

```

root@DevImranOps:/tmp/hellov2# docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
visualpath/k8s-demo  V2        39b2b75413a5   2 minutes ago  651.2 MB
<none>              <none>    5418af191c46   7 minutes ago  651.2 MB
samnode              latest    ebe9d473e708   11 hours ago   649.9 MB
visualpath/docker-demo latest   03be3f739a76   6 days ago    651.2 MB
node                 4.6      e834398209c1   4 months ago   645.9 MB
root@DevImranOps:/tmp/hellov2# docker push visualpath/k8s-demo:V2
The push refers to a repository [docker.io/visualpath/k8s-demo]
2987ec31172: Pushed
996cf1e370b0: Pushed
2b11498abede: Layer already exists
eldaa644611ce: Layer already exists
d79093d63949: Layer already exists
87cbe568afdd: Layer already exists
787c930753b4: Layer already exists
9f17712cba0b: Layer already exists
223c0d04a137: Layer already exists
fe4c16cbf7a4: Layer already exists
V2: digest: sha256:8496f6a052ca8b43eb97a6aa48fa4b17a8386bec148ec628051611bd5e47d2c3 size: 2420

```

PUBLIC REPOSITORY

## visualpath/k8s-demo ☆

Last pushed: 2 minutes ago

Repo Info Tags Collaborators Webhooks Settings

Tag Name	Compressed Size	Last Updated
V2	256 MB	2 minutes ago
latest	256 MB	11 hours ago

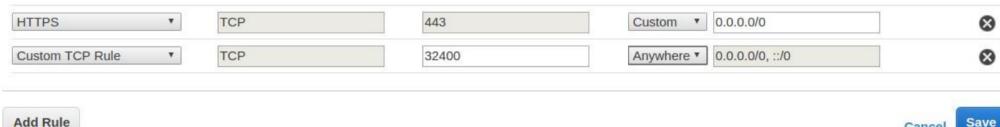
◆ Expose the existing deployment with NodePort.

```

root@DevImranOps:~/kube# kubectl expose deployment helloworld-deployment --type=NodePort
service "helloworld-deployment" exposed
root@DevImranOps:~/kube# kubectl get service
NAME         CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
helloworld-deployment  100.70.171.200  <nodes>      3000:32400/TCP  10s
kubernetes   100.64.0.1   <none>       443/TCP      35m
root@DevImranOps:~/kube# kubectl describe service helloworld-deployment
Name:           helloworld-deployment
Namespace:      default
Labels:         app=helloworld
Selector:       app=helloworld
Type:          NodePort
IP:            100.70.171.200
Port:          <unset>  3000/TCP
NodePort:       <unset>  32400/TCP
Endpoints:     100.96.1.6:3000,100.96.2.4:3000,100.96.2.5:3000
Session Affinity: None
No events.

```

- NodePort is 32400 as per above output.
- Allow this port in Security group of master node so that we can connect to this service.



961 824 5689 E-

- ◆ Verify Master node IP and exposed port.

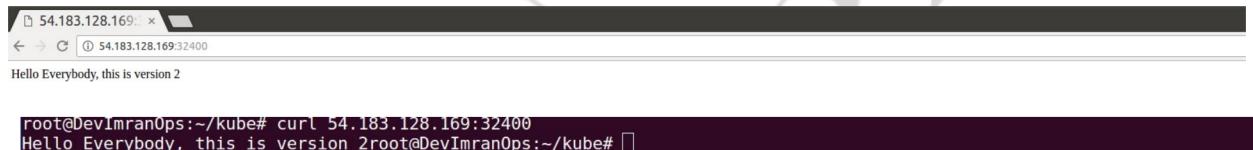


```
54.183.128.169:32400
Hello Everybody!
```

- ◆ Upgrade the image version to V2 from Deployment.

```
root@DevImranOps:~/kube# kubectl set image deployment/helloworld-deployment k8s-demo=visualpath/k8s-demo:V2
deployment "helloworld-deployment" image updated
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-deployment-3758606681-8dg52  1/1     Terminating   0          1m
helloworld-deployment-3758606681-bg12n  1/1     Terminating   0          1m
helloworld-deployment-3758606681-p423x  1/1     Terminating   0          1m
helloworld-deployment-3966748500-8grfs  1/1     Running     0          4s
helloworld-deployment-3966748500-9p0hq  1/1     Running     0          3s
helloworld-deployment-3966748500-r9t6d  1/1     Running     0          4s
root@DevImranOps:~/kube# kubectl rollout status deployment helloworld-deployment
deployment "helloworld-deployment" successfully rolled out
root@DevImranOps:~/kube#
```

- ◆ Verify from browser & curl.



```
54.183.128.169:32400
Hello Everybody, this is version 2
root@DevImranOps:~/kube# curl 54.183.128.169:32400
Hello Everybody, this is version 2
root@DevImranOps:~/kube#
```

- ◆ Deployment Rollout History.

```
root@DevImranOps:~/kube#
root@DevImranOps:~/kube# kubectl rollout history deployment helloworld-deployment
deployments "helloworld-deployment"
REVISION      CHANGE-CAUSE
1            <none>
3            <none>
4            <none>
5            <none>
```

- ◆ Rollback previous version.



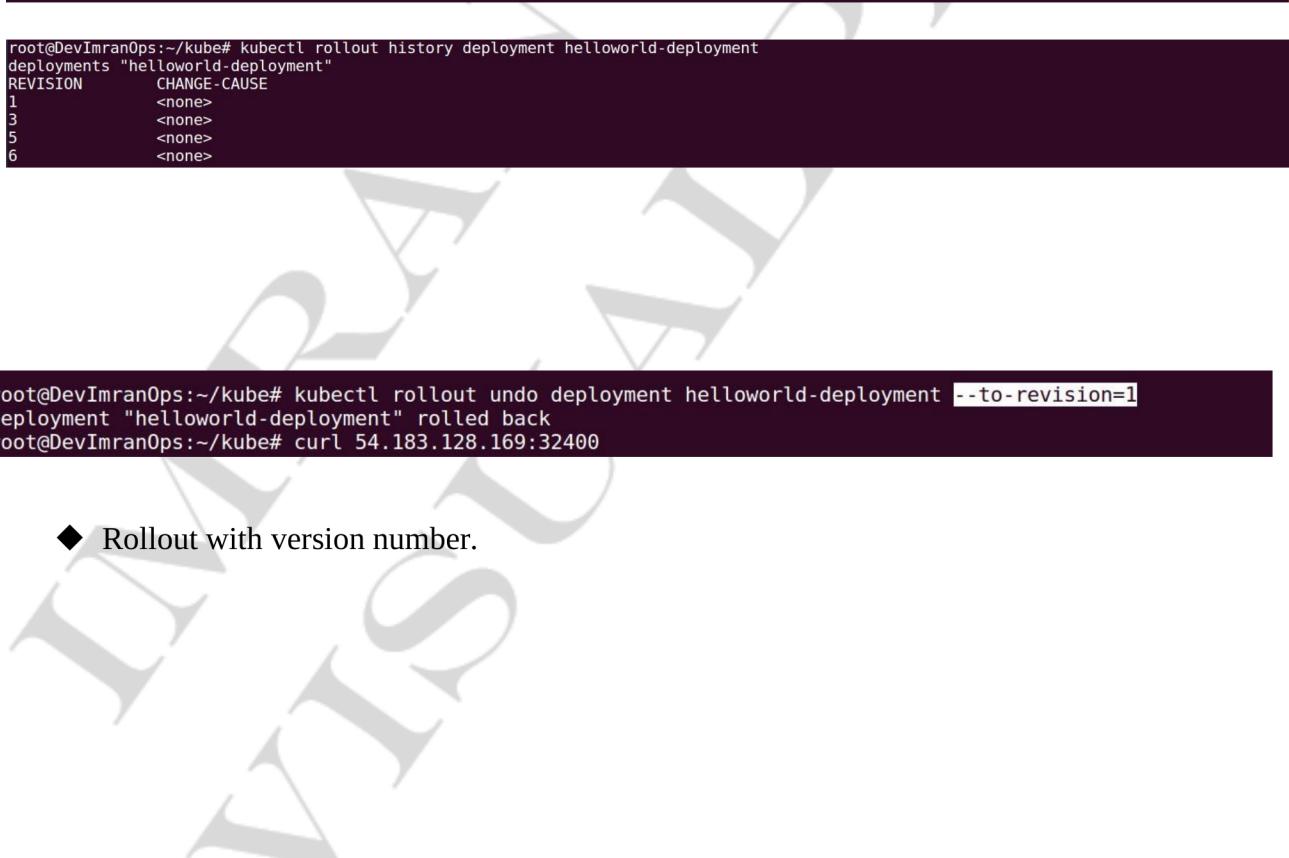
```
root@DevImranOps:~/kube# kubectl rollout undo deployment helloworld-deployment
deployment "helloworld-deployment" rolled back
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-deployment-3758606681-bnntb  1/1     Running   0          7s
helloworld-deployment-3758606681-ltx5r  1/1     Running   0          9s
helloworld-deployment-3758606681-wb0w6  1/1     Running   0          9s
helloworld-deployment-3966748500-8grfs  1/1     Terminating   0          15m
helloworld-deployment-3966748500-9p0hq  1/1     Terminating   0          15m
helloworld-deployment-3966748500-r9t6d  1/1     Terminating   0          15m
root@DevImranOps:~/kube# kubectl rollout status deployment helloworld-deployment
deployment "helloworld-deployment" successfully rolled out
root@DevImranOps:~/kube# curl 54.183.128.169:32400
Hello Everybody!
root@DevImranOps:~/kube#
```

- ◆ Increasing rollout history limit from just two to many.

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

```
root@DevImranOps:~/kube# kubectl edit deployment helloworld-deployment
deployment "helloworld-deployment" edited
```



```
File Edit View Search Terminal Tabs Help
root@DevImranOps:~/kube
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "6"
  creationTimestamp: 2017-04-02T05:14:12Z
  generation: 6
  labels:
    app: helloworld
  name: helloworld-deployment
  namespace: default
  resourceVersion: "5429"
  selfLink: /apis/extensions/v1beta1/namespaces/default/deployments/helloworld-deployment
  uid: 35897346-1763-11e7-a421-066ce5fa5d9e
spec:
  replicas: 3
  revisionHistoryLimit: 50
  selector:
    matchLabels:
      app: helloworld
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
      type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: helloworld
    spec:
      containers:
        - image: visualpath/k8s-demo:latest
          imagePullPolicy: Always
:wq■
```

```
root@DevImranOps:~/kube# kubectl rollout history deployment helloworld-deployment
deployments "helloworld-deployment"
REVISION      CHANGE-CAUSE
1            <none>
3            <none>
5            <none>
6            <none>
```

```
root@DevImranOps:~/kube# kubectl rollout undo deployment helloworld-deployment --to-revision=1
deployment "helloworld-deployment" rolled back
root@DevImranOps:~/kube# curl 54.183.128.169:32400
```

- ◆ Rollout with version number.

## 16. More about Services.

- Pods are very dynamic, they come and go on the Kubernetes cluster.
  - When using a Replication Controller, pods are terminated and created during scaling operations.
  - When using Deployments, when updating the image version, pods are terminated and new pods take the place of older pods.
- That's why Pods should never be accessed directly, but always through a Service.
- A service is the logical bridge between the “mortal” pods and other services or end-users.
- When using the “kubectl expose” command earlier, you created a new Service for your pod, so it could be accessed externally.
- Creating a service will create an endpoint for your pods
  - A ClusterIP: a virtual IP address only reachable from within the cluster.
  - A NodePort: a port that is the same on each node that is also reachable externally.
  - A LoadBalancer: a LoadBalancer created by the cloud provider that will route external traffic to every node on the NodePort.
- ◆ Create a helloworld pod.

```

File Edit View Search Terminal Tabs Help
root@DevImranOps:~/kube          imran@DevOps: ~/_/kops
Image:      visualpath/k8s-demo
Image ID:   docker-pullable://visualpath/k8s-demo@sha256:14051338fef44e6e37b82faeaeaa15bdceff2dd4f2d7af448772d793bc795a2a
Port:       3000/TCP
Requests:
  cpu:        100m
  State:     Running
  Started:   Sun, 02 Apr 2017 06:03:24 +0000
  Ready:     True
  Restart Count: 0
  Volume Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from default-token-h7kw0 (ro)
  Environment Variables: <none>
Conditions:
  Type Status
  Initialized True
  Ready True
  PodScheduled True
Volumes:
  default-token-h7kw0:
    Type: Secret (a volume populated by a Secret)
    SecretName: default-token-h7kw0
    QoS Class: Burstable
    Tolerations: <none>
Events:
FirstSeen  LastSeen  Count  From               SubObjectPath  Type  Reason
---        ---       ---   ---               ---           ---   ---
36s       36s       1      {default-scheduler} 
cheduled  Successfully assigned nodehelloworld.example.com to ip-172-20-63-195.us-west-1.compute.internal
35s       35s       1      {kubelet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal  P
ulling   pulling image "visualpath/k8s-demo"
34s       34s       1      {kubelet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal  P
ulled    Successfully pulled image "visualpath/k8s-demo"
34s       34s       1      {kubelet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal  C
reated   Created container with docker id 111b8b3b7dda; Security:[seccomp=unconfined]
34s       34s       1      {kubelet ip-172-20-63-195.us-west-1.compute.internal} spec.containers{k8s-demo}  Normal  S
tarted   Started container with docker id 111b8b3b7dda
root@DevImranOps:~/kube# 

```

- ◆ Create a Service for the above pod, selector value is the name of the pod as highlighted.

```

root@DevImranOps:~/kube# cat first-app/helloworld-nodeport-service.yml
apiVersion: v1
kind: Service
metadata:
  name: helloworld-service
spec:
  ports:
    - port: 31001
      nodePort: 31001
      targetPort: nodejs-port
      protocol: TCP
    selector:
      app: helloworld
      type: NodePort
root@DevImranOps:~/kube# kubectl create -f first-app/helloworld-nodeport-service.yml
service "helloworld-service" created
root@DevImranOps:~/kube# kubectl get service
NAME            CLUSTER-IP      EXTERNAL-IP   PORT(S)      AGE
helloworld-service  100.67.119.123  <nodes>      31001:31001/TCP  2m
kubernetes       100.64.0.1      <none>       443/TCP      1h

```

- ◆ Nodeport is explicitly defined in the service definition, observe in below screenshot.

```

root@DevImranOps:~/kube# cat first-app/helloworld-nodeport-service.yml
apiVersion: v1
kind: Service
metadata:
  name: helloworld-service
spec:
  ports:
    - port: 31001
      nodePort: 31001
      targetPort: nodejs-port
      protocol: TCP
    selector:
      app: helloworld
      type: NodePort

```

- ◆ We can use svc as a shortform for service.

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

```

root@DevImranOps:~/kube# kubectl get svc
NAME           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
helloworld-service  100.67.119.123 <none>        31001:31001/TCP  5m
kubernetes     100.64.0.1    <none>        443/TCP       1h
root@DevImranOps:~/kube# kubectl describe svc helloworld-service
Name:           helloworld-service
Namespace:      default
Labels:         <none>
Selector:       app=helloworld
Type:          NodePort
IP:            100.67.119.123
Port:          <unset>  31001/TCP
NodePort:      <unset>  31001/TCP
Endpoints:    100.96.2.21:3000
Session Affinity: None
No events.

```

- ◆ Delete and recreate service will change the internal cluster IP/virtual IP.

```

File Edit View Search Terminal Tabs Help
root@DevImranOps:~/kube# kubectl get svc
NAME           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
helloworld-service  100.67.119.123 <none>        31001:31001/TCP  5m
kubernetes     100.64.0.1    <none>        443/TCP       1h
root@DevImranOps:~/kube# kubectl delete svc helloworld-service
service "helloworld-service" deleted
root@DevImranOps:~/kube# kubectl create -f first-app/helloworld-service.yaml
service "helloworld-service" created
root@DevImranOps:~/kube# kubectl get svc
NAME           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
helloworld-service  108.69.176.221  ac6d9af47176f...  80:32678/TCP  7s
kubernetes     100.64.0.1    <none>        443/TCP       2h
root@DevImranOps:~/kube# kubectl describe svc helloworld-service
Name:           helloworld-service
Namespace:      default
Labels:         <none>
Selector:       app=helloworld
Type:          LoadBalancer
IP:            108.69.176.221
LoadBalancer Ingress: ac6d9af47176f11e7a421066ce5fa5d9-897774110.us-west-1.elb.amazonaws.com
Port:          <unset>  80/TCP
NodePort:      <unset>  32678/TCP
Endpoints:    100.96.2.21:3000
Session Affinity: None
Events:
FirstSeen     LastSeen     Count   From             SubObjectPath   Type      Reason           Message
-----     -----     ----   ...              ...           ...      ...
15s          15s          1      {service-controller}   Normal   CreatingLoadBalancer   Creating load balance
13s          13s          1      {service-controller}   Normal   CreatedLoadBalancer   Created load balancer
root@DevImranOps:~/kube# 

```

## 17. Labels

- Labels are key/value pair that can be attached to objects.
  - Labels are like tags in AWS.
- You can label your objects, for instance your pod, following an organizational structure.
  - Key: environment – Value: dev/staging/qa/prod
  - Key: department – Value: engineering/finance/marketing

```

root@DevImranOps:~/kube# cat first-app/helloworld.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nodehelloworld.example.com
  labels:
    app: helloworld
spec:

```

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

Labels are not unique and multiple labels can be added to one object. Once labels are attached to an object, you can use filters to narrow down results. This is called as Label Selectors.

Using Label Selectors, you can use matching expression to match labels. For example, a particular pod can only run on a node labelled with “environment” equals “development”.

You can use labels to tag nodes. Once nodes are tagged, you can use label selectors to let pods only run on specific nodes.

- ◆ Selecting a specific node from deployment definition.

```
root@DevImranOps:~/kube# cat deployment/helloworld-nodeselector.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: helloworld-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
        - name: k8s-demo
          image: wardviaene/k8s-demo
      ports:
        - name: nodejs-port
          containerPort: 3000
  nodeSelector:
    hardware: high-spec
```

- ◆ Tag hardware: high-spec is not on any node from our cluster.

```
root@DevImranOps:~/kube# kubectl get nodes --show-labels
NAME           STATUS   AGE     LABELS
ip-172-20-44-214.us-west-1.compute.internal   Ready   2h     beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=t2.micro,beta.kubernetes.io/os=linux,failure-domain.beta.kubernetes.io/region=us-west-1,failure-domain.beta.kubernetes.io/zone=us-west-1a,kubernetes.io/hostname=ip-172-20-44-214.us-west-1.compute.internal
ip-172-20-58-217.us-west-1.compute.internal   Ready,master   2h     beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=t2.micro,beta.kubernetes.io/os=linux,failure-domain.beta.kubernetes.io/region=us-west-1,failure-domain.beta.kubernetes.io/zone=us-west-1a,kubernetes.io/hostname=ip-172-20-58-217.us-west-1.compute.internal,kubernetes.io/role=master
ip-172-20-63-195.us-west-1.compute.internal   Ready   2h     beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=t2.micro,beta.kubernetes.io/os=linux,failure-domain.beta.kubernetes.io/region=us-west-1,failure-domain.beta.kubernetes.io/zone=us-west-1a,kubernetes.io/hostname=ip-172-20-63-195.us-west-1.compute.internal
```

- ◆ Create our deployment which has nodeSelector tag looking for hardware: high-spec tag.

```

root@DevImranOps:~/kube# cat deployment/helloworld-nodeselector.yml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: helloworld-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
        - name: k8s-demo
          image: visualpath/k8s-demo
        ports:
          - name: nodejs-port
            containerPort: 3000
      nodeSelector:
        hardware: high-spec
root@DevImranOps:~/kube# kubectl create -f deployment/helloworld-nodeselector.yml
deployment "helloworld-deployment" created
root@DevImranOps:~/kube# kubectl get pod
  NAME           READY   STATUS    RESTARTS   AGE
helloworld-deployment-2524499403-6pb8z  0/1     Pending   0          9s
helloworld-deployment-2524499403-9c76j  0/1     Pending   0          9s
helloworld-deployment-2524499403-qz0t3  0/1     Pending   0          9s

```

```

File Edit View Search Terminal Tabs Help
root@DevImranOps:~/kube
NAME           READY   STATUS    RESTARTS   AGE
helloworld-deployment-2524499403-6pb8z  0/1     Pending   0          9s
helloworld-deployment-2524499403-9c76j  0/1     Pending   0          9s
helloworld-deployment-2524499403-qz0t3  0/1     Pending   0          9s
nodehelloworld.example.com  1/1     Running   0          1h
root@DevImranOps:~/kube# kubectl describe pod helloworld-deployment-2524499403-6pb8z
Name:           helloworld-deployment-2524499403-6pb8z
Namespace:      default
Node:           /
Labels:         app=helloworld
                pod-template-hash=2524499403
Status:         Pending
IP:
Controllers:   ReplicaSet/helloworld-deployment-2524499403
Containers:
  k8s-demo:
    Image:        visualpath/k8s-demo
    Port:         3000/TCP
    Requests:
      cpu:        100m
    Volume Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-h7kw0 (ro)
    Environment Variables: <none>
Conditions:
  Type  Status
  PodScheduled  False
Volumes:
  default-token-h7kw0:
    Type:       Secret (a volume populated by a Secret)
    SecretName: default-token-h7kw0
QoS Class:      Burstable
Tolerations:    <none>
Events:
FirstSeen     LastSeen      Count   From               SubObjectPath   Type      Reason           Message
-----     -----      ----   ----               -----   ----      ----           -----
4m          5s          19   {default-scheduler }   Warning        FailedScheduling  pod (helloworld-deployment-2524499403-6pb8z) failed to fit in any node
fit failure summary on nodes : MatchNodeSelector (3), PodToleratesNodeTaints (1)
root@DevImranOps:~/kube#

```

- ✓ Observe the pod creation status is pending.
- ✓ Deployment is looking for a node with tag high-spec, which is not found.
- ✓ If we don't have such tag then pod creation will be pending.

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

◆ Let's Tag a node with hardware: high-spec.

```
root@DevImranOps:~/kube# kubectl get nodes
NAME           STATUS    AGE
ip-172-20-44-214.us-west-1.compute.internal Ready     3h
ip-172-20-58-217.us-west-1.compute.internal Ready,master 3h
ip-172-20-63-195.us-west-1.compute.internal Ready     3h
root@DevImranOps:~/kube# kubectl label nodes ip-172-20-44-214.us-west-1.compute.internal hardware=high-spec
node "ip-172-20-44-214.us-west-1.compute.internal" labeled
root@DevImranOps:~/kube# kubectl get nodes --show-labels
NAME           STATUS    AGE      LABELS
ip-172-20-44-214.us-west-1.compute.internal Ready     3h      beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=t2.micro,beta.kubernetes.io/os=linux,failure-domain.beta.kubernetes.io/region=us-west-1,failure-domain.beta.kubernetes.io/zone=us-west-1a,[hardware=high-spec],beta.kubernetes.io/hostname=ip-172-20-44-214.us-west-1.compute.internal
ip-172-20-58-217.us-west-1.compute.internal Ready,master 3h      beta.kubernetes.io/arch=amd64,beta.kubernetes.io/instance-type=t2.micro
```

◆ Pod will automatically scheduled once the label is available.

```
root@DevImranOps:~/kube# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
helloworld-deployment-2524499403-6pb8z 1/1     Running   0          11m
helloworld-deployment-2524499403-9c76j 1/1     Running   0          11m
helloworld-deployment-2524499403-qz0t3  1/1     Running   0          11m
nodehelloworld.example.com 1/1     Running   0          1h
root@DevImranOps:~/kube#
```

```
File Edit View Search Terminal Tabs Help
root@DevImranOps:~/kube
root@DevImranOps:~/kube# kubectl describe pod helloworld-deployment-2524499403-6pb8z
Name:           helloworld-deployment-2524499403-6pb8z
Namespace:      default
Node:          ip-172-20-44-214.us-west-1.compute.internal/172.20.44.214
Start Time:    Sun, 02 Apr 2017 07:44:19 +0000
Labels:        app=helloworld
               pod-template-hash=2524499403
Status:        Running
IP:          100.96.1.13
Controllers:   ReplicaSet/helloworld-deployment-2524499403
Containers:
  k8s-demo:
    Container ID:   docker://d23cadaade95e576e2693db75c8cd82cb31444006144326a54460fc3452c144b
    Image:          visualpath/k8s-demo
    Image ID:       docker-pullable://visualpath/k8s-demo@sha256:14051338fef44e6e37b82faeaal5bdceff2dd4f2d7af448772d793bc795a2a
    Port:          3000/TCP
    Requests:
      cpu:        100m
    State:        Running
    Started:     Sun, 02 Apr 2017 07:44:21 +0000
    Ready:        True
    Restart Count: 0
    Volume Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-h7kw0 (ro)
    Environment Variables: <none>
Conditions:
  Type  Status
  Initialized  True
  Ready  True
  PodScheduled  True
Volumes:
  default-token-h7kw0:
    Type:  Secret (a volume populated by a Secret)
    SecretName: default-token-h7kw0
    Oss Class:  Burstable
    Tolerations: <none>
Events:
FirstSeen  LastSeen  Count  From                    SubObjectPath  Type    Reason            Message
-----  -----  -----  -----  -----  -----  -----  -----
12m       5m        29   {default-scheduler }           Warning   FailedScheduling  pod (helloworld-deployment-2524499403-6pb8z) failed to fit in an
y node
fit failure summary on nodes : MatchNodeSelector (3), PodToleratesNodeTaints (1)
499403-6pb8z to ip-172-20-44-214.us-west-1.compute.internal
4m 4m 1 {kubelet ip-172-20-44-214.us-west-1.compute.internal} spec.containers{k8s-demo} Normal  Pulling  pulling image "visualpath/k8s-demo"
4m 4m 1 {kubelet ip-172-20-44-214.us-west-1.compute.internal} spec.containers{k8s-demo} Normal  Pulled  Successfully pulled image "visualpath/k8s-demo"
4m 4m 1 {kubelet ip-172-20-44-214.us-west-1.compute.internal} spec.containers{k8s-demo} Normal  Created  Created container with docker id d23cadaade95; S
ecurity:[seccomp=unconfined]
4m 4m 1 {kubelet ip-172-20-44-214.us-west-1.compute.internal} spec.containers{k8s-demo} Normal  Started  Started container with docker id d23cadaade95
```

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in)

## 18. Secrets

Secrets provides a way in kubernetes to distribute Credentials, keys and passwords to the pods.

Kubernetes itself uses this secrets mechanism to provide the credentials to access the internal API.

Secrets can be used in following ways.

- Use secrets as environment variables.
- Use secrets as a file in a pod.
  - This setup uses volumes to be mounted in a container.
  - In the volumes, you have files.
- ◆ Base 64 encoding, encoding username & password.

```
root@DevImranOps:~/kube# echo -n "root" | base64  
cm9vdA==  
root@DevImranOps:~/kube# echo -n "password" | base64  
cGFzc3dvcmQ=  
root@DevImranOps:~/kube#
```

- ◆ Creating Secret.

```
root@DevImranOps:~/kube# cat deployment/helloworld-secrets.yml  
apiVersion: v1  
kind: Secret  
metadata:  
  name: db-secrets  
type: Opaque  
data:  
  username: cm9vdA==  
  password: cGFzc3dvcmQ=  
root@DevImranOps:~/kube# kubectl create -f deployment/helloworld-secrets.yml  
secret "db-secrets" created
```

- ◆ Creating deployment which mounts our secret(db-secrets) as a volume.
- ◆ Describe shows the mounted volumes, highlighted is our db-secret mounted as volume.