controlplane:~# kubectl create -f deploy-def.yaml
deployment.apps/myfirstdeploy created
root@controlplane:~# kubectl get deploy -l app=myfirst-app
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
myfirstdeploy  2/2     2            2           38s
root@controlplane:~# kubectl get rs -l app=myfirst-app
NAME                               DESIRED   CURRENT   READY   AGE
myfirstdeploy-68d796bb8c           2         2         2       54s
root@controlplane:~# kubectl get pod -l app=myfirst-app
NAME                               READY   STATUS    RESTARTS   AGE
myfirstdeploy-68d796bb8c-fj9tt     1/1     Running   0           78s
myfirstdeploy-68d796bb8c-m2l4w     1/1     Running   0           78s
root@controlplane:~#
```

Task: Hint 31:17

How many Services exist on the system? in the current(default) namespace

3
1
0
4
2

```
myfirstdeploy-68d796bb8c 2 2 2 54s
root@controlplane:~# kubectl get pod -l app=myfirst-app
NAME                               READY   STATUS    RESTARTS   AGE
myfirstdeploy-68d796bb8c-fj9tt     1/1     Running   0           78s
myfirstdeploy-68d796bb8c-m2l4w     1/1     Running   0           78s
root@controlplane:~# kubectl describe deploy myfirstdeploy
Name:          myfirstdeploy
Namespace:     default
CreationTimestamp: Tue, 04 May 2021 11:51:44 +0000
Labels:        app=myfirst-app
Annotations:    deployment.kubernetes.io/revision: 1
Selector:      app=myfirst-app
Replicas:      2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=myfirst-app
  Containers:
    nginx:
      Image:      nginx:1.19.10
      Port:       80/TCP
      Host Port:  80/TCP
      Environment: <none>
      Mounts:      <none>
      Volumes:     <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available      True    MinimumReplicasAvailable
```

Task: How many Services exist on the system? in the current(default) namespace

01 02 03 04 05 06 07 08 09 10 11

3
1
0
4
2

```

CreationTimestamp: Tue, 04 May 2021 11:51:44 +0000
Labels:
  app=myfirst-app
Annotations:
  deployment.kubernetes.io/revision: 1
Selector:
  app=myfirst-app
Replicas:
  2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=myfirst-app
  Containers:
    nginx:
      Image: nginx:1.19.10
      Port: 80/TCP
      Host Port: 80/TCP
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available       True    MinimumReplicasAvailable
    Progressing     True    NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  myfirstdeploy-68d796bb8c (2/2 replicas created)
  Events:
    Type     Reason      Age   From              Message
    ----     -
    Normal   ScalingReplicaSet   2m12s   deployment-controller   Scaled up replica set myfirstdeploy-68d796bb8c to 2
  
```

6) Change the image of web-deploy to nginx:1.20.0 and record the change

kubectl set image deploy myfirstdeploy nginx=nginx:1.20.0 --record

Task: How many Services exist on the system? in the current(default) namespace

01 02 03 04 05 06 07 08 09 10 11

3
1
0
4
2

```

root@controlplane:~# kubectl rollout status deployment/myfirstdeploy
deployment "myfirstdeploy" successfully rolled out
root@controlplane:~# kubectl rollout history deployment/myfirstdeploy
deployment.apps/myfirstdeploy
REVISION  CHANGE-CAUSE
1         <none>
2         kubectl set image deploy myfirstdeploy nginx=nginx:1.20.0 --record=true
root@controlplane:~# kubectl scale deployment myfirstdeploy --replicas=5
deployment.apps/myfirstdeploy scaled
root@controlplane:~# kubectl get deploy -l app=myfirst-app
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
myfirstdeploy  5/5    5           5          19m
root@controlplane:~# kubectl get rs -l app=myfirst-app
NAME                                DESIRED  CURRENT  READY  AGE
myfirstdeploy-68d796bb8c             0        0        0      19m
myfirstdeploy-8c7f7d8bb              5        5        5      9m16s
root@controlplane:~# kubectl get pod -l app=myfirst-app
NAME                                READY  STATUS  RESTARTS  AGE
myfirstdeploy-8c7f7d8bb-7nm7b        1/1    Running  0          53s
myfirstdeploy-8c7f7d8bb-bps8s        1/1    Running  0          53s
myfirstdeploy-8c7f7d8bb-rqng8        1/1    Running  0          53s
myfirstdeploy-8c7f7d8bb-t5bm6        1/1    Running  0          9m13s
myfirstdeploy-8c7f7d8bb-vhjhz        1/1    Running  0          9m29s
  
```

7) Scale web-deploy to 5 replica

kubectl scale deployment myfirstdeploy - --replicas=5

The screenshot shows a web-based terminal environment for a Kubernetes deployment tutorial. On the left, there is a 'Task' panel with a hint: 'How many Services exist on the system? in the current(default) namespace'. Below the hint are five buttons labeled 0, 1, 2, 3, and 4. The main area is a terminal window titled 'Terminal 1' showing the following commands and output:

```

root@controlplane:~# kubectl rollout status deployment/myfirstdeploy
deployment "myfirstdeploy" successfully rolled out
root@controlplane:~# kubectl rollout history deployment/myfirstdeploy
deployment.apps/myfirstdeploy
REVISION  CHANGE-CAUSE
1          <none>
2          kubectl set image deploy myfirstdeploy nginx=nginx:1.20.0 --record=true

root@controlplane:~# kubectl scale deployment myfirstdeploy --replicas=5
deployment.apps/myfirstdeploy scaled
root@controlplane:~# kubectl get deploy -l app=myfirst-app
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
myfirstdeploy  5/5    5           5           19m
root@controlplane:~# kubectl get rs -l app=myfirst-app
NAME                                DESIRED  CURRENT  READY  AGE
myfirstdeploy-68d796bb8c             0         0        0      19m
myfirstdeploy-8c7f7d8bb              5         5        5      9m16s
root@controlplane:~# kubectl get pod -l app=myfirst-app
NAME                                READY  STATUS    RESTARTS  AGE
myfirstdeploy-8c7f7d8bb-7nm7b        1/1    Running   0          53s
myfirstdeploy-8c7f7d8bb-bps8s        1/1    Running   0          53s
myfirstdeploy-8c7f7d8bb-rqng8        1/1    Running   0          53s
myfirstdeploy-8c7f7d8bb-t5bm6        1/1    Running   0          9m13s
myfirstdeploy-8c7f7d8bb-vhjhz        1/1    Running   0          9m29s
root@controlplane:~#
  
```

8) Create a persistent volume redis-pv with below specs:

- i) hostpath /mnt/redis/data
- ii) storage size 2Gi
- iii) access mode – ReadWriteOnce

redispv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: redis-pv
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/redis/data"
```

```
kubectl apply -f redispv.yaml
```

```
kubectl get pv redis-pv
```


The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue" buttons. Below the navigation bar, there is a "Task" section with a "Hint" button and a timer showing 48:11. The task asks: "How many Services exist on the system? In the current(default) namespace". Below the question, there are five radio button options: 3, 0, 2, 4, and 1. To the right of the task is a "Terminal 1" window with a dark background. The terminal shows the following YAML content:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: redis-pv
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/redis/data"
```

The terminal also shows the file name "redis-pv.yaml" and its size "14L, 317C".

The screenshot shows the same web browser window as the previous one, but the timer now shows 47:43. The task and options remain the same. The "Terminal 1" window now shows the output of the following commands:

```
root@controlplane:~# kubectl apply -f redis-pv.yaml
persistentvolume/redis-pv created
root@controlplane:~# vim redis-pv.yaml
root@controlplane:~# kubectl get pv redis-pv
NAME      CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM  STORAGECLASS  REASON  AGE
redis-pv  2Gi       RWO           Retain          Available  Claim  manual         34s
root@controlplane:~#
```

9) Create a persistent volume claim redis-pvc that claims redis-pv persistent volume

redis-pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: redis-pvc
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

kubectl apply -f redis-pvc.yaml

kubectl get pv redis-pv

kubectl get pvc redis-pvc

The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue". Below the navigation bar, there is a "Task" section with a "Hint" button and a timer showing 43:26. The task asks: "How many Services exist on the system? in the current(default) namespace". There are five radio button options: 3, 0, 2, 4, and 1. To the right of the task is a "Terminal 1" window showing a YAML configuration for a PersistentVolumeClaim:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: redis-pvc
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

The terminal window also shows the filename "redis-pvc.yaml" and its size "12L, 260C".

The screenshot shows the same web browser window as the previous one, but the timer now shows 42:08. The task and options remain the same. The "Terminal 1" window now shows the output of two Kubernetes commands:

```
root@controlplane:~# kubectl get pv redis-pv
NAME      CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM          STORAGECLASS  REASON  AGE
redis-pv  2Gi       RWO           Retain          Bound   default/redis-pvc  manual              5m56s

root@controlplane:~# kubectl get pvc redis-pvc
NAME      STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  AGE
redis-pvc  Bound   redis-pv  2Gi       RWO           manual         80s
```

The terminal window also shows the filename "redis-pvc.yaml" and its size "12L, 260C".

10) Create a pod redis which binds the redis-pvc to the path /data with image redis:6.2.2

pod-def.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  volumes:
    - name: redis-storage
      persistentVolumeClaim:
        claimName: redis-pvc
  containers:
    - name: redisv-container
      image: redis:6.2.2
      ports:
        - containerPort: 80
          name: redis
      volumeMounts:
        - mountPath: "/data"
          name: redis-storage
```

The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue". Below the navigation bar, there is a "Task" section on the left and a "Terminal 1" section on the right.

Task Section:

How many Services exist on the system?
in the current(default) namespace

Options: 3, 0, 2, 4, 1

Terminal 1:

```
root@controlplane:~# kubectl apply -f pod-def.yaml
pod/redis created
root@controlplane:~# kubectl get pod redis
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          69s
root@controlplane:~#
```

The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue". Below the navigation bar, there is a "Task" section on the left and a "Terminal 1" section on the right.

Task Section:

How many Services exist on the system?
in the current(default) namespace

Options: 3, 0, 2, 4, 1

Terminal 1:

```
root@controlplane:/mnt/redis/data# touch index.html
root@controlplane:/mnt/redis/data# vim index.html
root@controlplane:/mnt/redis/data# ls
index.html
root@controlplane:/mnt/redis/data# cat index.html
Hello Friends, This is Sakthi Sri
root@controlplane:/mnt/redis/data#
```

The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with "Previous Lecture" and "Complete and Continue" buttons. On the left, a "Task" sidebar shows a progress bar from 01 to 11, with 01 selected. Below the progress bar, a hint asks: "How many Services exist on the system? In the current(default) namespace". Below the hint are five buttons with numbers: 3, 0, 2, 4, and 1. The main area is a terminal window with the following commands and output:

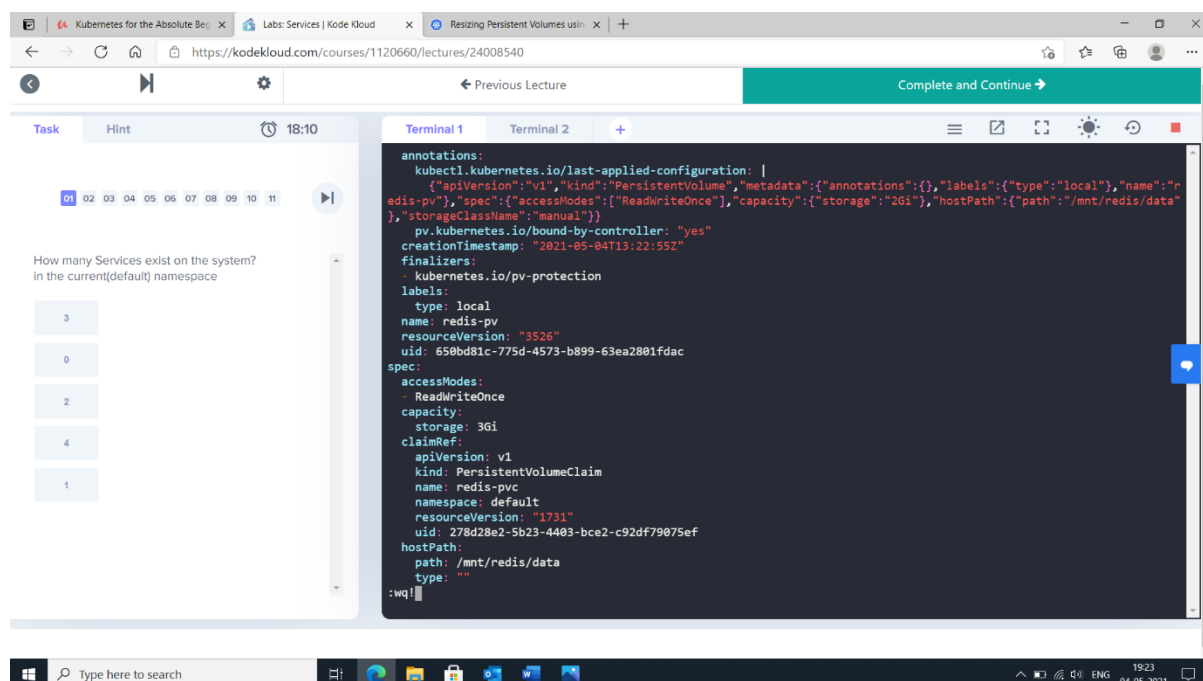
```
root@controlplane:~# pwd
/root
root@controlplane:~# ls
pod-def.yaml redis-pv.yaml service-definition-1.yaml
root@controlplane:~# kubectl apply -f redis-pv.yaml
persistentvolume/redis-pv unchanged
root@controlplane:~# kubectl apply -f redis-pvc.yaml
persistentvolumeclaim/redis-pvc unchanged
root@controlplane:~# kubectl apply -f pod-def.yaml
pod/redis configured
root@controlplane:~# kubectl get pod redis
NAME READY STATUS RESTARTS AGE
redis 1/1 Running 0 11m
root@controlplane:~# kubectl get pv redis-pv
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE
redis-pv 2Gi RWO Retain Bound default/redis-pvc manual 23m
root@controlplane:~# kubectl get pvc redis-pvc
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
redis-pvc Bound redis-pv 2Gi RWO manual 18m
root@controlplane:~#
```

The screenshot shows the same web browser window as above, but the terminal window has been updated with the following commands and output:

```
root@controlplane:~# kubectl exec -it redis -- /bin/bash
root@redis:/data# cat index.html
Hello Friends, This is Sakthi Sri
root@redis:/data# ls
index.html
root@redis:/data# ls
index.html
root@redis:/data# exit
exit
root@controlplane:~# ls
pod-def.yaml redis-pv.yaml service-definition-1.yaml
root@controlplane:~#
```

11) Update the storage size of the redis persistent volume to 3Gi and record the change

```
kubectl edit pv redis-pv
```



The screenshot shows a web browser window with the URL <https://kodekloud.com/courses/1120660/lectures/24008540>. The page has a navigation bar with 'Previous Lecture' and 'Complete and Continue' buttons. Below the navigation bar, there is a 'Task' section with a 'Hint' button and a timer showing 16:48. The task question is: 'How many Services exist on the system? in the current(default) namespace'. There are five radio button options: 3, 0, 2, 4, and 1. To the right of the task is a terminal window labeled 'Terminal 1' and 'Terminal 2'. The terminal shows the following commands and output:

```
root@controlplane:~# kubectl edit pv redis-pv --record
persistentvolume/redis-pv edited
root@controlplane:~# kubectl get pv redis-pv
NAME      CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM          STORAGECLASS  REASON  AGE
redis-pv  3Gi       RWO           Retain          Bound   default/redis-pvc  manual              38m
root@controlplane:~# kubectl get pvc redis-pvc
NAME      STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  AGE
redis-pvc  Bound   redis-pv  2Gi       RWO           manual         26m
root@controlplane:~# kubectl get pod redis
NAME      READY  STATUS  RESTARTS  AGE
redis     1/1    Running  0          19m
root@controlplane:~#
```

The terminal window also has a 'Terminal 2' tab and a '+ ' button to open more terminals. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock in the bottom right corner shows 19:24 on 04-05-2021.