The ramped deployment strategy consists of slowly rolling out a version of an application by replacing instances one after the other until all the instances are rolled out. It usually follows the following process: with a pool of version A behind a load balancer, one instance of version B is deployed. When the service is ready to accept traffic, the instance is added to the pool. Then, one instance of version A is removed from the pool and shut down.

Depending on the system taking care of the ramped deployment, you can tweak the following parameters to increase the deployment time:

* Parallelism, max batch size: Number of concurrent instances to roll out.
* Max surge: How many instances to add in addition of the current amount.
* Max unavailable: Number of unavailable instances during the rolling update procedure.

**Steps to follow**

1. version 1 is serving traffic
2. deploy version 2
3. wait until all replicas are replaced with version 2

**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 the following command:

$ watch kubectl get po

Then deploy the version 2 of the application:

$ kubectl apply -f app-v2.yaml

Test the second deployment progress:

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

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

In case you discover some issue with the new version, you can undo the rollout:

$ kubectl rollout undo deploy my-app

If you can also pause the rollout if you want to run the application for a subset of users:

$ kubectl rollout pause deploy my-app

Then if you are satisfy with the result, rollout:

$ kubectl rollout resume deploy my-app

$ kubectl get rs

To see the ReplicaSet (rs) created by the Deployment

$ kubectl get pods --show-labels

**Cleanup**

$ kubectl delete all -l app=my-app