# **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

… as per the document at …

<https://medium.com/platformer-blog/enable-rolling-updates-in-kubernetes-with-zero-downtime-31d7ec388c81>

(The document in the link had the command badly indented by the way which caused me more angst)

So if we use the backend feed deployment as an example, I added a new version number to the deployment…

**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 as follows with my feed service and 3 replicas.

A screenshot of a cell phone

Description automatically generated

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.

A screenshot of a cell phone

Description automatically generated

And **get pod** shows a new feed pod being instantiated with the old ones still in place.

A screenshot of a cell phone

Description automatically generated

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

A screenshot of a cell phone

Description automatically generated

**Get pod** now shows two new ones running and one old one and another old one terminating.

A screenshot of a cell phone

Description automatically generated

**Get pod** now shows one old one with two new ones running and a third coming on line

A screenshot of a cell phone

Description automatically generated

**Get pod** now shows the new deployment without any down time as the final old one terminates.

A screenshot of a cell phone

Description automatically generated

Finally checking the replica sets we can see we have a successful deployment.

A screenshot of a cell phone

Description automatically generated