

Deployments

» Blue-Green Deployment

Let's create a Blue-Green Deployment

Create a new Service object manifest.

```
cat > bluegreen-service.yaml<<EOF
apiVersion: v1
kind: Service
metadata:
  name: webserver
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
  selector:
    type: webserver
    color: blue
EOF
```

Create the service.

```
kubectl create -f bluegreen-service.yaml
```

Fetch the AWS ELB URL.

```
kubectl get svc webserver -o wide
```

Create the `blue` deployment object manifest.

```
cat > blue-deployment.yaml<<EOF
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: webserver-blue
spec:
  replicas: 3
  template:
    metadata:
      labels:
        type: webserver
        color: blue
    spec:
      containers:
      - image: quay.io/coreosstrainme/hieveryone:1.0.0
        name: webserver-container
        ports:
        - containerPort: 80
          name: http-server
EOF
```

Create the `blue` deployment.

```
kubectl create -f blue-deployment.yaml
```

Create the `green` deployment object manifest.

```
cat > green-deployment.yaml<<EOF
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: webserver-green
spec:
  replicas: 3
  template:
    metadata:
      labels:
        type: webserver
        color: green
    spec:
      containers:
      - image: quay.io/coreosstrainme/hieveryone:2.0.0
        name: webserver-container
        ports:
        - containerPort: 80
          name: http-server
EOF
```

Create the `green` deployment.

```
kubectl create -f green-deployment.yaml
```

Edit the Service so it sends traffic to the `green` deployment.

```
kubectl edit svc webserver
```

Clean-up

```
kubectl delete deploy webserver-blue webserver-green
kubectl delete svc webserver
```