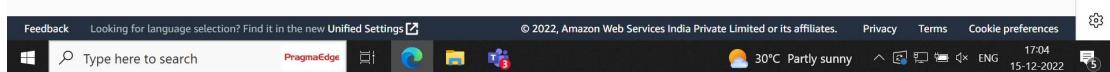


```
apiVersion: v1
kind: ReplicationController
metadata:
  name: myrc1
spec:
  replicas: 1
  selector:
    dc: IN
  template:
    metadata:
      name: "mypod"
      labels:
        dc : IN
    spec:
      containers:
        - name : "myrc1"
          image : "httpd"
```

"rc.yaml" [readonly] 19L, 279B

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 65.0.173.121 PrivateIPs: 172.31.12.246

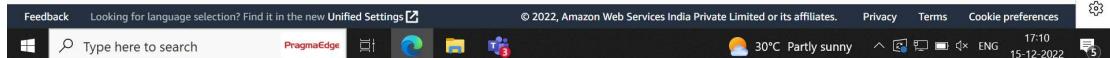


```
apiVersion: v1
kind: ReplicationController
metadata:
  name: myrc1
spec:
  replicas: 1
  selector:
    dc: US
  template:
    metadata:
      name: "mypod"
      labels:
        dc : US
    spec:
      containers:
        - name : "myrc1"
          image : "httpd"
```

-- INSERT --

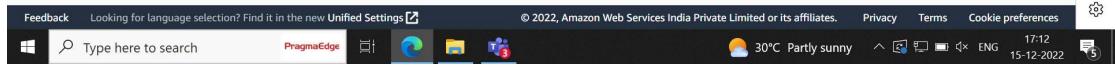
i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 65.0.173.121 PrivateIPs: 172.31.12.246



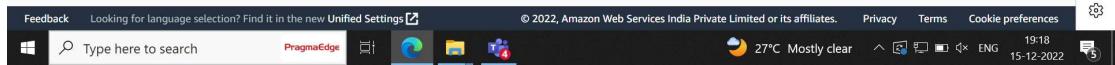
```
[ec2-user@ip-172-31-12-246 ~]$ sudo vim rc.yml
[ec2-user@ip-172-31-12-246 ~]$ kubectl create -f rc.yml
replicationcontroller/myrc1 created
[ec2-user@ip-172-31-12-246 ~]$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
mypod5     1/1     Running   0          3m7s
myrc1-c468b 1/1     Running   0          6s
[ec2-user@ip-172-31-12-246 ~]$ kubectl get pods --show-labels
NAME        READY   STATUS    RESTARTS   AGE   LABELS
mypod5     1/1     Running   0          3m24s   dc=IN,team=team3
myrc1-c468b 1/1     Running   0          23s    dc=US
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)
Public IPs: 65.0.173.121 Private IPs: 172.31.12.246



```
Last login: Thu Dec 15 13:20:15 2022 from ec2-13-233-177-4.ap-south-1.compute.amazonaws.com
[ec2-user@ip-172-31-12-246 ~]$ sudo yum update
20 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po
NAME        READY   STATUS    RESTARTS   AGE
myrc1-7fq9s  1/1     Running   0          122m
[ec2-user@ip-172-31-12-246 ~]$ kubectl get rc
NAME   DESIRED   CURRENT   READY   AGE
myrc1  1          1          1      123m
[ec2-user@ip-172-31-12-246 ~]$ kubectl scale rc/myrc1 --replicas=3
replicationcontroller/myrc1 scaled
[ec2-user@ip-172-31-12-246 ~]$ kubectl get rc
NAME   DESIRED   CURRENT   READY   AGE
myrc1  3          3          3      127m
[ec2-user@ip-172-31-12-246 ~]$
```

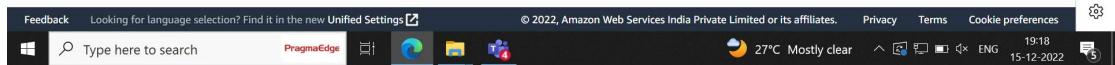
i-03d5a71bf0df22cbe (kubernetes)
Public IPs: 65.0.173.121 Private IPs: 172.31.12.246



```
https://aws.amazon.com/amazon-linux-2/
20 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po
NAME        READY   STATUS    RESTARTS   AGE
myrc1-7fq9s  1/1     Running   0          122m
[ec2-user@ip-172-31-12-246 ~]$ kubectl get rc
NAME      DESIRED   CURRENT   READY   AGE
myrc1     1         1         1       123m
[ec2-user@ip-172-31-12-246 ~]$ kubectl scale rc/myrc1 --replicas=3
replicationcontroller/myrc1 scaled
[ec2-user@ip-172-31-12-246 ~]$ kubectl get rc
NAME      DESIRED   CURRENT   READY   AGE
myrc1     3         3         3       127m
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po
NAME        READY   STATUS    RESTARTS   AGE
myrc1-7fq9s  1/1     Running   0          126m
myrc1-gn192  1/1     Running   0          14s
myrc1-xxl85  1/1     Running   0          14s
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 65.0.173.121 PrivateIPs: 172.31.12.246



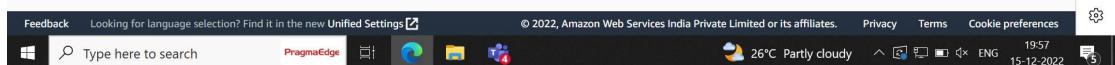
```
https://aws.amazon.com/amazon-linux-2/
20 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-12-246 ~]$ kubectl expose -h
Expose a resource as a new Kubernetes service.

Looks up a deployment, service, replica set, replication controller or pod by name and uses the selector for that resource as the selector for a new service on the specified port. A deployment or replica set will be exposed as a service only if its selector is convertible to a selector that service supports, i.e. when the selector contains only the matchLabels component. Note that if no port is specified via --port and the exposed resource has multiple ports, all will be re-used by the new service. Also if no labels are specified, the new service will re-use the labels from the resource it exposes.

Possible resources include (case insensitive):
pod (po), service (svc), replicationcontroller (rc), deployment (deploy), replicaset (rs)
```

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 65.0.173.121 PrivateIPs: 172.31.12.246



```
If non-empty, set the session affinity for the service to this; legal values: 'None', 'ClientIP'  
--show-managed-fields=false:  
    If true, keep the managedFields when printing objects in JSON or YAML format.  
--target-port='':  
    Name or number for the port on the container that the service should direct traffic to. Optional.  
--template='':  
    Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format  
    is golang templates [http://golang.org/pkg/text/template/#pkg-overview].  
--type='':  
    Type for this service: ClusterIP, NodePort, LoadBalancer, or ExternalName. Default is 'ClusterIP'.  
  
Usage:  
  kubectl expose {--f FILENAME | --type NAME} [--port=port] [--protocol=TCP|UDP|SCTP] [--target-port=number-or-name]  
  [-name=name] [-external-ip=external-ip-of-service] [-type=type] [options]  
  
Use "kubectl options" for a list of global command-line options (applies to all commands).  
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)
Public IPs: 65.0.173.121 Private IPs: 172.31.12.246

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl run myweb --image=vimal13/apache-webserver-php  
pod/myweb created  
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po  
NAME READY STATUS RESTARTS AGE  
myweb 0/1 ContainerCreating 0 4s  
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po  
NAME READY STATUS RESTARTS AGE  
myweb 0/1 ContainerCreating 0 9s  
[ec2-user@ip-172-31-12-246 ~]$ watch kubectl get po  
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po  
NAME READY STATUS RESTARTS AGE  
myweb 1/1 Running 0 33s  
[ec2-user@ip-172-31-12-246 ~]$ kubectl exec -it myweb -- bash  
error: unknown flag: --bash  
See 'kubectl exec --help' for usage.  
[ec2-user@ip-172-31-12-246 ~]$ kubectl exec -it myweb -- bash  
[root@myweb ~]$ ls  
anaconda-post.log bin boot dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var  
[root@myweb ~]$
```

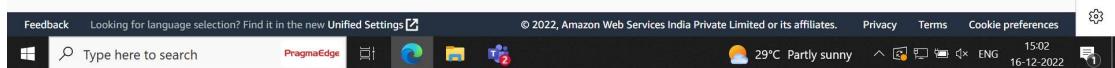
i-03d5a71bf0df22cbe (kubernetes)
Public IPs: 13.126.189.175 Private IPs: 172.31.12.246

```
[root@myweb ~]$
```

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl expose pod myweb --type=NodePort --port=80
service/myweb exposed
[ec2-user@ip-172-31-12-246 ~]$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP     2d20h
myweb      NodePort  10.99.183.240  <none>        80:31611/TCP  11s
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

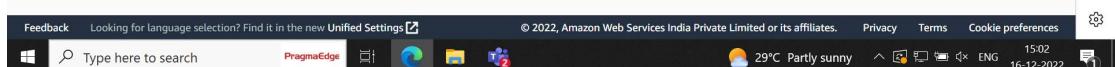
Public IPs: 13.126.189.175 Private IPs: 172.31.12.246



```
kubernetes  ClusterIP  10.96.0.1    <none>        443/TCP     2d20h
myweb      NodePort  10.99.183.240  <none>        80:31611/TCP  11s
[ec2-user@ip-172-31-12-246 ~]$ kubectl describe svc/myweb
Name:           myweb
Namespace:      default
Labels:         run=myweb
Annotations:   <none>
Selector:       run=myweb
Type:          NodePort
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.99.183.240
IPs:           10.99.183.240
Port:          <unset>  80/TCP
TargetPort:    80/TCP
NodePort:      <unset>  31611/TCP
Endpoints:    172.17.0.2:80
Session Affinity: None
External Traffic Policy: Cluster
Events:        <none>
```

i-03d5a71bf0df22cbe (kubernetes)

Public IPs: 13.126.189.175 Private IPs: 172.31.12.246



```

apiVersion : v1
kind : Service
metadata :
  name : mywebservice
spec :
  type : NodePort
  selector :
    dc : IN
  ports :
    - port : 80
      targetPort : 80
      nodePort : 30000

```

"nodesvc.yml" [readonly] 12L, 207B 1,1 All

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Windows Type here to search PragmaEdge 29°C Partly sunny 15:06 16-12-2022

```

myweb   NodePort  10.99.183.240 <none>     80:31611/TCP  16m
mywebservice  NodePort  10.105.147.176 <none>     80:30000/TCP  8s
[ec2-user@ip-172-31-12-246 ~]$ kubectl describe svc/mywebservice
Name:           mywebservice
Namespace:      default
Labels:         <none>
Annotations:   <none>
Selector:      dc=IN
Type:          NodePort
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.105.147.176
IPs:           10.105.147.176
Port:          <unset>  80/TCP
TargetPort:    80/TCP
NodePort:      <unset>  30000/TCP
Endpoints:    <none>
Session Affinity: None
External Traffic Policy: Cluster
Events:        <none>
[ec2-user@ip-172-31-12-246 ~]$ vim nodesvc.yml

```

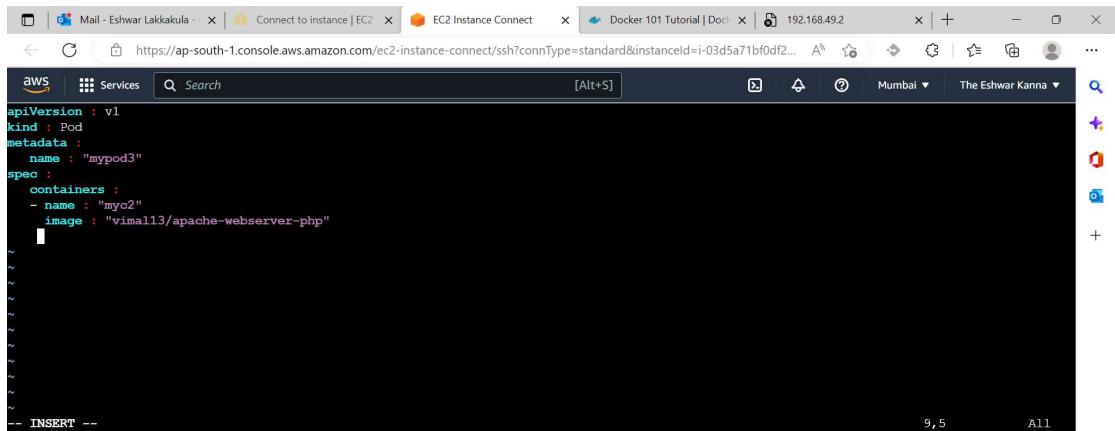
i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Windows Type here to search PragmaEdge 30°C Partly sunny 15:20 16-12-2022



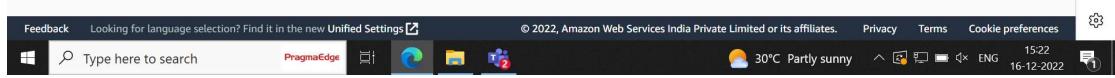
```

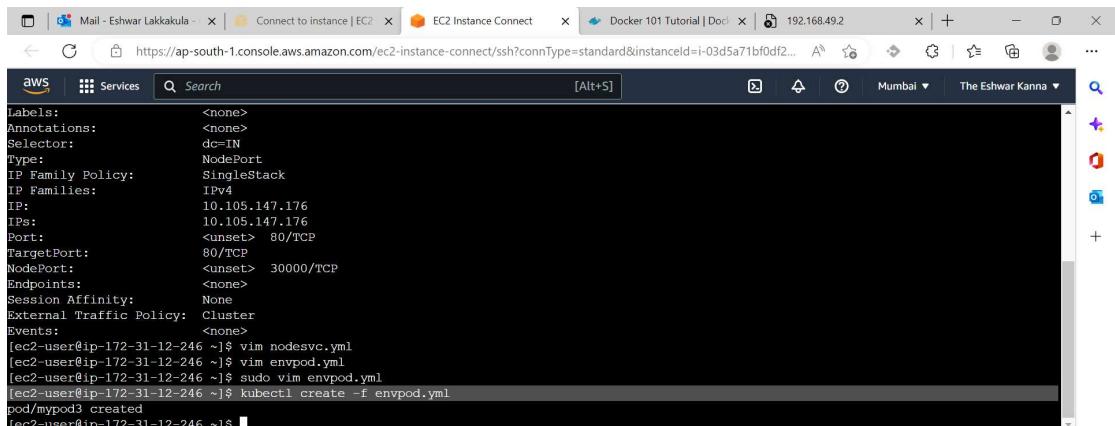
apiVersion : v1
kind : Pod
metadata :
  name : "mypod3"
spec :
  containers :
    - name : "myc2"
      image : "vimal13/apache-webserver-php"

```

-- INSERT --

i-03d5a71bf0df22cbe (kubernetes)
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246



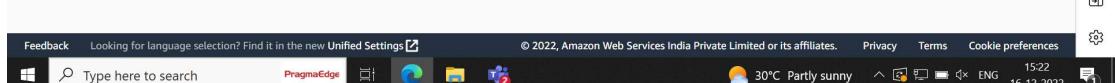


```

Labels: <none>
Annotations: <none>
Selector: dc=IN
Type: NodePort
IP Family Policy: SingleStack
IP Families: IPv4
IP: 10.105.147.176
IPs: 10.105.147.176
Port: <unset> 80/TCP
TargetPort: 80/TCP
NodePort: <unset> 30000/TCP
Endpoints: <none>
Session Affinity: None
External Traffic Policy: Cluster
Events: <none>
[ec2-user@ip-172-31-12-246 ~]$ vim nodesvc.yml
[ec2-user@ip-172-31-12-246 ~]$ vim envpod.yml
[ec2-user@ip-172-31-12-246 ~]$ sudo vim envpod.yml
[ec2-user@ip-172-31-12-246 ~]$ kubectl create -f envpod.yml
pod/mypod3 created
[ec2-user@ip-172-31-12-246 ~]$ 

```

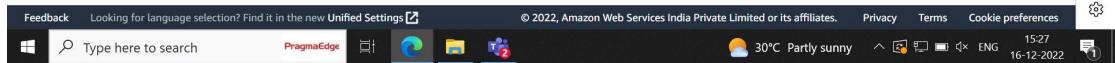
i-03d5a71bf0df22cbe (kubernetes)
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246



```
https://aws.amazon.com/amazon-linux-2/
20 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-12-246 ~]$ kubectl create -f nodesvc.yml
service/mywebservice created
[ec2-user@ip-172-31-12-246 ~]$ kubectl get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
kubernetes   ClusterIP  10.96.0.1    <none>        443/TCP     2d20h
myweb       NodePort   10.99.183.240  <none>        80:31611/TCP 16m
mywebservice  NodePort  10.105.147.176  <none>        80:30000/TCP  8s
[ec2-user@ip-172-31-12-246 ~]$ kubectl describe svc/mywebservice
Name:           mywebservice
Namespace:      default
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po
NAME     READY   STATUS    RESTARTS   AGE
mypod3  1/1     Running   0          4m16s
myweb   1/1     Running   0          27m
```

i-03d5a71bf0df22cbe (kubernetes)

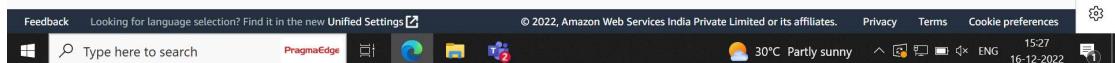
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246



```
Image ID: docker-pullable://vimall13/apache-webserver-php@sha256:faed0a5afaf9f04b6915d73f7247f6f5a71db9274ca44118d38f4601c0080a91
Port: <none>
Host Port: <none>
State: Running
Started: Fri, 16 Dec 2022 09:52:15 +0000
Ready: True
Restart Count: 0
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-zcdsc (ro)
Conditions:
Type Status
Initialized True
Ready True
ContainersReady True
PodScheduled True
Volumes:
  kube-api-access-zcdsc:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
```

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246



aws Services Search [Alt+S] Mumbai The Eshwar Kannan

```
apiVersion : v1
kind : Pod
metadata :
  name : "mypod3"
spec :
  containers :
    - name : "myc2"
      image : "vimal13/apache-webserver-php"
  env:
    - name: X
      value: "67"

-- INSERT --
```

i-03d5a71bf0df22cbe (kubernetes)
 PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 30°C Partly sunny 15:31 16-12-2022

aws Services Search [Alt+S] Mumbai The Eshwar Kannan

```
Restart Count: 0
Environment: <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-zcdsc (ro)
Conditions:
  Type      Status
  Initialized  True
  Ready      True
  ContainersReady  True
  PodScheduled  True
[ec2-user@ip-172-31-12-246 ~]$ kubectl apply -f envpod.yaml
Warning: resource pods/mypod3 is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply.
kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
The Pod "mypod3" is invalid: spec: Forbidden: pod updates may not change fields other than `spec.containers[*].image`, `spec.initContainers[*].image`, `spec.activeDeadlineSeconds`, `spec.tolerations` (only additions to existing tolerations) or `spec.terminationGracePeriodSeconds` (allow it to be set to 1 if it was previously negative)
  core.PodSpec{
    Volumes:    [{Name: "kube-api-access-zcdsc", VolumeSource: {Projected: {Sources: [{ServiceAccountToken: &{ExpirationSeconds: 3607, Path: "token"}}, {ConfigMap: &{LocalObjectReference: {Name: "kube-root-ca.crt"}, Items: [{Key: "ca.crt", Path: "ca.crt"}]}], (DownwardAPI: &{Items: [{Path: "namespace", FieldRef: &{APIVersion: "v1", FieldPath: "metadata.namespace"}]}]}}, DefaultMode: &420}],
```

i-03d5a71bf0df22cbe (kubernetes)
 PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 30°C Partly sunny 15:32 16-12-2022

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl get pods --all --list
# Pod mypod3, container myc2
x=67
# Pod myweb, container myweb
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246


```
apiVersion : v1
kind : Pod
metadata :
  name : "dbpod"
spec :
  containers :
    - name : "mydb"
      image : "mysql"
      env :
        - name : MYSQL_ROOT_PASSWORD
          value : "redhat"
```

"sqldb pod.yaml" [readonly] 11L, 188B

1,1 All

i-03d5a71bf0df22cbe (kubernetes)
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246


```
30°C Partly sunny 15:35 16-12-2022
```

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl set env pods --all --list
# Pod mypod3, container myc2
x=67
# Pod myweb, container myweb
[ec2-user@ip-172-31-12-246 ~]$ ls
clustersvc.yaml first_pod.yaml kubeconfig minikube-linux-amd64 pod_label.yaml rc.yaml rs.yaml test
envpod.yaml get_helm.sh kubectl.sha256 nodesvc.yaml project README.md sqldb_pod.yaml
[ec2-user@ip-172-31-12-246 ~]$ vim sqldb_pod.yaml
[ec2-user@ip-172-31-12-246 ~]$ kubectl apply -f sqldb_pod.yaml
pod/dbpod created
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

Public IPs: 13.126.189.175 Private IPs: 172.31.12.246

```
PodScheduled: True
Volumes:
  kube-api-access-s9p8s:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    DownwardAPI:        true
  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  4m50s  default-scheduler  Successfully assigned default/dbpod to minikube
  Normal  Pulling   4m50s  kubelet         Pulling image "mysql"
  Normal  Pulled   4m37s  kubelet         Successfully pulled image "mysql" in 12.923454547s
  Normal  Created   4m33s  kubelet         Created container mydb
  Normal  Started   4m33s  kubelet         Started container mydb
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

Public IPs: 13.126.189.175 Private IPs: 172.31.12.246

```
11
```

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl exec -it pods/dbpod -- bash  
bash-4.4# mysql -u root -prehat  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 8  
Server version: 8.0.31 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> 
```

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl delete all --all  
pod "dbpod" deleted  
pod "mypod3" deleted  
pod "myweb" deleted  
service "kubernetes" deleted  
service "myweb" deleted  
service "mywebservice" deleted  
[ec2-user@ip-172-31-12-246 ~]$ 
```

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246

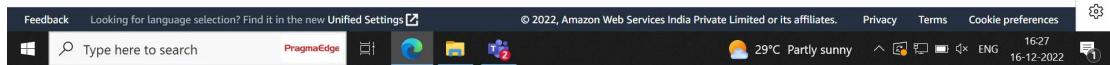
```
[ec2-user@ip-172-31-12-246 ~]$ kubectl delete all --all  
pod "dbpod" deleted  
pod "mypod3" deleted  
pod "myweb" deleted  
service "kubernetes" deleted  
service "myweb" deleted  
service "mywebservice" deleted  
[ec2-user@ip-172-31-12-246 ~]$ 
```

```
apiVersion : v1
kind : Pod
metadata :
  name : "mypod3"
  labels:
    region: IN
    do: south
    team: team1
spec :
  containers :
  - name : "myc2"
    image : "vimal13/apache-webserver-php"

  env:
  - name: x
    value: "67"

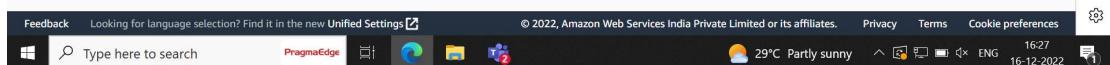
-- INSERT --
```

i-03d5a71bf0df22cbe (kubernetes)
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246



```
[ec2-user@ip-172-31-12-246 ~]$ kubectl apply -f envpod.yaml
pod/mypod3 unchanged
[ec2-user@ip-172-31-12-246 ~]$ kubectl apply -f envpod.yaml
```

i-03d5a71bf0df22cbe (kubernetes)
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246



```
[ec2-user@ip-172-31-12-246 ~]$ kubectl apply -f envpod.yaml
pod/mypod3 unchanged
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
mypod3   1/1     Running   0          31s
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po --selector region=IN
NAME      READY   STATUS    RESTARTS   AGE
mypod3   1/1     Running   0          63s
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po --selector region=IN --show-labels
NAME      READY   STATUS    RESTARTS   AGE   LABELS
mypod3   1/1     Running   0          71s   dc=south,region=IN,team=team1
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

Public IPs: 13.126.189.175 Private IPs: 172.31.12.246

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl apply -f envpod.yaml
pod/mypod3 unchanged
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
mypod3   1/1     Running   0          31s
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po --selector region=IN
NAME      READY   STATUS    RESTARTS   AGE
mypod3   1/1     Running   0          63s
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po --selector region=IN --show-labels
NAME      READY   STATUS    RESTARTS   AGE   LABELS
mypod3   1/1     Running   0          71s   dc=south,region=IN,team=team1
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po --selector "region in (IN) --show-labels
> ^C
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po --selector "region in (IN)" --show-labels
NAME      READY   STATUS    RESTARTS   AGE   LABELS
mypod3   1/1     Running   0          113s  dc=south,region=IN,team=team1
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

Public IPs: 13.126.189.175 Private IPs: 172.31.12.246

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po --selector "region in (IN)" --show-labels
NAME      READY   STATUS    RESTARTS   AGE   LABELS
mypod3   1/1     Running   0          113s  dc=south,region=IN,team=team1
[ec2-user@ip-172-31-12-246 ~]$
```

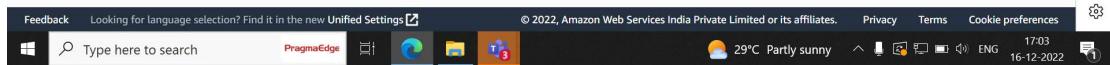
```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myrs1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: prod
  matchExpressions:
    - [key: region, operator: In, values: [south]]
    - [key: team, operator: In, values: [team1]]
  template:
    metadata:
      name: "rspod"
      labels:
        region: south
        team: team1
        app: prod
    spec:
```

"rs.yaml" 23L, 439B

8,15 Top

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246

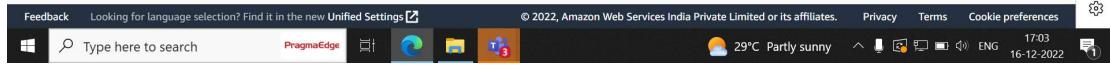


```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myrs1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: prod
  matchExpressions:
    - [key: region, operator: In, values: [south]]
    - [key: team, operator: In, values: [team1]]
  template:
    metadata:
      name: "rspod"
      labels:
        region: south
        team: team1
        app: prod
    spec:
      containers:
        - name: "rsos"
          image: "httpd"
```

23,22 Bot

i-03d5a71bf0df22cbe (kubernetes)

PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246



```
[ec2-user@ip-172-31-12-246 ~]$ kubectl create -f rs.yaml
replicaset.apps/myrs1 created
[ec2-user@ip-172-31-12-246 ~]$ kubectl get rs
NAME    DESIRED   CURRENT   READY   AGE
myrs1   1         1         1       15s
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

Public IPs: 13.126.189.175 Private IPs: 172.31.12.246

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl scale rs/myrs1 --replicas=2
replicaset.apps/myrs1 scaled
[ec2-user@ip-172-31-12-246 ~]$ kubectl get rs
NAME    DESIRED   CURRENT   READY   AGE
myrs1   2         2         1       100s
[ec2-user@ip-172-31-12-246 ~]$ kubectl get po
NAME        READY   STATUS    RESTARTS   AGE
mypod3     1/1     Running   0          38m
myrs1-j5cj4 1/1     Running   0          106s
myrs1-tnfxt 1/1     Running   0          8s
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)

Public IPs: 13.126.189.175 Private IPs: 172.31.12.246

```
[ec2-user@ip-172-31-12-246 ~]$ kubectl create -f deployment.yaml
deployment.apps/myrs1 created
[ec2-user@ip-172-31-12-246 ~]$ kubectl get deploy
NAME    DESIRED   CURRENT   READY   AGE
myrs1   1         1         1       15s
[ec2-user@ip-172-31-12-246 ~]$
```

A screenshot of a Windows desktop environment. At the top, there is a taskbar with several open browser tabs. From left to right, the tabs are: 'Mail - Eshwar L', 'Connect to inst...', 'EC2 Instance C...', 'Docker 101 Tu...', '192.168.49.2...', 'selector match...', 'Labels and Sel...', and '+'. Below the taskbar, a window titled 'aws Services' is open, displaying Kubernetes pod information. The pod details include annotations, replicas, pod status, and a pod template section with labels (app=prod, region=south, team=team1). It also shows containers (rsos), events (Normal SuccessfulCreate), and volumes. At the bottom of the screen, the Windows Start button is visible, followed by a search bar containing 'Type here to search', and a taskbar with icons for PragmaEdge, File Explorer, and other system utilities. The system tray shows the date (16-12-2022), time (17:07), weather (29°C Partly sunny), and battery status.

```
Annotations: <none>
Replicas: 2 current / 2 desired
Pods Status: 2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels: app=prod
           region=south
           team=team1
  Containers:
    rsos:
      Image: httpd
      Port: <none>
      Host Port: <none>
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
Events:
  Type Reason Age From Message
  ---- ---- -- -- -----
  Normal SuccessfulCreate 3m29s replicaset-controller Created pod: myrs1-j5cj4
  Normal SuccessfulCreate 111s replicaset-controller Created pod: myrs1-tnfxt
[ec2-user@ip-172-31-12-246 ~]$
```

i-03d5a71bf0df22cbe (kubernetes)
PublicIPs: 13.126.189.175 PrivateIPs: 172.31.12.246