In order to support rolling update, we need to configure the update strategy first.

SSH to the AWS instance and go to the dir where you created the deployment object for lab 3.
$cd /home/devops/
$ curl -k https://pastebin.com/raw/7gzgrVdA **>** <your-name>-update-deployment.yaml
$ vim <your-name>-update-deployment.yaml

Replace <your-name> with your name, and update the **image:** section as well.

So we add following part into **spec and update the container image to the new version that needs to be updated on the deployments.**

```
spec:
minReadySeconds: 5
strategy:
  # indicate which strategy we want for rolling update
  type: RollingUpdate
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 1
--------
      containers:
      - name: <your-name>-container
        image: <your-docker-hub-username>/repository-name:v2 (tag)  # update the image version to v2 that was created during Docker labs
```

The final <your-name>-update-deployment.yam wiould be like the following

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: arshad-deployment
spec:
  selector:
    matchLabels:
      app: arshad-app
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 2
  minReadySeconds: 5
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: arshad-app
    spec:
      containers:
      - name: arshad-container
        image: asyed755/labs:v2 #ex : asyed755/labs:v1 - it should be the same as on docker hub
        ports:
        - containerPort: 80
~
~
```

Lets apply the new deployment.yaml

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

Now, for example, if we want to update the docker image, we have three ways to perform the rolling update.

- **set image**

Format

$ kubectl set image deployment <your-deployment-name> <your-container-name>=<new-container-v2-image> --record

Example

$ kubectl set image deployment arshad-deployment arshad-demo=asyed755/ril:v2 --record

- **Edit**

Format

$ kubectl edit deployment <deployment> --record#

Example

$ kubectl edit deployment nginx --record

This command opens the editor, and you just need to change the image version in it.

Rollout Status

$ kubectl rollout status deployment <deployment>

Pause Rolling Update

```
$ kubectl rollout pause deployment <deployment>
```

Resume Rolling Update

$ kubectl rollout resume deployment <deployment>