# **Deploying a New Version Without Downtime**

To deploy a new version without downtime I added the following strategy to each of my deployments…

**strategy**:  
 **type**: RollingUpdate  
 **rollingUpdate**:  
 **maxSurge**: 1  
 **maxUnavailable**: 0

… and then to ensure that there was no down time I added the following readinessProbe to the container spec …

**readinessProbe**:  
 **httpGet**:  
 **path**: /  
 **port**: 8080  
 **initialDelaySeconds**: 5  
 **periodSeconds**: 5  
 **successThreshold**: 1

So if we use the backend feed deployment as an example, I added a new version number to the deployment and changes the number of replicas from 2 to 4…

**metadata**:  
 **labels**:  
 **service**: reverseproxy  
 **version**: v2

and from the command line I ran …

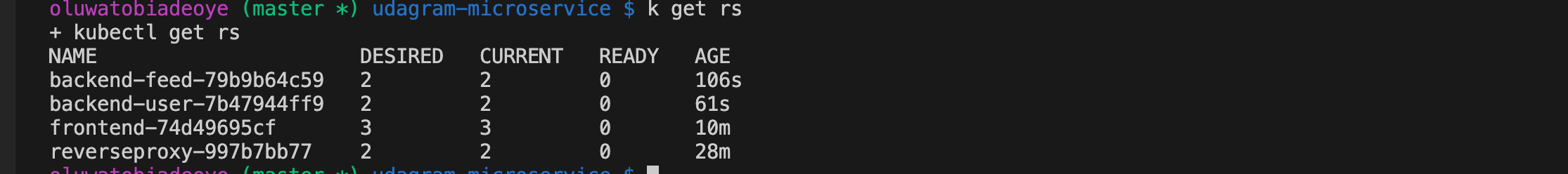
* kubectl get rs

… to show the current state of the replica sets .

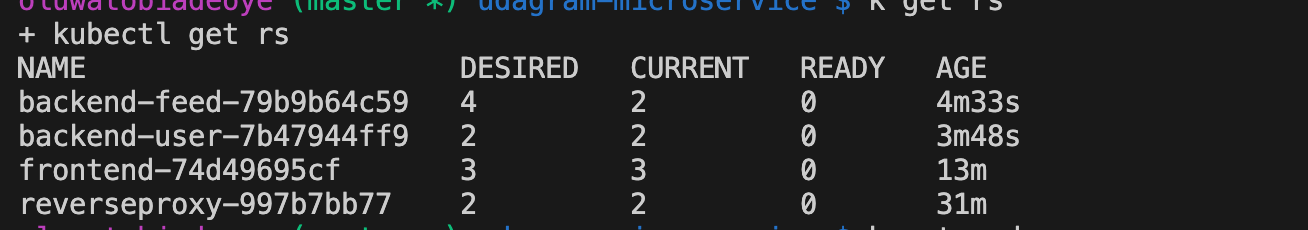
I then ran …

kubectl apply -f backend-feed-deployment.yaml

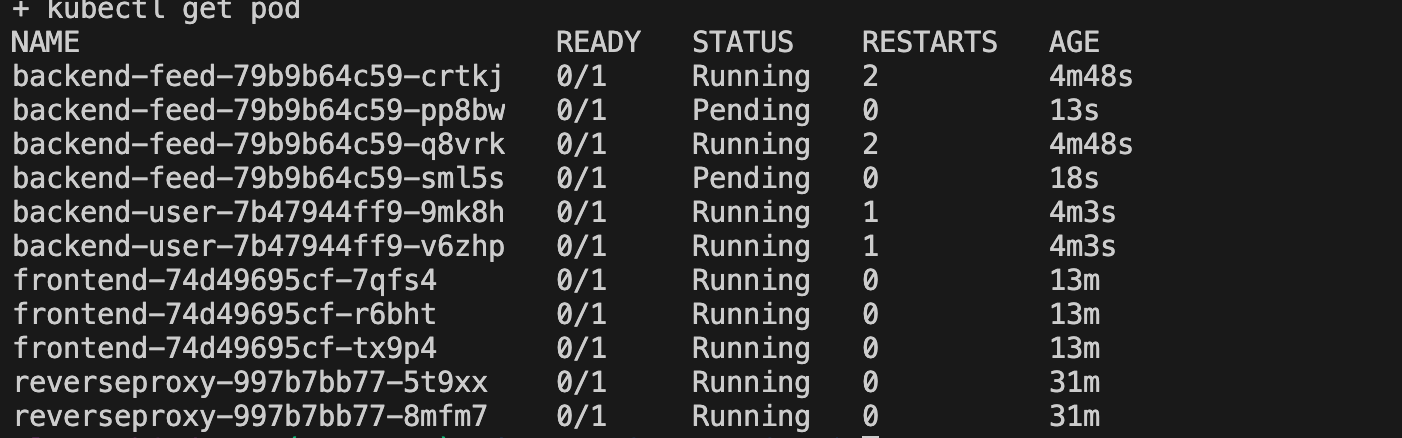
… in order to deploy the new service without downtime.



Running **kubectl get rs** now shows the old and new deployment taking place for feed. It maintains the current 2 and updates desired to 4



**Get pod** now shows two old pods and two new one coming up.



**Get pod** now shows the new deployment without any down time.

