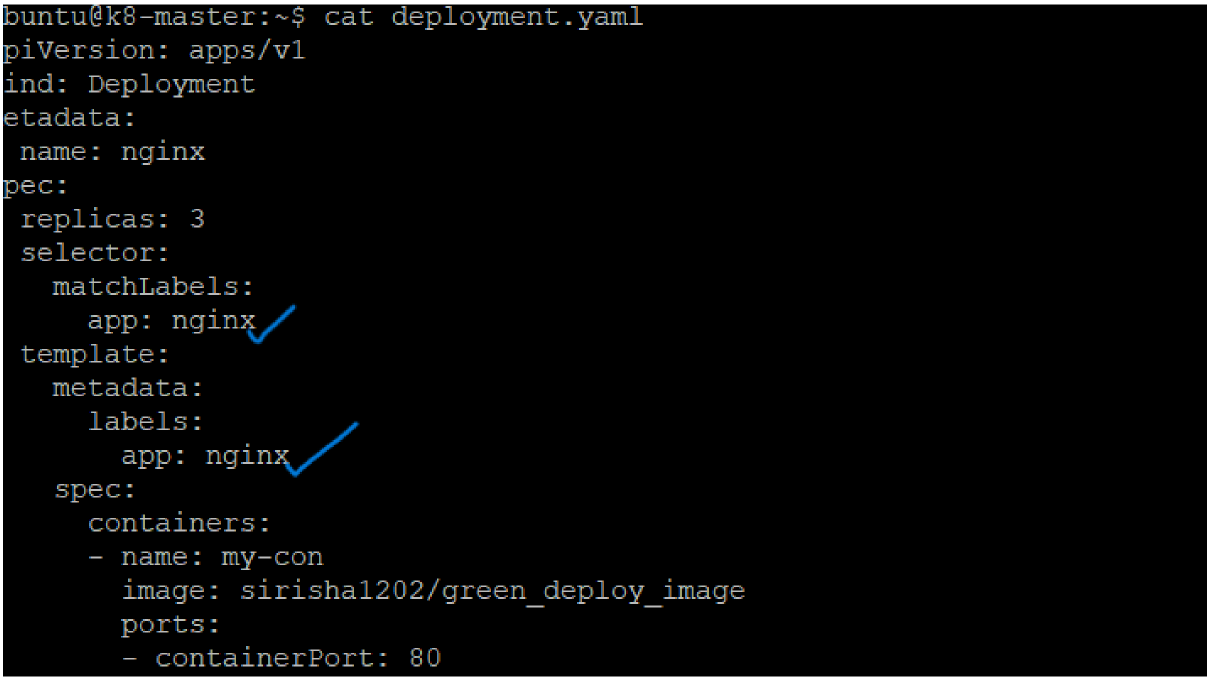
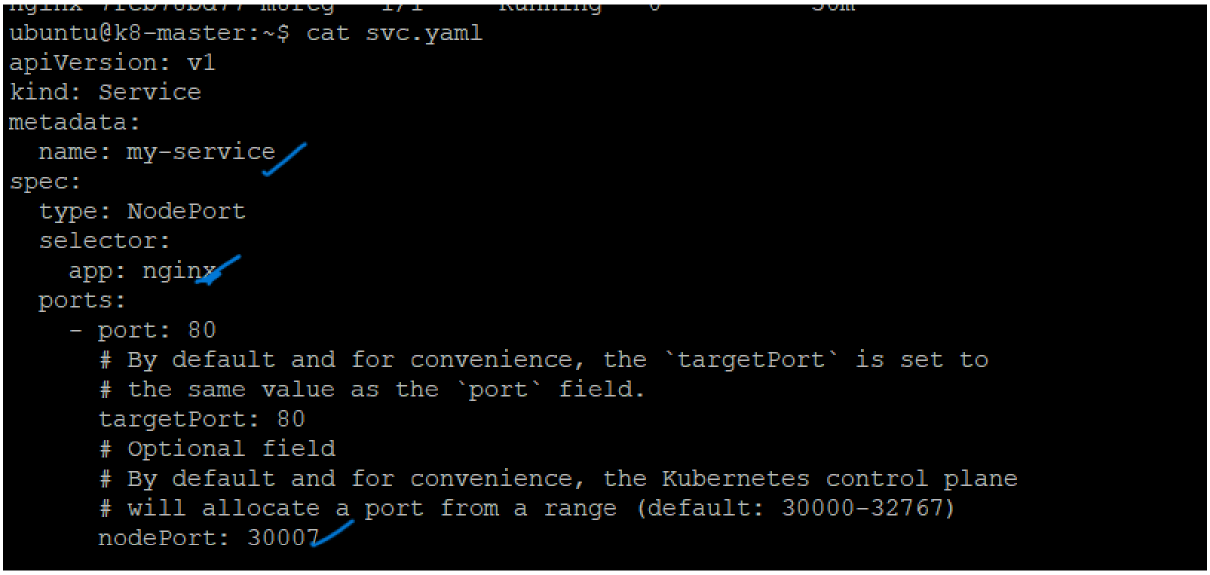
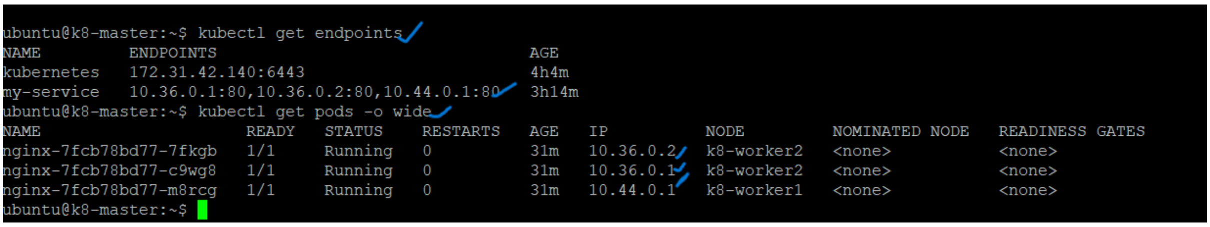
blue_green_deployment:



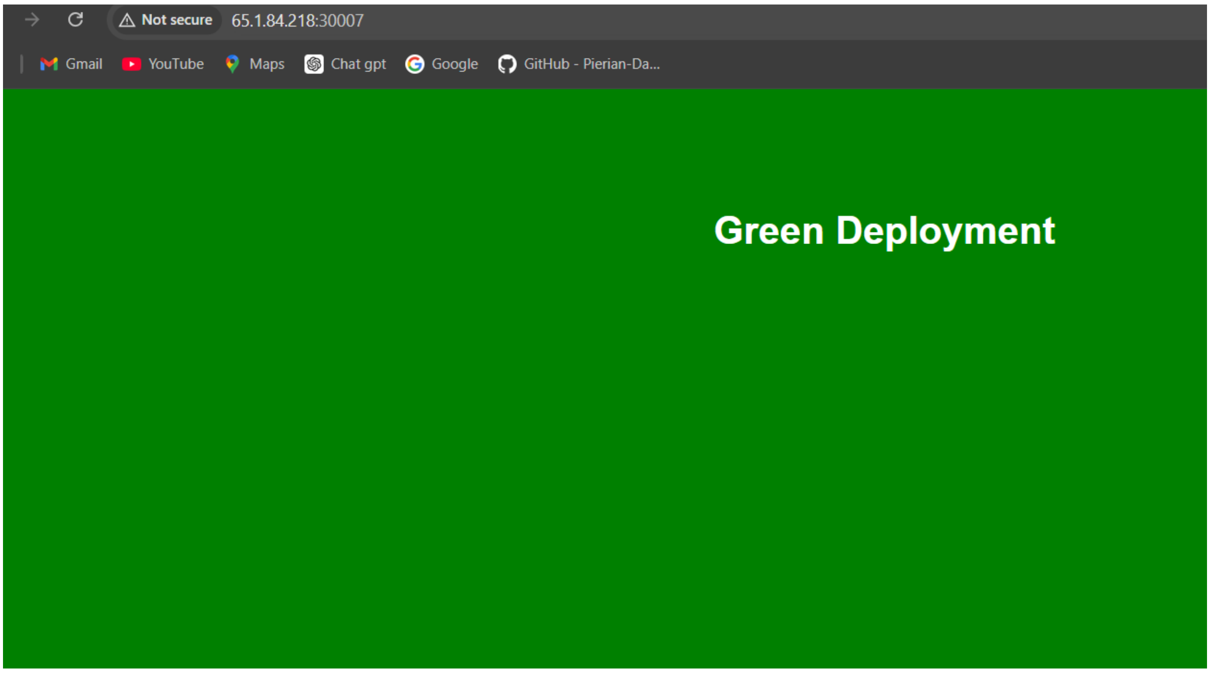
```
buntu@k8-master:~$ cat deployment.yaml
piVersion: apps/v1
ind: Deployment
etadata:
 name: nginx
pec:
 replicas: 3
 selector:
   matchLabels:
     app: nginx
 template:
   metadata:
     labels:
       app: nginx
   spec:
     containers:
     - name: my-con
       image: sirisha1202/green_deploy_image
       ports:
       - containerPort: 80
```



```
ubuntu@k8-master:~$ cat svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      # By default and for convenience, the `targetPort` is set to
      # the same value as the `port` field.
      targetPort: 80
      # Optional field
      # By default and for convenience, the Kubernetes control plane
      # will allocate a port from a range (default: 30000-32767)
      nodePort: 30007
```



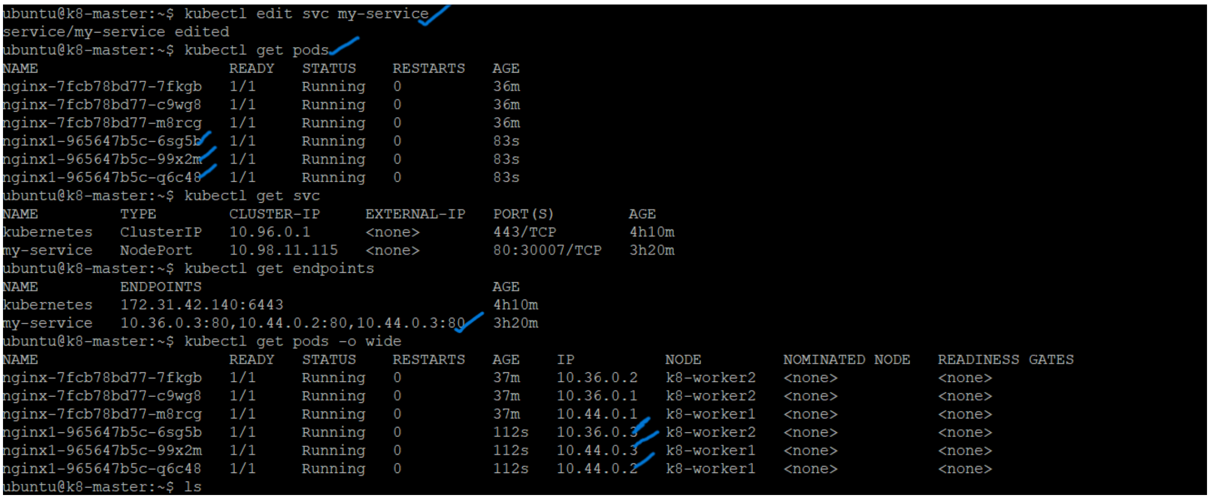
```
ubuntu@k8-master:~$ kubectl get endpoints
NAME         ENDPOINTS                                      AGE
kubernetes   172.31.42.140:6443                             4h4m
my-service   10.36.0.1:80,10.36.0.2:80,10.44.0.1:80         3h14m
ubuntu@k8-master:~$ kubectl get pods -o wide
NAME                       READY   STATUS    RESTARTS   AGE   IP          NODE         NOMINATED NODE   READINESS GATES
nginx-7fcb78bd77-7fkgb     1/1     Running   0          31m   10.36.0.2   k8-worker2   <none>           <none>
nginx-7fcb78bd77-c9wg8     1/1     Running   0          31m   10.36.0.1   k8-worker2   <none>           <none>
nginx-7fcb78bd77-m8rcg     1/1     Running   0          31m   10.44.0.1   k8-worker1   <none>           <none>
ubuntu@k8-master:~$
```



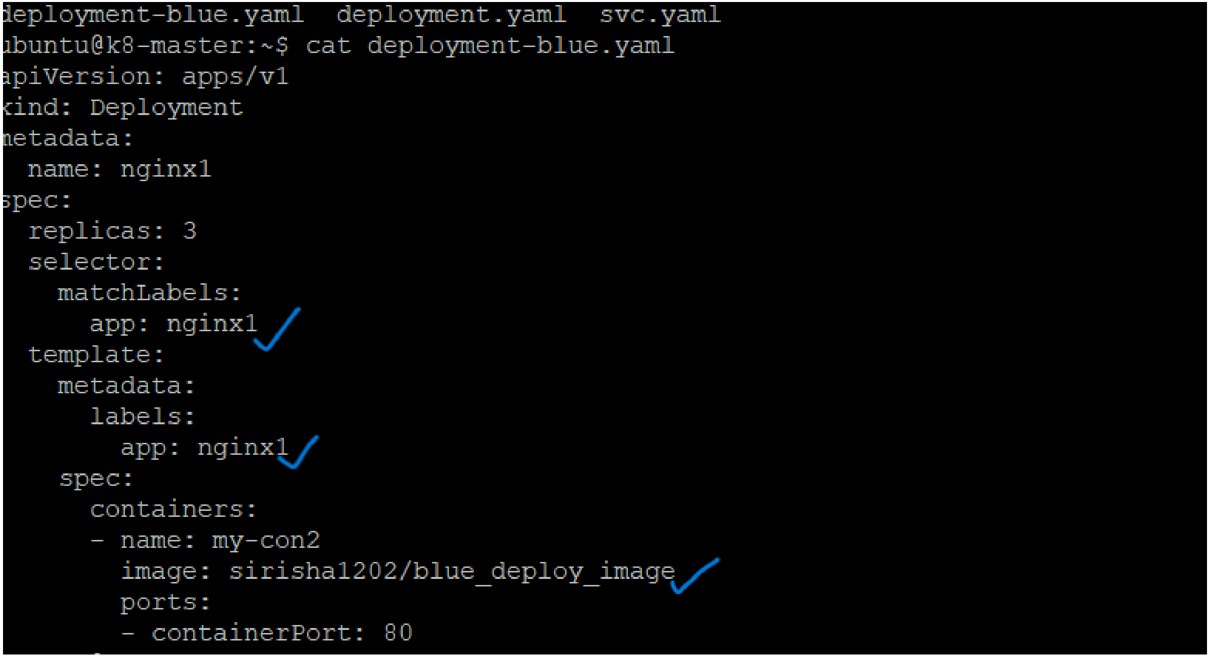
Green Deployment

I created a Deployment using the green_deploy_image with the selector app:nginx. I also created a Service with type NodePort, specifying the same selector as the Deployment. Now, I can access the application using the public IP of the server and the NodePort.

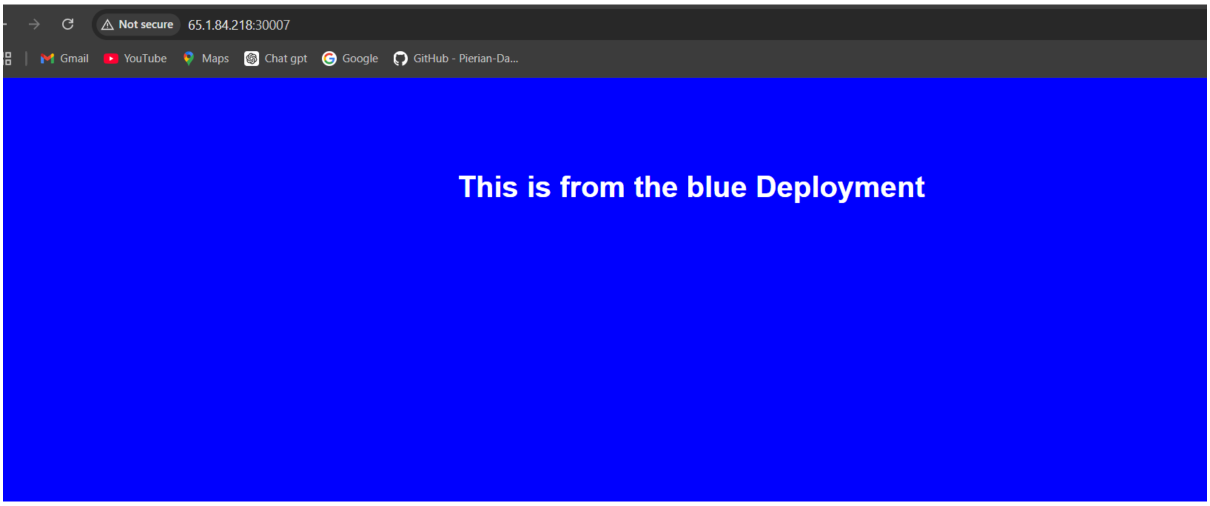
After some time, I updated the image and now I want to deploy the latest image. To do this, I'll follow these steps:



```
ubuntu@k8-master:~$ kubectl edit svc my-service
service/my-service edited
ubuntu@k8-master:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-7fcb78bd77-7fkgb   1/1     Running   0          36m
nginx-7fcb78bd77-c9wg8   1/1     Running   0          36m
nginx-7fcb78bd77-m8rcg   1/1     Running   0          36m
nginx1-965647b5c-6sg5b   1/1     Running   0          83s
nginx1-965647b5c-99x2m   1/1     Running   0          83s
nginx1-965647b5c-q6c48   1/1     Running   0          83s
ubuntu@k8-master:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1      <none>        443/TCP        4h10m
my-service   NodePort    10.98.11.115   <none>        80:30007/TCP   3h20m
ubuntu@k8-master:~$ kubectl get endpoints
NAME         ENDPOINTS                                 AGE
kubernetes   172.31.42.140:6443                        4h10m
my-service   10.36.0.3:80,10.44.0.2:80,10.44.0.3:80    3h20m
ubuntu@k8-master:~$ kubectl get pods -o wide
NAME                     READY   STATUS    RESTARTS   AGE    IP          NODE         NOMINATED NODE   READINESS GATES
nginx-7fcb78bd77-7fkgb   1/1     Running   0          37m    10.36.0.2   k8-worker2   <none>           <none>
nginx-7fcb78bd77-c9wg8   1/1     Running   0          37m    10.36.0.1   k8-worker2   <none>           <none>
nginx-7fcb78bd77-m8rcg   1/1     Running   0          37m    10.44.0.1   k8-worker1   <none>           <none>
nginx1-965647b5c-6sg5b   1/1     Running   0          112s   10.36.0.3   k8-worker2   <none>           <none>
nginx1-965647b5c-99x2m   1/1     Running   0          112s   10.44.0.3   k8-worker1   <none>           <none>
nginx1-965647b5c-q6c48   1/1     Running   0          112s   10.44.0.2   k8-worker1   <none>           <none>
ubuntu@k8-master:~$ ls
```



```
deployment-blue.yaml  deployment.yaml  svc.yaml
ubuntu@k8-master:~$ cat deployment-blue.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx1
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx1
  template:
    metadata:
      labels:
        app: nginx1
    spec:
      containers:
      - name: my-con2
        image: sirisha1202/blue_deploy_image
        ports:
        - containerPort: 80
```



This is from the blue Deployment

I created another Deployment using the updated image and updated the Service's selector to point to the new Deployment. Now, I'm able to access the latest application