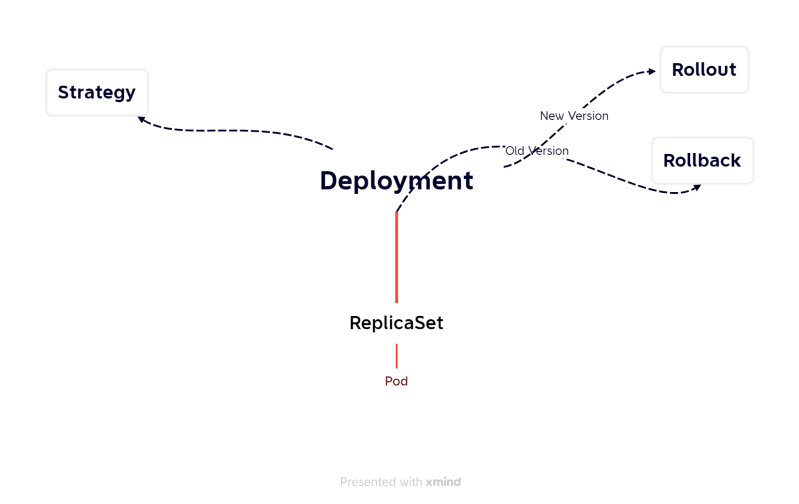
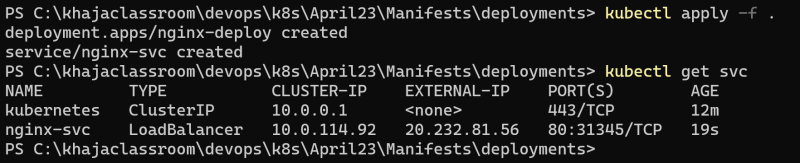
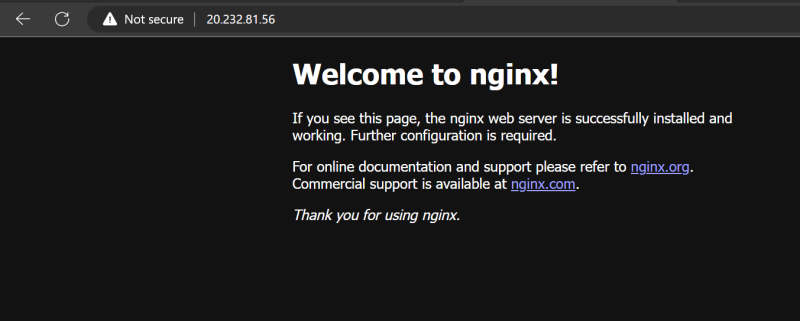
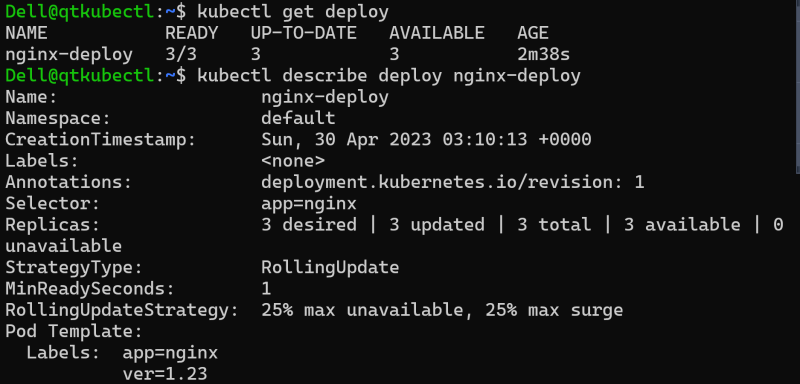
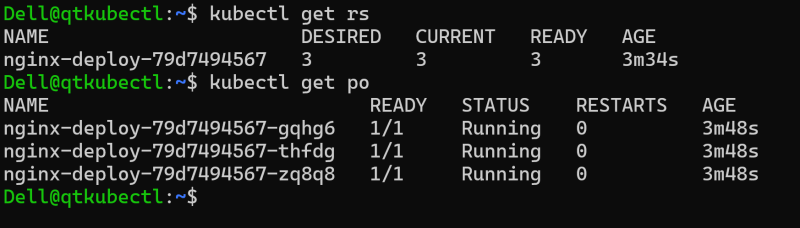
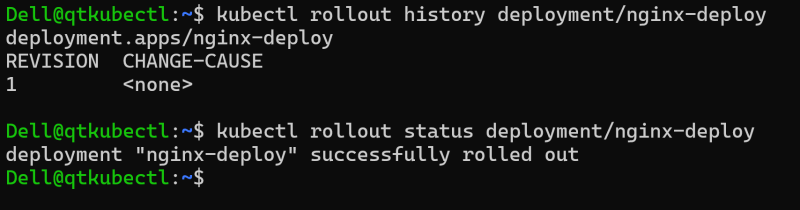
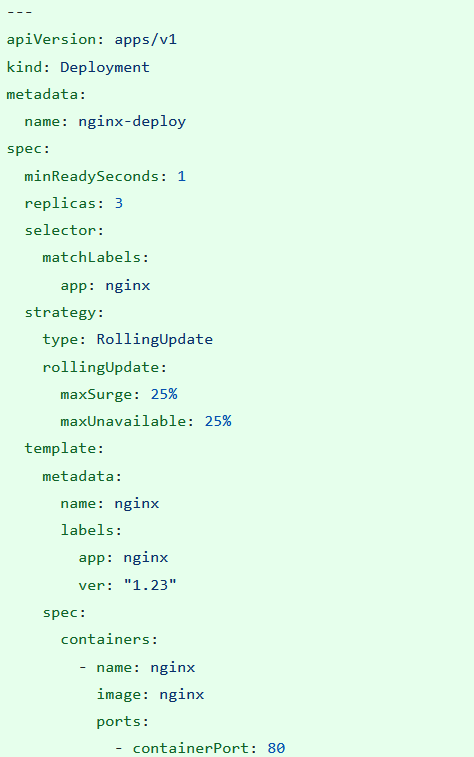
**Deployment**

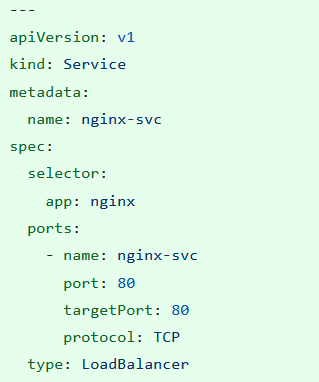
* Deployment is a k8s object which can help in rolling out and rolling back updates
* Deployment controls replica set and replica set controls pods  
  
* Let’s create a manifest with some application deployment
* apply deployment and service. Access the application  
    
  

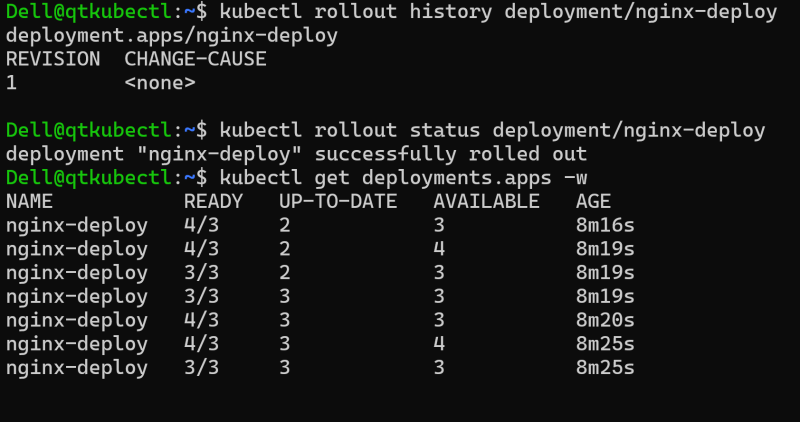
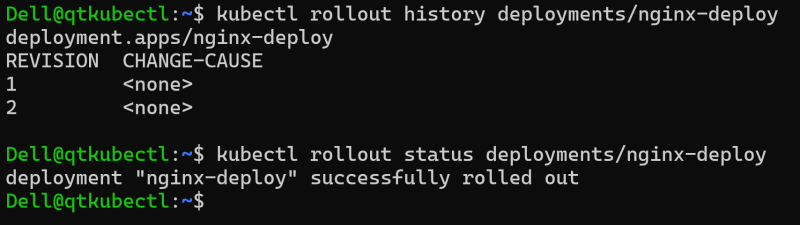
Let’s get deployment information  
  


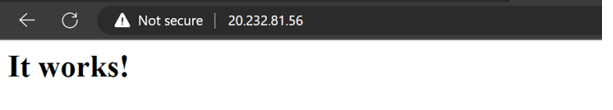
let’s explore rollout command  


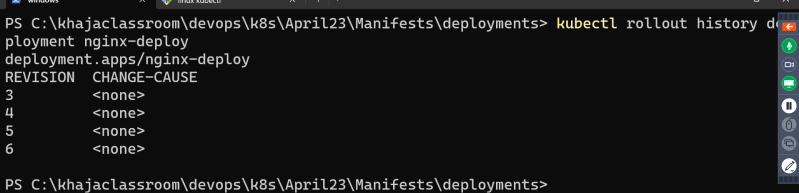
**YAML file**







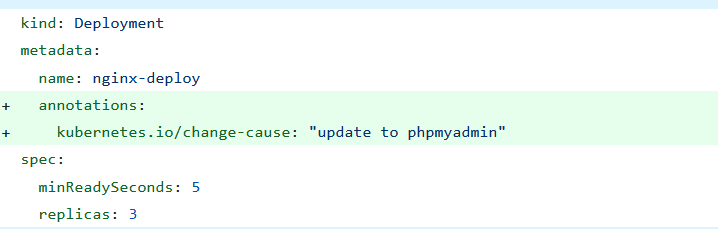
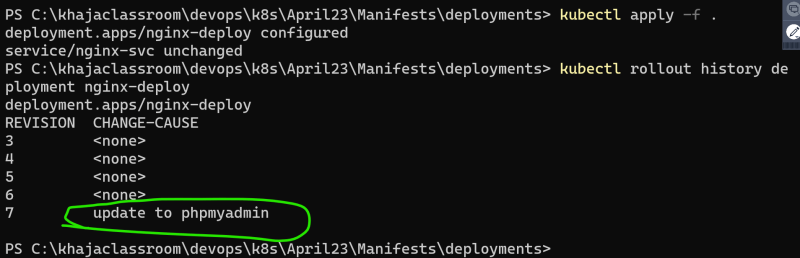
Now to rollback to previous versions and update multiple versions kubectl rollout undo  


The change-cause is showing as none which is not good. What can be done to have a valid change cause.

**Annotations**

<https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations/> for official docs

Manifest for the manifest with change cause annotation

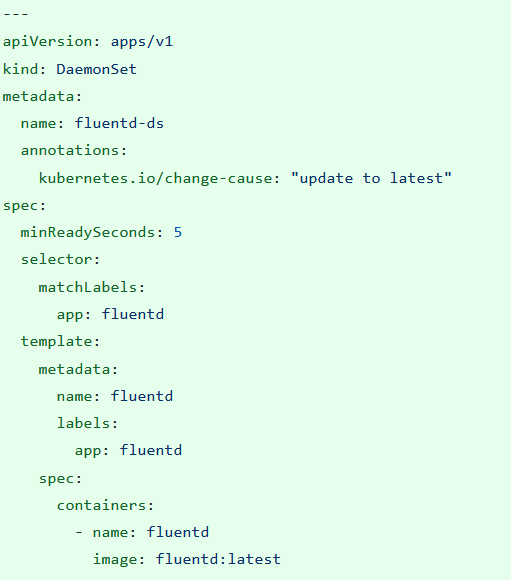
<https://azure.github.io/application-gateway-kubernetes-ingress/annotations/> for some annotations specific to azure aks ingress

<https://github.com/kubernetes-sigs/aws-load-balancer-controller/blob/main/docs/guide/ingress/annotations.md> for some annotations specific to aws eks ingress

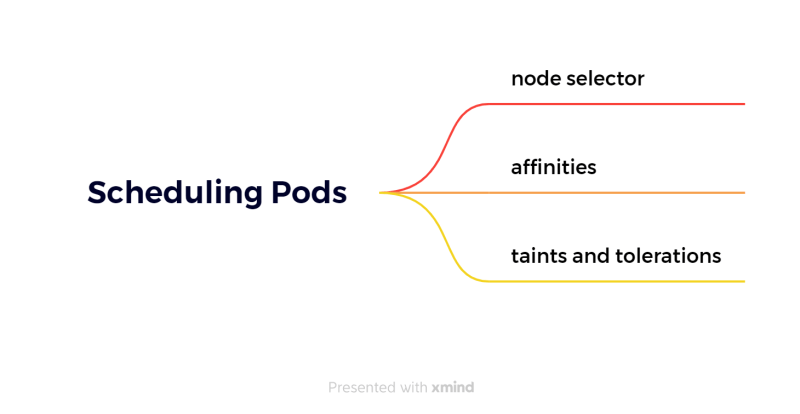
**DaemonSet**

* DaemonSet is a controller which creates pod on every/selected nodes in k8s cluster
* Use cases:
  + log collectors
  + agents etc
* <https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/> for official docs

YAML manifest for daemonset



**Scheduling Pods**

* possible ways  
  

minReadySeconds *adds latency after a Pod is Ready*

.spec.minReadySeconds is an optional field that specifies the minimum number of seconds for which a newly created Pod should be running and ready without any of its containers crashing, for it to be considered available

**Node Selectors**

* We have two nodes lets attach the following labels
  + node 0: purpose: poc
  + node 1: purpose: testing
* When we have tried to create a pod with nodeSelector matching purpose: poc it was created on node 0 and when we created a pod with purpose: testing it created in node 1 and when created a pod with purpose: development it was in pending state (not created)

---

apiVersion: v1

kind: Pod

metadata:

name: nodeselector

labels:

app: nginx

purpose: nodeselector

spec:

nodeSelector:

purpose: testing

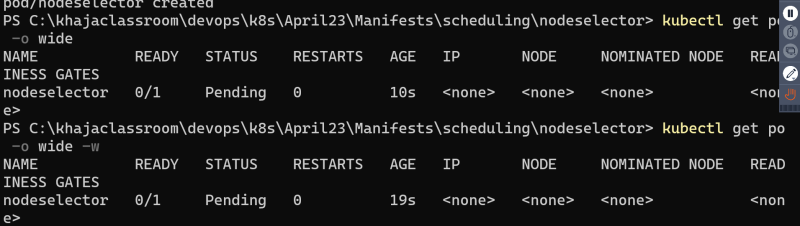
containers:

- name: jenkins

image: jenkins/jenkins:jdk11

ports:

- containerPort: 8080



**Affinity/Anti Affinity Based**

* <https://kubernetes.io/docs/tasks/configure-pod-container/assign-pods-nodes-using-node-affinity/> for official docs

**Taints and Tolerations**

* <https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/> for official docs