# Deployment Updates

1. Scale the Deployment to 10 replicas in advance:

$ kubectl scale deployment nginx-deployment --replicas=10

1. Consider the following updates of container’s image along with different parameters of update strategy:
2. maxSurge: 100%, maxUnavailable: 0%, image=1.16.1
3. maxSurge: 0%, maxUnavailable: 100%, image=1.18.0
4. maxSurge: 20%, maxUnavailable: 60%, image=1.19.8

🤔 How pods will be deleted and created in every case?

1. For every case do the following:
   1. Update a deployment specification and fill the placeholders according to the case:

metadata:  
 annotations:  
 kubernetes.io/change-cause: image updated to <image value>  
spec:  
 minReadySeconds: 2  
 replicas: 10  
 strategy:  
 rollingUpdate:  
 maxSurge: <maxSurge value>  
 maxUnavailable: <maxUnvailable value>  
 template:  
 containers:  
 - image: nginx:<image value>

1. In terminal window #2 run kubectl get pods -w or kubelive get pods.
2. In terminal window #3 run kubectl get events -w
3. Apply the changes to deployment specification..
4. What’s the order of pod update? How do they removed and created?
5. Leave deployment as is. Don’t delete it. We will use a chain of updates in the next practice.

## Solution

Scale the Deployment to 10 replicas in advance:

$ kubectl scale deployment nginx-deployment --replicas=10  
deployment.apps/nginx-deployment scaled

Prepare specification for updates:

apiVersion: apps/v1  
 kind: Deployment  
 metadata:  
 name: nginx-deployment  
+ annotations:  
+ kubernetes.io/change-cause: image updated to <image value>  
 spec:  
+ minReadySeconds: 2  
+ replicas: 10  
+ strategy:  
+ rollingUpdate:  
+ maxSurge: <maxSurge value>  
+ maxUnavailable: <maxUnvailable value>  
 selector:  
 matchLabels:  
 app: nginx  
 template:  
 metadata:  
 labels:  
 app: nginx  
 spec:  
 containers:  
 - name: nginx  
+ image: nginx:<image value>  
 imagePullPolicy: IfNotPresent  
 ports:  
 - containerPort: 80

Case a) maxSurge: 100%, maxUnavailable: 0%, image=1.16.1

# deploy-update/nginx-deploy-custom1.yaml  
 apiVersion: apps/v1  
 kind: Deployment  
 metadata:  
 name: nginx-deployment  
 annotations:  
- kubernetes.io/change-cause: image updated to <image value>  
+ kubernetes.io/change-cause: image updated to 1.16.1  
 spec:  
 minReadySeconds: 2  
 replicas: 10  
 strategy:  
 rollingUpdate:  
- maxSurge: <maxSurge value>  
- maxUnavailable: <maxUnvailable value>  
+ maxSurge: 100%  
+ maxUnavailable: 0%  
 selector:  
 matchLabels:  
 app: nginx  
 template:  
 metadata:  
 labels:  
 app: nginx  
 spec:  
 containers:  
 - name: nginx  
- image: nginx:<image value>  
+ image: nginx:1.16.1  
 imagePullPolicy: IfNotPresent  
 ports:  
 - containerPort: 80

It starts new 10 pods (100%) and when any new one has started, an old one has terminated:

$ kubectl describe deploy nginx-deployment  
...  
 Normal ScalingReplicaSet 28s deployment-controller Scaled up replica set nginx-deployment-559d658b74 to 10  
 Normal ScalingReplicaSet 18s deployment-controller Scaled down replica set nginx-deployment-66b6c48dd5 to 8  
 Normal ScalingReplicaSet 17s deployment-controller Scaled down replica set nginx-deployment-66b6c48dd5 to 5  
 Normal ScalingReplicaSet 15s deployment-controller Scaled down replica set nginx-deployment-66b6c48dd5 to 0

Case b) maxSurge: 0%, maxUnavailable: 100%, image=1.18.0

# deploy-update/nginx-deploy-custom2.yaml  
 apiVersion: apps/v1  
 kind: Deployment  
 metadata:  
 name: nginx-deployment  
 annotations:  
- kubernetes.io/change-cause: image updated to 1.16.1  
+ kubernetes.io/change-cause: image updated to 1.18.0  
 spec:  
 minReadySeconds: 2  
 replicas: 10  
 strategy:  
 rollingUpdate:  
- maxSurge: 100%  
- maxUnavailable: 0%  
+ maxSurge: 0%  
+ maxUnavailable: 100%  
 selector:  
 matchLabels:  
 app: nginx  
 template:  
 metadata:  
 labels:  
 app: nginx  
 spec:  
 containers:  
 - name: nginx  
- image: nginx:1.16.1  
+ image: nginx:1.18.0  
 imagePullPolicy: IfNotPresent  
 ports:  
 - containerPort: 80

It terminates all old pods and while they are terminating, Kubernetes start creating new ones:

$ kubectl describe deploy nginx-deployment  
...  
 Normal ScalingReplicaSet 5m31s deployment-controller Scaled down replica set nginx-deployment-559d658b74 to 0  
 Normal ScalingReplicaSet 5m31s deployment-controller Scaled up replica set nginx-deployment-67dfd6c8f9 to 10

Case c) maxSurge: 20%, maxUnavailable: 60%, image=1.19.8

# deploy-update/nginx-deploy-custom3.yaml  
 apiVersion: apps/v1  
 kind: Deployment  
 metadata:  
 name: nginx-deployment  
 annotations:  
- kubernetes.io/change-cause: image updated to 1.18.0  
+ kubernetes.io/change-cause: image updated to 1.19.6  
 spec:  
 minReadySeconds: 2  
 replicas: 10  
 strategy:  
 rollingUpdate:  
- maxSurge: 0%  
- maxUnavailable: 100%  
+ maxSurge: 20%  
+ maxUnavailable: 20%  
 selector:  
 matchLabels:  
 app: nginx  
 template:  
 metadata:  
 labels:  
 app: nginx  
 spec:  
 containers:  
 - name: nginx  
- image: nginx:1.18.0  
+ image: nginx:1.19.6  
 imagePullPolicy: IfNotPresent  
 ports:  
 - containerPort: 80

It creates 2 new pods, then delete 2 old pods:

$ kubectl describe deploy nginx-deployment  
...  
 Normal ScalingReplicaSet 26s deployment-controller Scaled up replica set nginx-deployment-76ccf9dd9d to 2  
 Normal ScalingReplicaSet 26s deployment-controller Scaled down replica set nginx-deployment-67dfd6c8f9 to 8  
 Normal ScalingReplicaSet 26s deployment-controller Scaled up replica set nginx-deployment-76ccf9dd9d to 4  
 Normal ScalingReplicaSet 20s deployment-controller Scaled down replica set nginx-deployment-67dfd6c8f9 to 4  
 Normal ScalingReplicaSet 20s deployment-controller Scaled up replica set nginx-deployment-76ccf9dd9d to 8  
 Normal ScalingReplicaSet 12s deployment-controller Scaled down replica set nginx-deployment-67dfd6c8f9 to 1  
 Normal ScalingReplicaSet 12s deployment-controller Scaled up replica set nginx-deployment-76ccf9dd9d to 10  
 Normal ScalingReplicaSet 11s deployment-controller Scaled down replica set nginx-deployment-67dfd6c8f9 to 0

Also useful to see results using rollout status:

$ kubectl rollout status deploy/nginx-deployment  
Waiting for deployment "nginx-deployment" rollout to finish: 4 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 4 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 4 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 4 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 4 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 5 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 5 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 5 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 5 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 5 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 7 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 7 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 7 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 9 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 9 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 9 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 9 out of 10 new replicas have been updated...  
Waiting for deployment "nginx-deployment" rollout to finish: 1 old replicas are pending termination...  
Waiting for deployment "nginx-deployment" rollout to finish: 1 old replicas are pending termination...  
Waiting for deployment "nginx-deployment" rollout to finish: 1 old replicas are pending termination...  
Waiting for deployment "nginx-deployment" rollout to finish: 1 old replicas are pending termination...  
Waiting for deployment "nginx-deployment" rollout to finish: 8 of 10 updated replicas are available...  
Waiting for deployment "nginx-deployment" rollout to finish: 9 of 10 updated replicas are available...  
deployment "nginx-deployment" successfully rolled out