Rolling Updates

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

1. SSH to the AWS instance and go to the home dir.

$cd /home/devops/
$ curl -k https://pastebin.com/raw/7gzgrVdA **>** <your-name>-update-deployment.yaml
$ vim <your-name>-update-deployment.yaml

2. To replace **<your-name>** with your name
$ vim <your-name>-update-deployment.yaml

Update all the fields marked with <your-name> with your name, save and exit the vim editor (:wq)

3. Create the Deployment
$ kubectl create -f <your-name>-update-deployment.yaml **--record**

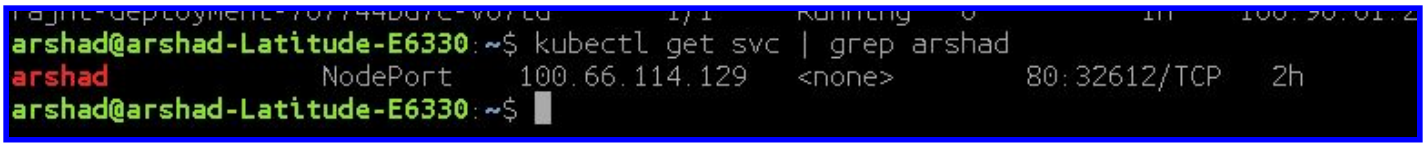
4. Expose the Deployment
$ kubectl expose deployment <your-deployment-name> --type=NodePort --port=80

Steps to Verify the rollout

$ kubectl get pods -o wide | grep <your-name>
$ kubectl get svc | grep <your-name>
Example
$ kubectl get pods -o wide | grep arshad
$ kubectl get svc | grep arshad

Node-IP screenshot

NodePort Screenshot



From the above screenshot we can notice that our app is exposed on port **32612** and is running on **ip-172-20-34-151.ec2.internal** worker node

Note the NODE-IP and NodePort (port on which app is exposed) from the output of the above commands. For example the output above says that the pod **arshad** is running on **ip-172-20-34-151.ec2.internal** and exposed on port **32612**.

This is the NODE INTERNAL IP where the POD has been scheduled.

On the AWS Console search for this IP as shown in the below screenshot

Copy the Public-IP of the Node where the pod has been deployed and try to browse the public ip on the NodePort

http://<node-pub-ip>:nodeport
Example
http://100.27.16.148:32612

Now, if we want to update the docker image, we have **two way**s to perform the rolling update.

A Set image

5. $ kubectl set image deployment <your-deployment-name> <your-container-name>=asyed755/delldemo:v1 --record

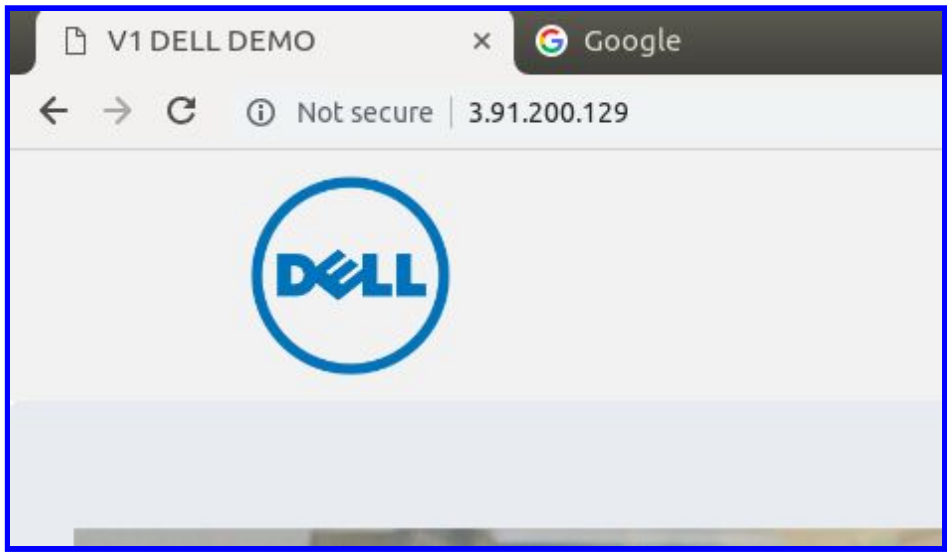
6. $ cat <your-name>-update-deployment.yaml ##to check your-deployment-name & your-container-name.

Example
$ kubectl set image deployment arshad-deployment arshad-container=asyed755/delldemo:v1 --record

You can verify that the rollout has been successful by accessing the application from the NodePort from the Workers Public-IP where the application has been deployed.

You can verify the rollout by refreshing your webpage.
http://<node-pub-ip>:nodeport
Example
http://100.27.16.148:32612

The browser should show “V1 Dell Demo” in the tab.



2.Edit

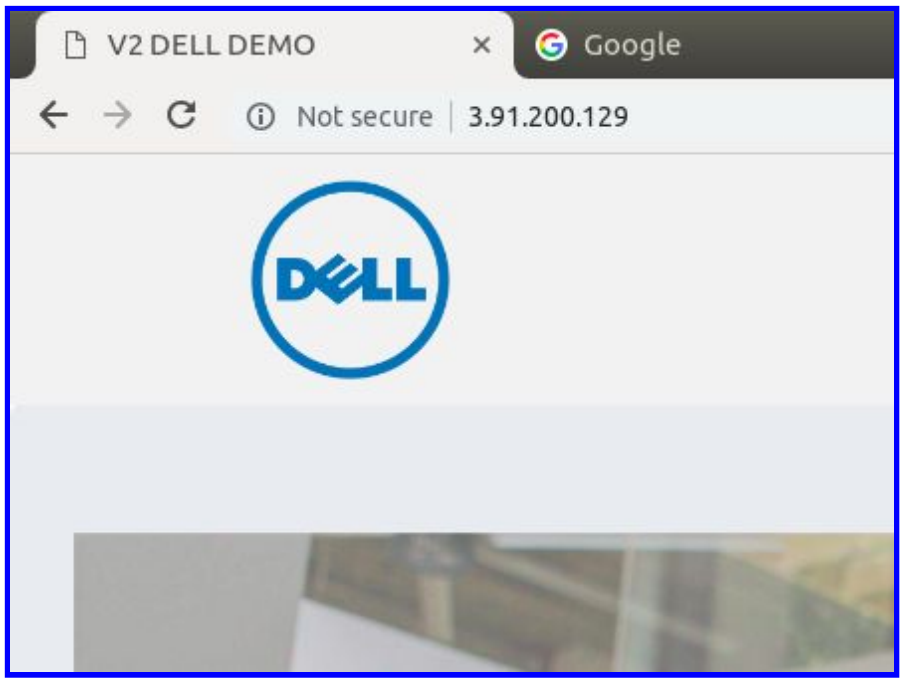
Format
$ kubectl edit deployment <your-deployment-name> --record

Example
$ kubectl edit deployment arshad-deployment --record

This command opens the **kubectl-editor**, and you need to change the image version from **v1 to v2** under “spec: containers”

You can verify the rollout by refreshing your webpage you accessed in step

http://<node-pub-ip>:nodeport
Example
http://100.27.16.148:32612



Rollout Status

$ kubectl rollout status deployment <your-deployment-name>

Pause Rolling Update

$ kubectl rollout pause deployment <your-deployment-name>

Resume Rolling Update

$ kubectl rollout resume deployment <your-deployment-name>