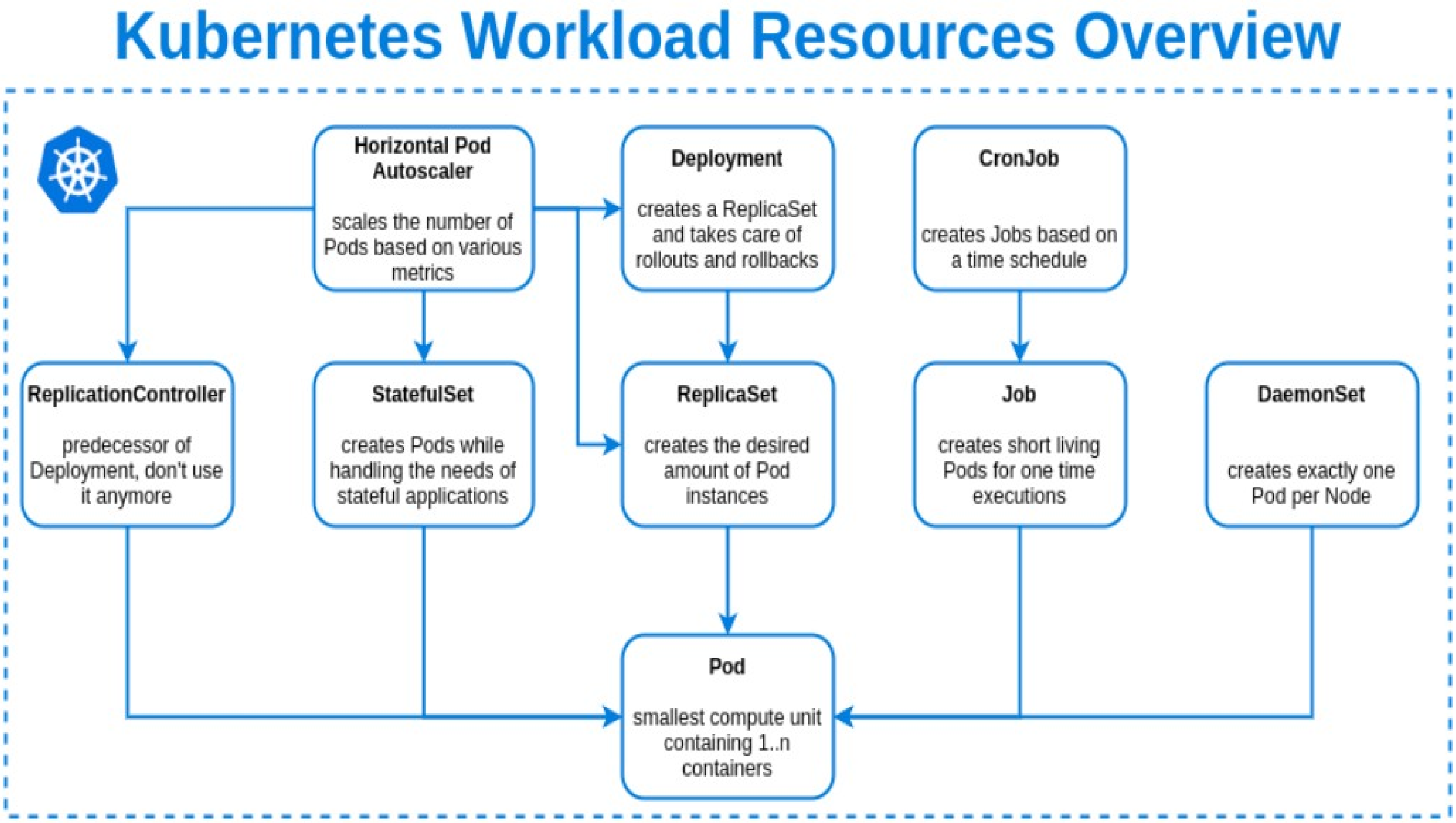
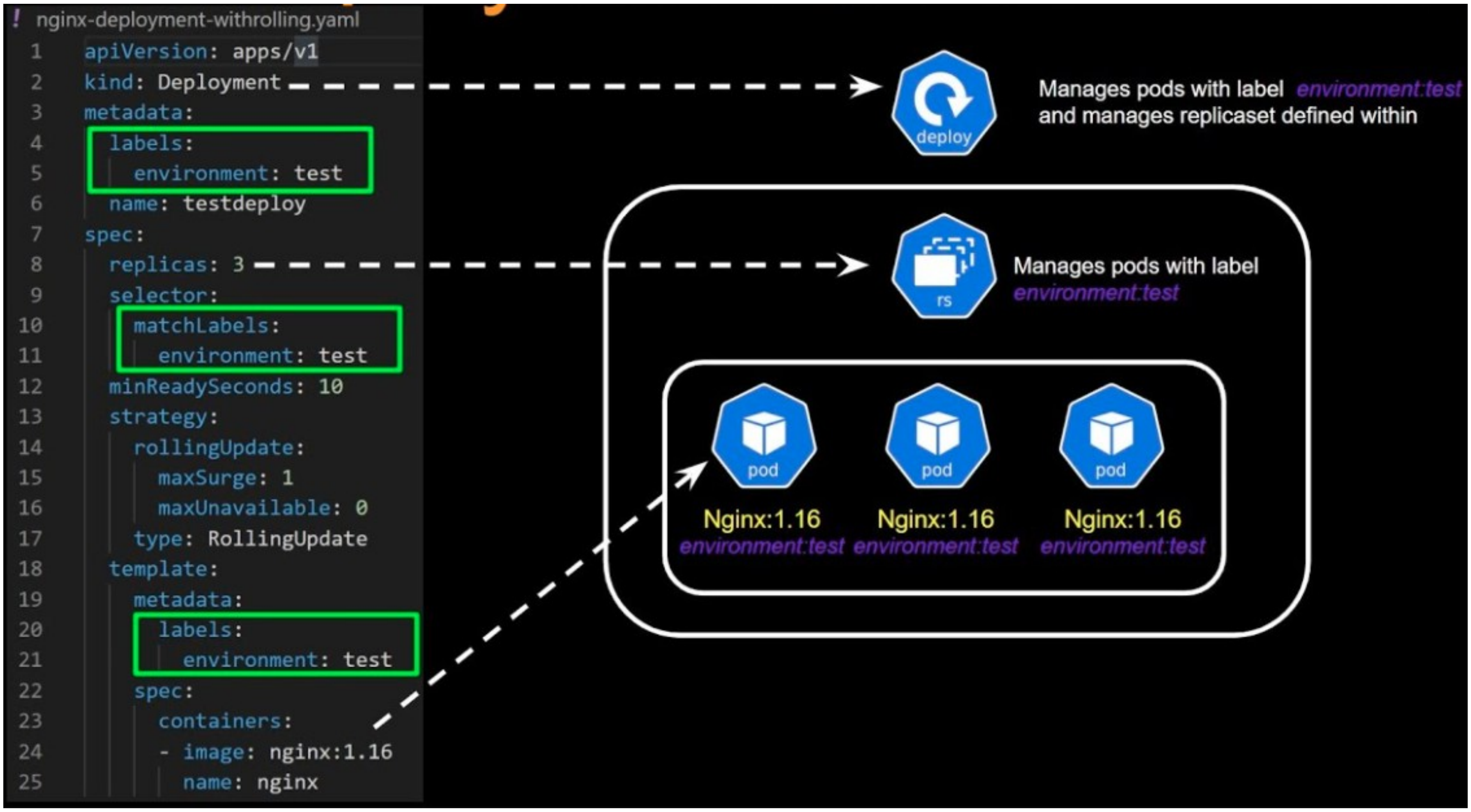
Workloads

What are Workloads?

A workload is an application running on Kubernetes. Whether you're workload is a single component or several that work together, workloads are the combination of resources that make it possible.


Kubernetes Workload Resources Overview
Horizontal Pod Autoscaler
scales the number of Pods based on various metrics
Deployment
creates a ReplicaSet and takes care of rollouts and rollbacks
CronJob
creates Jobs based on a time schedule
ReplicationController
predecessor of Deployment, don't use it anymore
StatefulSet
creates Pods while handling the needs of stateful applications
ReplicaSet
creates the desired amount of Pod instances
Job
creates short living Pods for one time executions
DaemonSet
creates exactly one Pod per Node
Pod
smallest compute unit containing 1..n containers

In Kubernetes a **deployment** is a resource object that provides declarative updates to applications. A deployment allows you to describe an application's life cycle, such as which images to use for the app, the number of pods there should be, and the way in which they should be updated. The deployment is the describes the desired state of an object.



! nginx-deployment-withrolling.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    environment: test
  name: testdeploy
spec:
  replicas: 3
  selector:
    matchLabels:
      environment: test
  minReadySeconds: 10
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 0
    type: RollingUpdate
  template:
    metadata:
      labels:
        environment: test
    spec:
      containers:
      - image: nginx:1.16
        name: nginx
```

deploy
Manages pods with label environment:test and manages replicaset defined within
rs
Manages pods with label environment:test
pod
pod
pod
Nginx:1.16 environment:test
Nginx:1.16 environment:test
Nginx:1.16 environment:test

In this example we can read the above deployment using field selectors:

- A Deployment named ‘testdeploy’ is created, indicated by the *.metadata.name* field.
- The Deployment creates three replicated Pods, indicated by the *.spec.replicas* field.
- These fields create the deployment, which then defines the object running in a pod.

A Kubernetes **object** is a way to tell the Kubernetes system how you want your cluster’s workload to look. After an object has been created, the cluster works to ensure that the object exists, maintaining the desired state of your cluster.

You can also deploy **rolling updates** incrementally replace your resource's Pods with new ones, which are then scheduled on nodes with available resources. Rolling updates are designed to update your workloads without downtime. You can specify *maxUnavailable* and *maxSurge* to control the process.

Deployment ensures that only a certain number of Pods are down while they are being updated. By default, it ensures that at least 75% of the desired number of Pods are up (25% **max unavailable**).

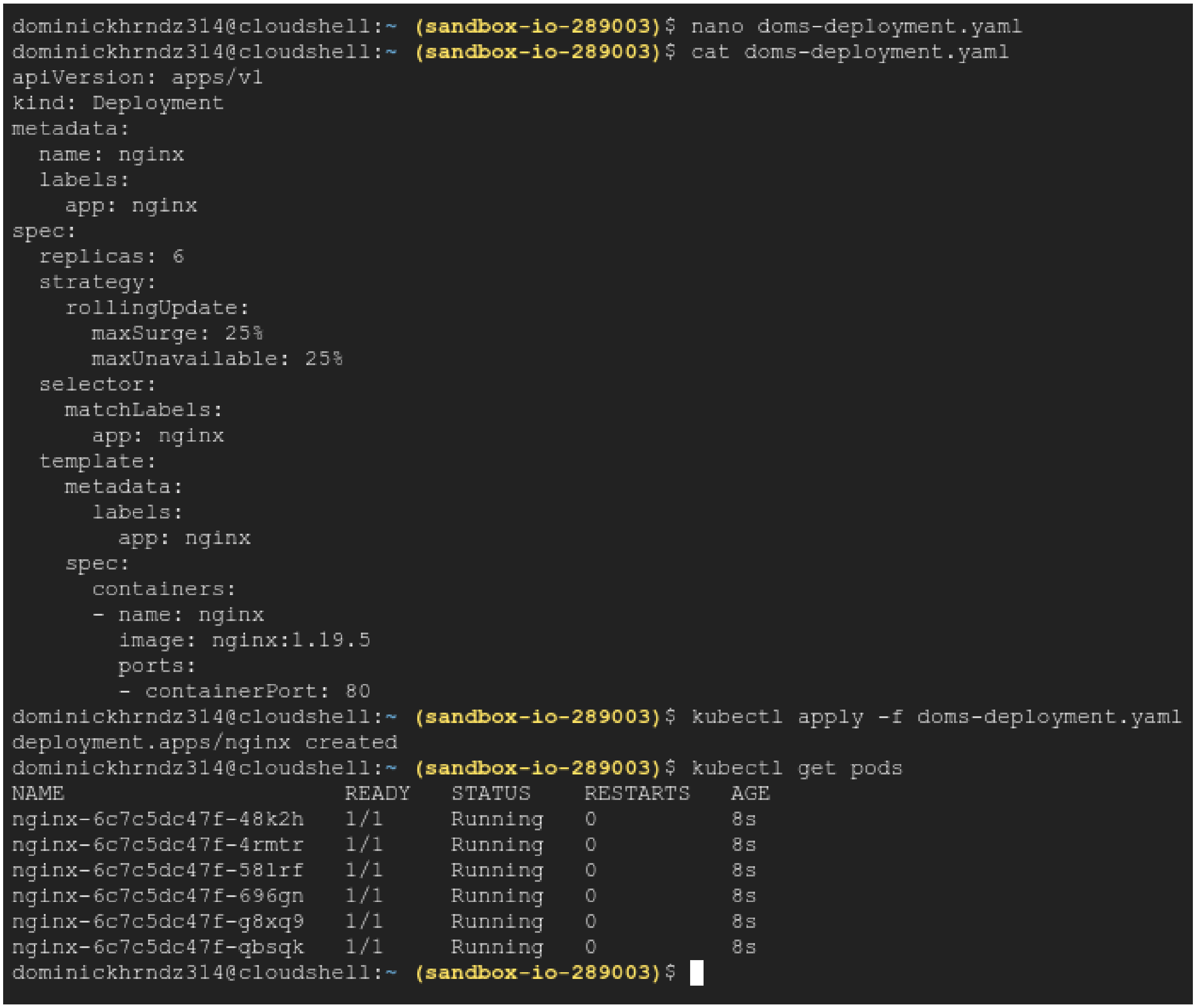
Deployment also ensures that only a certain number of Pods are created above the desired number of Pods. By default, it ensures that at most 125% of the desired number of Pods are up (25% **max surge**).

Sometimes, you may want to **rollback** a Deployment; for example, when the Deployment is not stable, such as crash looping. By default, all of the Deployment's rollout history is kept in the system so that you can rollback anytime you want.

Hands on – Create, update, and rollback a deployment.

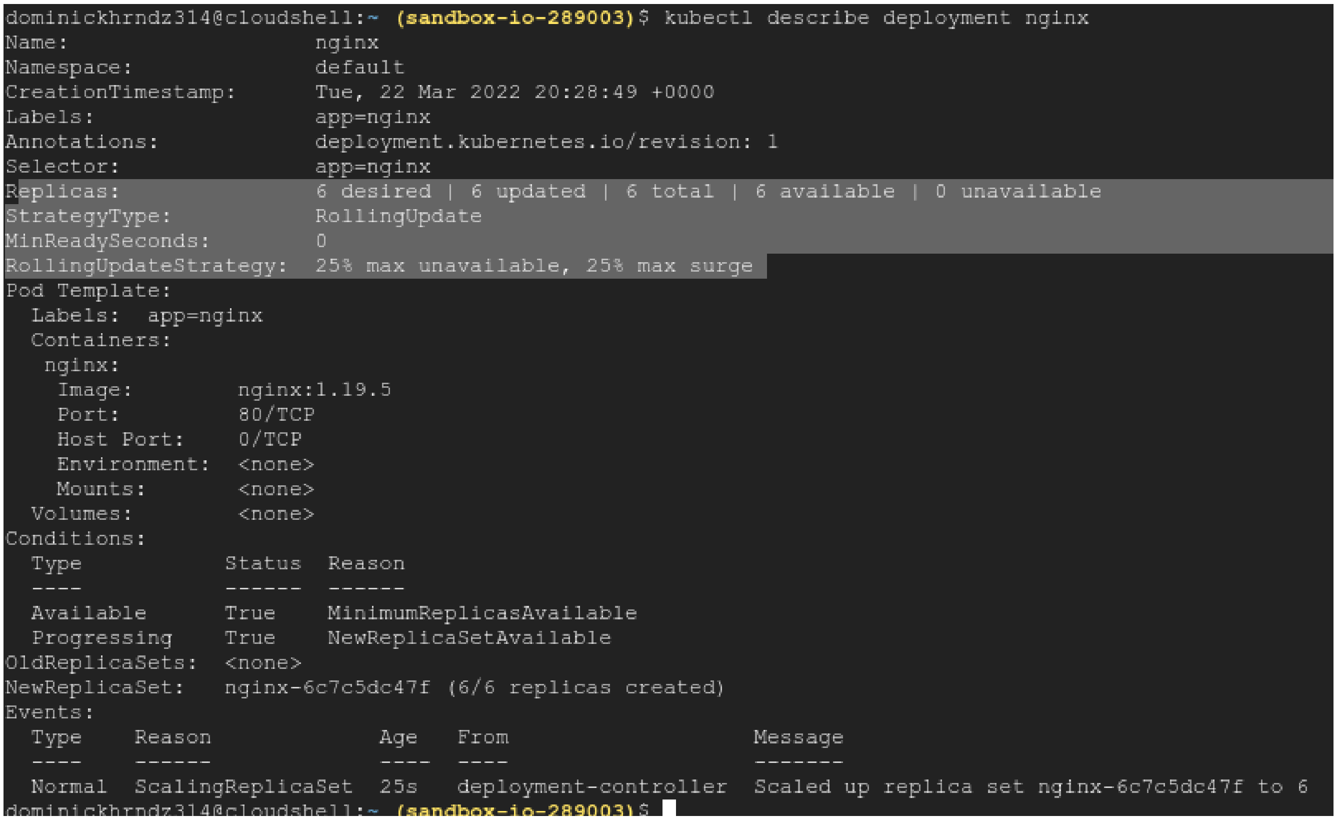
I. Create a deployment of 6 nginx instances running version 1.19.5, open to http traffic. The Deployment should also do a rolling update strategy, with a maxSurge of 50% and a maxUnavailable of 50%.

First, lets deploy our nginx pods and then verify that they are all running as expected.



```
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ nano doms-deployment.yaml
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ cat doms-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 6
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.19.5
        ports:
        - containerPort: 80
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ kubectl apply -f doms-deployment.yaml
deployment.apps/nginx created
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-6c7c5dc47f-48k2h   1/1     Running   0          8s
nginx-6c7c5dc47f-4rmtr   1/1     Running   0          8s
nginx-6c7c5dc47f-58lrf   1/1     Running   0          8s
nginx-6c7c5dc47f-696gn   1/1     Running   0          8s
nginx-6c7c5dc47f-g8xq9   1/1     Running   0          8s
nginx-6c7c5dc47f-qbsqk   1/1     Running   0          8s
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$
```

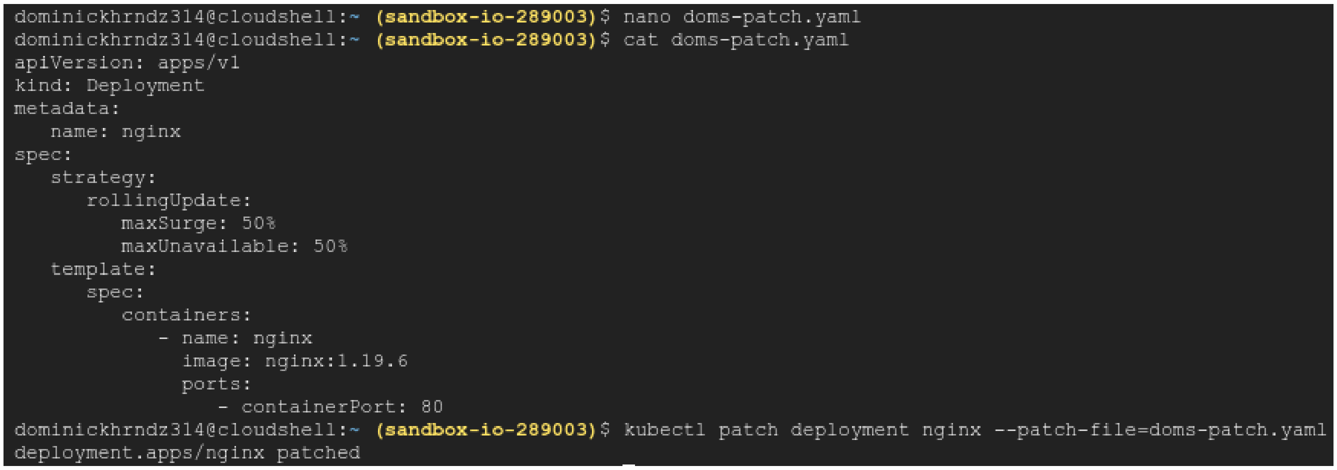
Describe the deployment and ensure you have 6 replicas, a StrategyType:RollingUpdate, and are running 25% on max surge/unavailable.



```
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ kubectl describe deployment nginx
Name:                   nginx
Namespace:              default
CreationTimestamp:      Tue, 22 Mar 2022 20:28:49 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               6 desired | 6 updated | 6 total | 6 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.19.5
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-6c7c5dc47f (6/6 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  25s   deployment-controller  Scaled up replica set nginx-6c7c5dc47f to 6
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$
```

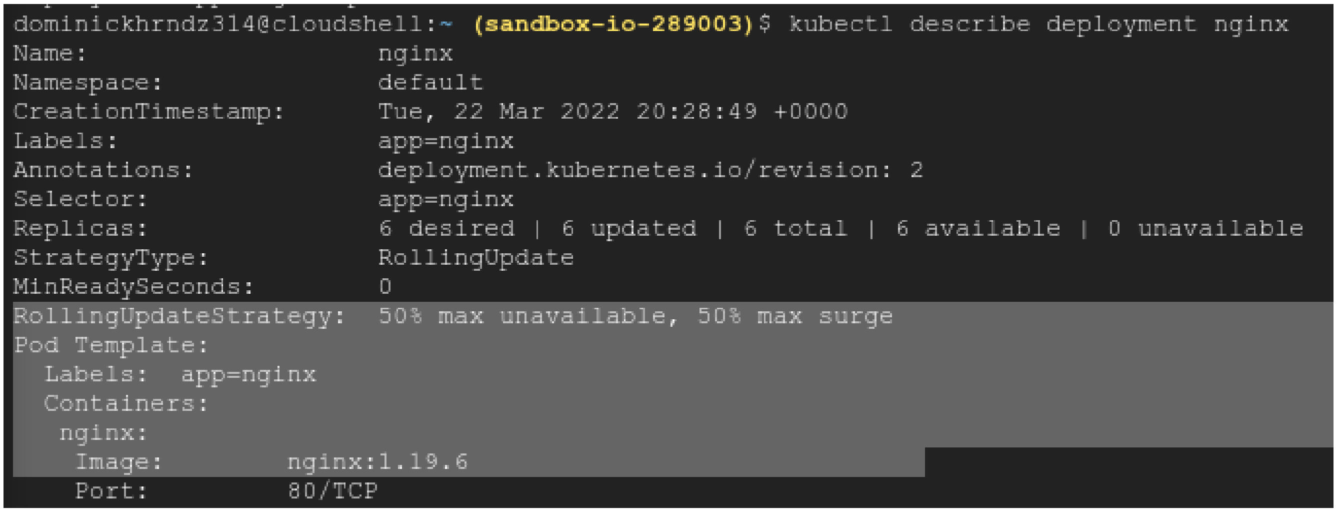
II. Update the nginx image to 1.19.6 and monitor the rollout.

To test our rolling update and roll back we need to upgrade our nginx version. Create a patch file that will update the version to 1.19.6 and will also change your maxSurge and maxUnavailable to 50%. Run kubectl patch deployment nginx –patch-file=yourname-patch.yaml to execute the update.



```
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ nano doms-patch.yaml
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ cat doms-patch.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
   name: nginx
spec:
   strategy:
      rollingUpdate:
         maxSurge: 50%
         maxUnavailable: 50%
   template:
      spec:
         containers:
            - name: nginx
              image: nginx:1.19.6
              ports:
                 - containerPort: 80
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ kubectl patch deployment nginx --patch-file=doms-patch.yaml
deployment.apps/nginx patched
```

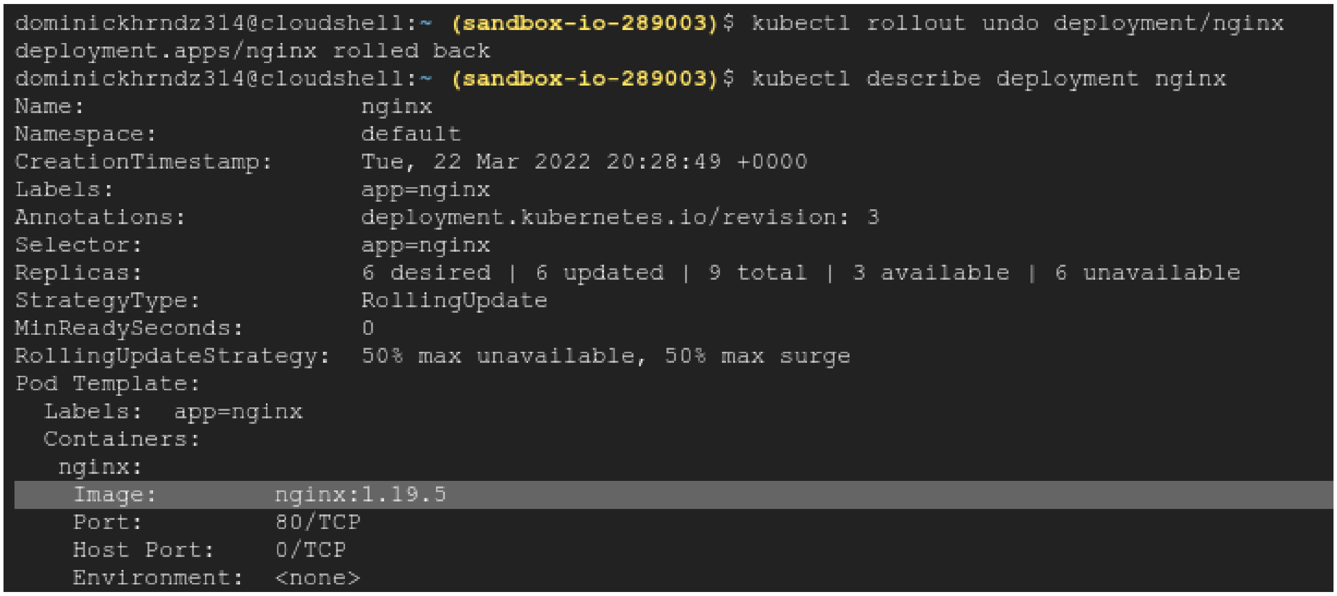
Verify that the RollingUpdate strategy has changed as well as the version of nginx.



```
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ kubectl describe deployment nginx
Name:                   nginx
Namespace:              default
CreationTimestamp:      Tue, 22 Mar 2022 20:28:49 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nginx
Replicas:               6 desired | 6 updated | 6 total | 6 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  50% max unavailable, 50% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.19.6
    Port:         80/TCP
```

III. Rollback your changes

Finally let’s rollback our changes to the previous version. Do a final kubectl describe deployment nginx to ensure you’re running 1.19.5 again.



```
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ kubectl rollout undo deployment/nginx
deployment.apps/nginx rolled back
dominickhrndz314@cloudshell:~ (sandbox-io-289003)$ kubectl describe deployment nginx
Name:                   nginx
Namespace:              default
CreationTimestamp:      Tue, 22 Mar 2022 20:28:49 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 3
Selector:               app=nginx
Replicas:               6 desired | 6 updated | 9 total | 3 available | 6 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  50% max unavailable, 50% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.19.5
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
```

If you see the following, you have completed this training.