

Managing Kubernetes Applications with Deployments

You are working for a company called BeeBox, a subscription service that ships weekly shipments of bees to customers. The company is using Kubernetes to run their infrastructure of containerized applications.

One of these applications is a simple web server. It is being managed in Kubernetes using a deployment called beebox-web. Unfortunately, there are some problems with the app, and it is performing poorly under large user load.

Two steps will need to be taken to fix this issue. First, you will need to deploy a newer version of the app (1.0.2) that contains some performance improvements from the developers. Second, you will need to scale the app deployment, increasing the number of replicas from 2 to 5.

.....

Update the App to a New Version of the Code

First, we need to update the container image that is being used for the pods in our deployment.

1. Edit the beebox-web deployment

```
kubectl edit deployment beebox-web
```

2. That will put us into an editor here with all of the YAML metadata for our deployment. We need to change the image version tag from 1.0.1 to 1.0.2

Figure 1-1

```
spec:
  containers:
  - image: acgorg/beebox-web:1.0.2
    imagePullPolicy: IfNotPresent
    name: web-server
  ports:
```

Above I changed the version ending in a number 2

3. Now we check the status of the deployment and watch the update occur

```
kubectl rollout status deployment.v1.apps/beebox-web
```

Figure 1-2

```
cloud_user@k8s-control:~$ kubectl rollout status deployment.v1.apps/beebox-web
deployment "beebox-web" successfully rolled out
```

This has successfully rolled out and running the kubectl get pods command shows the status running

Scale the App to a Larger Number of Replicas

So here let's verify if the version did roll out

4. After doing the kubectl get pods we need to verify if the version is correct

```
kubectl describe pod beebox-web-8467bd49fb-4s57w
```

Figure 1-3

```
Status:      Running
IP:          192.168.194.69
IPs:
  IP:        192.168.194.69
Controlled By: ReplicaSet/beebox-web-8467bd49fb
Containers:
  web-server:
    Container ID: containerd://a888bc92c06a91a52487f124516063c816f335b71858c912051
    Image:        acgorg/beebox-web:1.0.2
    Image ID:     docker.io/acgorg/beebox-web@sha256:df40267f49bedaf527490bdc99025
    Port:        80/TCP
    Host Port:    0/TCP
    State:        Running
      Started:    Mon, 21 Nov 2022 22:23:44 +0000
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-jjh7c (ro)
```

So it did update to the newest version

5. No we scaled the deployment

```
kubectl scale deployment.v1.apps/beebox-web --replicas=5
```

```
cloud_user@k8s-control:~$ kubectl get deployment beebox-web
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
beebox-web    5/5     5            5           126m
```

You can see it has already scaled up that previous. All 5 replicas are ready. Which means those 3 new replicas were created and they're already up running

We performed a rolling deployment to change the image version run by the containers in our deployment.