

◆ Deleting service.

```
root@DevImranOps:~/kube# kubectl get svc
NAME           CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
helloworld-service  100.68.5.83   <nodes>        31001:31001/TCP  14m
kubernetes      100.64.0.1    <none>         443/TCP       49m
root@DevImranOps:~/kube# kubectl delete service helloworld-service
service "helloworld-service" deleted
root@DevImranOps:~/kube#
```

◆ Creating AWS ELB as a Service for our pod.

```
File Edit View Search Terminal Help
root@DevImranOps:~/kube# cat first-app/helloworld-service.yml
apiVersion: v1
kind: Service
metadata:
  name: helloworld-service
spec:
  ports:
  - port: 80
    targetPort: nodejs-port
    protocol: TCP
  selector:
    app: helloworld
  type: LoadBalancer
root@DevImranOps:~/kube# kubectl create -f first-app/helloworld-service.yml
service "helloworld-service" created
root@DevImranOps:~/kube# kubectl get service
NAME           CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
helloworld-service  100.71.244.70  aled2cfa01715...  80:31809/TCP  11s
kubernetes      100.64.0.1    <none>         443/TCP       51m
root@DevImranOps:~/kube# kubectl describe service helloworld-service
Name:           helloworld-service
Namespace:      default
Labels:         <none>
Selector:       app=helloworld
Type:          LoadBalancer
IP:            100.71.244.70
LoadBalancer Ingress: aled2cfa0171511e7a4aa06705515583-201543200.us-west-1.elb.amazonaws.com
Port:          <unset>:80/TCP
NodePort:       <unset>:31809/TCP
Endpoints:     100.96.2.2:3000
Session Affinity: None
Events:
  FirstSeen  LastSeen  Count  From               SubObjectPath  Type      Reason      Message
  ----  ----  ----  ----  ----  ----  ----  ----
  27s        27s      1  {service-controller }  Normal      CreatingLoadBalancer  Creating load balancer
  24s        24s      1  {service-controller }  Normal      CreatedLoadBalancer  Created load balancer
root@DevImranOps:~/kube#
root@DevImranOps:~/kube#
root@DevImranOps:~/kube#
```

◆ Verify ELB from AWS.

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
a1ed2cfa0171511e7a4aa06705515583	a1ed2cfa0171511e7a4aa067...	vpc-5e0da43a	us-west-1a	classic	April 2, 2017 at 1:25:14 AM U...	

Instance ID	Name	Availability Zone	Status	Actions
i-0c0c37a135205a0a1	nodes.kubernetes.devimranops.club	us-west-1a	InService	Remove from Load Balancer
i-0b642810b6ab4a680	nodes.kubernetes.devimranops.club	us-west-1a	InService	Remove from Load Balancer
i-05b594a3675f91e7d	master-us-west-1a.masters.kubernetes.devimranops.club	us-west-1a	InService	Remove from Load Balancer

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
us-west-1a	subnet-3c5acf64	172.20.32.0/19	3	Yes	-

◆ Assign DNS name to our ELB from Route 53.

- Go to AWS route53 dashboard => Hosted Zone => Create Record Set
=> Name:helloworld => Alias:Yes => ELB Classic load balancer
- Select your ELB and use helloworld.kubernetes.devimranops.club (diff domain in your case.)

Create Record Set

Name: .kubernetes.devimranops

Type:

Alias: Yes No

Alias Target:

Alias Hosted Zone ID: Z368ELLRRE2KJ0

You can also type the domain name for the resource. Examples:
 - CloudFront distribution domain name: d111111abcdef8.cloudfront.net
 - Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
 - ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
 - S3 website endpoint: s3-website.us-east-2.amazonaws.com
 - Resource record set in this hosted zone: www.example.com

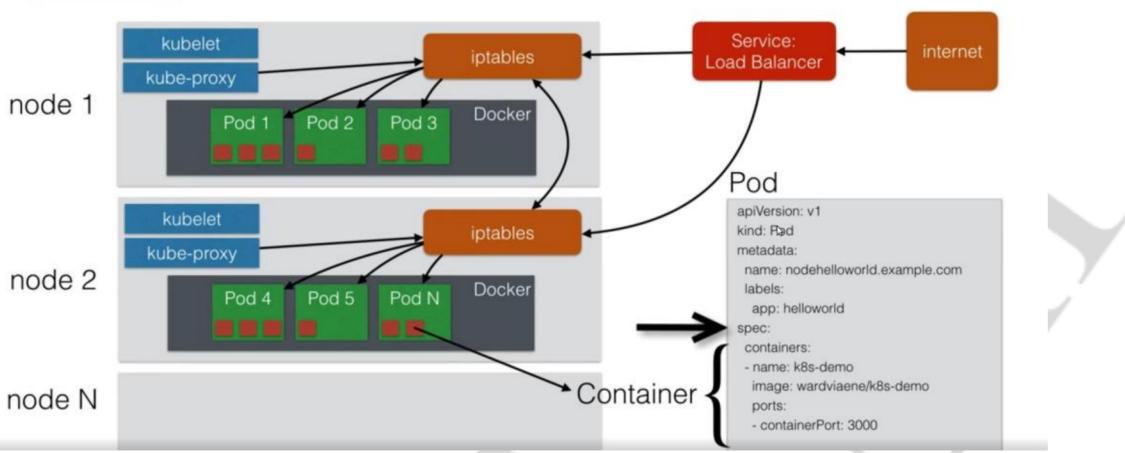
[Learn More](#)

Routing Policy:

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, Website : www.visualpath.in

13. Node Architecture.



14. Scaling Pods

Scaling

- If your application is stateless you can horizontally scale it.
 - Stateless means your application does not store files/data on local file system.
 - All Databases are statefull. They store database files locally.
- Most Web applications can be made stateless.
 - Session Management needs to be done outside the container
 - Any file that need to be saved can't be saved locally on the computer
- Scaling in Kubernetes can be done using the Replication Controller.
- The replication controller will ensure a specified number of pod replicas will run at all time
- A pod created with the replica controller will automatically be replaced if they fail, get deleted or are terminated.
- Using the replication controller is also recommended if you just want to make sure 1 pod is always running, even after reboots.

- ◆ Replicating our pod two times.

```
root@DevImranOps:~/kube# cat replication-controller/helloworld-repl-controller.yml
apiVersion: v1
kind: ReplicationController
metadata:
  name: helloworld-controller
spec:
  replicas: 2
  selector:
    app: helloworld
  template:
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
        - name: k8s-demo
          image: visualpath/k8s-demo
        ports:
          - name: nodejs-port
            containerPort: 3000
```

- ◆ Check pod & rc (replication controller) with kubectl get & kubectl describe command.

```

File Edit View Search Terminal Help
root@DevImranOps:~/kube# kubectl create -f replication-controller/helloworld-repl-controller.yml
replicationcontroller "helloworld-controller" created
root@DevImranOps:~/kube# kubectl get rc
NAME      DESIRED   CURRENT   READY   AGE
helloworld-controller  2         2         2       8s
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-514j9  1/1    Running   0          14s
helloworld-controller-hfdc3  1/1    Running   0          14s
root@DevImranOps:~/kube# kubectl describe pod helloworld-controller-514j9
Name:           helloworld-controller-514j9
Namespace:      default
Labels:         app=helloworld
Status:        Running
IP:            100.96.2.4
Controllers:   ReplicationController/helloworld-controller
Containers:
  k8s-demo:
    Container ID:   docker://8e501fb12b96963e092a0b5e5623707807a3262f24fba1321c825156b4cale
    Image:          visualpath/k8s-demo
    Image ID:       docker-pullable://visualpath/k8s-demo@sha256:877837266a5e78becebb3ed03f0e72ba3acfae9d6fd8f0cc0d83c4a82889d1b1
    Port:          3000/TCP
    Requests:
      cpu:        100m
      memory:     100Mi
    Status:       Running
    Started:     Sat, 01 Apr 2017 20:44:52 +0000
    Ready:        True
    Restart Count: 0
    Volume Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-348lc (ro)
      Environment Variables: <none>
    Conditions:
      Type        Status
      Initialized  True
      Ready       True
      PodScheduled  True
    Volumes:
      default-token-348lc:
        Type:  Secret (a volume populated by a Secret)

```

◆ Delete one pod to see scaling.

```

root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-514j9  1/1    Running   0          1m
helloworld-controller-hfdc3  1/1    Running   0          1m
root@DevImranOps:~/kube# kubectl delete pod helloworld-controller-514j9
pod "helloworld-controller-514j9" deleted
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-514j9  1/1    Terminating   0          1m
helloworld-controller-8lmxh  1/1    Running   0          4s
helloworld-controller-hfdc3  1/1    Running   0          1m

```

◆ Scale Up/Down the replication controller.

```

root@DevImranOps:~/kube# kubectl get rc
NAME      DESIRED   CURRENT   READY   AGE
helloworld-controller  2         2         2       3m
root@DevImranOps:~/kube# kubectl scale --replicas=4 rc helloworld-controller
replicationcontroller "helloworld-controller" scaled
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-8lmxh  1/1    Running   0          2m
helloworld-controller-hfdc3  1/1    Running   0          4m
helloworld-controller-hvzbr  1/1    Running   0          8s
helloworld-controller-qtc93  1/1    Running   0          8s
root@DevImranOps:~/kube# kubectl scale --replicas=1 rc helloworld-controller
replicationcontroller "helloworld-controller" scaled
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-8lmxh  1/1    Terminating   0          2m
helloworld-controller-hfdc3  1/1    Running   0          4m
helloworld-controller-hvzbr  1/1    Terminating   0          24s
helloworld-controller-qtc93  1/1    Terminating   0          24s
root@DevImranOps:~/kube#

```

- ◆ Scale Up/Down with the replication controller's definition file.

```
File Edit View Search Terminal Help
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-htdc3  1/1    Running   0          7m
root@DevImranOps:~/kube# kubectl get rc
NAME          DESIRED  CURRENT  READY     AGE
helloworld-controller   1        1        1        7m
root@DevImranOps:~/kube# kubectl scale --replicas=4 -f replication-controller/helloworld-repl-controller.yml
replicationcontroller "helloworld-controller" scaled
root@DevImranOps:~/kube# kubectl get rc
NAME          DESIRED  CURRENT  READY     AGE
helloworld-controller   4        4        4        8m
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-bqbf3  1/1    Running   0          9s
helloworld-controller-dsprx  1/1    Running   0          9s
helloworld-controller-dw5k5  1/1    Running   0          9s
helloworld-controller-htdc3  1/1    Running   0          8m
root@DevImranOps:~/kube# kubectl scale --replicas=1 -f replication-controller/helloworld-repl-controller.yml
replicationcontroller "helloworld-controller" scaled
root@DevImranOps:~/kube# kubectl get rc
NAME          DESIRED  CURRENT  READY     AGE
helloworld-controller   1        1        1        8m
root@DevImranOps:~/kube# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
helloworld-controller-bqbf3  1/1    Terminating   0          27s
helloworld-controller-dsprx  1/1    Terminating   0          27s
helloworld-controller-dw5k5  1/1    Terminating   0          27s
helloworld-controller-htdc3  1/1    Running   0          8m
root@DevImranOps:~/kube#
```

15. Deployments

Replication Set

- Replica set is the next generation Replication Controller.
- It supports a new selector that can do selection based on filtering according a set of values.
 - e.g “environment” either “dev” or “qa”
 - not only based on equality like the Replication Controller
 - e.g “environment” == “dev”
- This Replica Set rather than the Replication Controller, is used by the Deployment object.

Deployments

- A deployment declaration in Kubernetes allows you to do app deployments and updates.
- When using the deployment object, you define the state of your application.
 - Kubernetes will then make the clusters matches your desired state.
- Just using the replication controller or replication set might be cumbersome to deploy apps.
 - The Deployment Object is easier to use and gives you more possibilities.
- With a deployment object, you can:
 - Create a deployment (e.g deploying an app)
 - Update a deployment (e.g deploying a latest version)
 - Do rolling updates (zero downtime deployments)
 - Roll back to a previous version
 - Pause/Resume a deployment (e.g to roll-out to only a certain percentage)

◆ Sample Deployment Definition.

```
root@DevImranOps:~/kube# cat deployment/helloworld.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
root@DevImranOps:~/kube#
```

◆ Useful Commands

Command	Description
kubectl get deployments	Get information on current deployments
kubectl get rs	Get information about the replica sets
kubectl get pods --show-labels	get pods, and also show labels attached to those pods
kubectl rollout status deployment/helloworld-deployment	Get deployment status
kubectl set image deployment/helloworld-deployment k8s-demo=k8s-demo:2	Run k8s-demo with the image label version 2
kubectl edit deployment/helloworld-deployment	Edit the deployment object
kubectl rollout status deployment/helloworld-deployment	Get the status of the rollout
kubectl rollout history deployment/helloworld-deployment	Get the rollout history
kubectl rollout undo deployment/helloworld-deployment	Rollback to previous version
kubectl rollout undo deployment/helloworld-deployment --to-revision=n	Rollback to any version version

◆ Deployment:

- Create deployment
- Get info on deployment, Replication set (rs), pod, rollout status

```

root@DevImranOps:~/kube# kubectl create -f deployment/helloworld.yml
deployment "helloworld-deployment" created
root@DevImranOps:~/kube# kubectl get deployment
NAME          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
helloworld-deployment  3        3        3           0          27s
root@DevImranOps:~/kube# kubectl get rs
NAME          DESIRED  CURRENT  READY     AGE
helloworld-deployment-706070162  3        3        3          37s
root@DevImranOps:~/kube# kubectl get pods
NAME                           READY   STATUS    RESTARTS  AGE      LABELS
helloworld-deployment-706070162-8n3bs  1/1    Running   0          46s  app=helloworld,pod-template-hash=706070162
helloworld-deployment-706070162-nlpdl  1/1    Running   0          46s  app=helloworld,pod-template-hash=706070162
helloworld-deployment-706070162-q350d  1/1    Running   0          46s  app=helloworld,pod-template-hash=706070162
root@DevImranOps:~/kube# kubectl get pods --show-labels
NAME                           READY   STATUS    RESTARTS  AGE      LABELS
helloworld-deployment-706070162-8n3bs  1/1    Running   0          59s  app=helloworld,pod-template-hash=706070162
helloworld-deployment-706070162-nlpdl  1/1    Running   0          59s  app=helloworld,pod-template-hash=706070162
helloworld-deployment-706070162-q350d  1/1    Running   0          59s  app=helloworld,pod-template-hash=706070162
root@DevImranOps:~/kube# kubectl rollout status deployment helloworld-deployment
deployment "helloworld-deployment" successfully rolled out
root@DevImranOps:~/kube# 

```

◆ Build k8s-demo image with next version, V2.

- Copy docker-demo directory content into another directory.
- Edit index.js file, update string in res.send to something like “version 2”.
- Build the image with tag V2.
- Push the latest image to dockerhub registry.
- Verify from dockerhub

```

root@DevImranOps:~/kube/docker-demo# ls
docker-compose.yml  Dockerfile  index-db.js  index.js  misc  package.json
root@DevImranOps:~/kube/docker-demo# cp * -r /tmp/hellov2/
root@DevImranOps:~/kube/docker-demo# cd /tmp/hellov2/
root@DevImranOps:/tmp/hellov2# vi index.js

```

```

root@DevImranOps:/tmp/hellov2# cat index.js
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello Everybody, this is version 2');
});

var server = app.listen(3000, function () {
  var host = server.address().address;
  var port = server.address().port;

  console.log('Example app listening at http://%s:%s', host, port);
});
```

```

root@DevImranOps:/tmp/hellov2# docker build -t visualpath/k8s-demo:V2 .
Sending build context to Docker daemon 8.704 kB
Step 1 : FROM node:4.6
--> e834398209c1
Step 2 : WORKDIR /app
--> Using cache
--> d808ac4780ac
Step 3 : ADD . /app
--> 7f2f05cfef161
Removing intermediate container 356b8e9ec00a
Step 4 : RUN npm install
--> Running in 5f0fe5862183
npm info it worked if it ends with ok

```

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, Website : www.visualpath.in