A canary deployment consists of gradually shifting production traffic from version A to version B. Usually the traffic is split based on weight. For example, 90 percent of the requests go to version A, 10 percent go to version B.

This technique is mostly used when the tests are lacking or not reliable or if there is little confidence about the stability of the new release on the platform.

**Steps to follow**

1. 10 replicas of version 1 is serving traffic
2. deploy 1 replicas version 2 (meaning ~10% of traffic)
3. wait enought time to confirm that version 2 is stable and not throwing unexpected errors
4. scale up version 2 replicas to 10
5. wait until all instances are ready
6. shutdown version 1

**In practice**

Deploy the first application:

$ kubectl apply -f app-v1.yaml

Test if the deployment was successful:

$ curl $(minikube service my-app --url)

2018-01-28T00:22:04+01:00 - Host: host-1, Version: v1.0.0

To see the deployment in action, open a new terminal and run a watch command. It will show you a better view on the progress:

$ watch kubectl get po

Then deploy the version 2 of the application:

$ kubectl apply -f app-v2.yaml

Only one pod with the new version should be running.

You can test if the second deployment was successful:

$ service=$(minikube service my-app --url)

$ while sleep 0.1; do curl "$service"; done

If you are happy with it, scale up the version 2 to 10 replicas:

kubectl scale -n default deployment my-app --replicas=10

Then, when all pods are running, you can safely delete the old deployment:

kubectl delete deploy my-app-v1

**Cleanup**

$ kubectl delete all -l app=my-app