Rollback

After the image update, your colleague finds the service become unstable you may want to go back to the previous version. Unfortunately, he/she dunno how the previous config looks like. Well, you don't need the time machine, just let rollback to do its job.
At previous part, the parameter --record comes with command let the Kubernetes record the command you typed, so that you can distinguish between the revisions.

At previous part, the parameter --record comes with command let the Kubernetes record the command you typed, so that you can distinguish between the revisions.

You can use the <your-name>-deployment.yaml from Kubernetes Rollout Lab.

$ **kubectl apply -f <your-name>-deployment --record**

$ **kubectl set image deployment <your-name>-deployment <your-name>-container=<docker-hub-image>/image:<tag> --record**

Example
$ kubectl set image deployment arshad-deployment arshad-container=asyed755/delldemo:latest --record

$ **kubectl rollout history deployment <your-name>-deployment**

REVISION CHANGE-CAUSE
1 **kubectl apply -f <your-name>-deployment --record**
2 **kubectl set image deployment <your-container-name>=<docker-hub-image>/image:<new tag> --record**

Example
deployment.extensions/arshad-deployment
REVISION CHANGE-CAUSE
1 kubectl apply --filename=deploy.yaml --record=true
2 kubectl set image deployment arshad-deployment arshad-container=asyed755/delldemo:latest --record=true

Now, lets go back to revision 1

to previous revision
$ kubectl rollout undo deployment <your-name>-deployment
to specific revision
$ kubectl rollout undo deployment <your-name>-deployment --to-revision=<revision>
exmaple
$ kubectl rollout undo deployment arshad-deployment --to-revision=1

All revision history is stored in the replica sets that deployment controls. If you want to keep more revision history, please set .spec.revisionHistoryLimit in yaml to specify the number of old Replica Sets to retain to allow rollback. (set this field at the first time apply)

```
...
spec:
  replicas: 10
  selector:
```

```
    matchLabels:
      service: http-server
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  minReadySeconds: 5
  revisionHistoryLimit: 10
…
```

Example

$ kubectl rollout history deployment/arshad-deployment

```
deployment.extensions/arshad-deployment
REVISION  CHANGE-CAUSE
2         kubectl set image deployment arshad-deployment
arshad-container=asyed755/delldemo:latest --record=true
3         kubectl apply --filename=deploy.yaml --record=true
```