# Übung: Rolling Update

In dieser Übung erstellen wir mehrere Pods ab dem gleichen Image mit jeweils einem ReplicaSet, Deployment, Service und **Ingress**.

Das passiert in einer eigenen Namespace um die Resultate gezielt Darstellen zu können:

In [1]:

```
! kubectl create namespace depl
```

```
namespace/depl created
```

Wir Erzeugen den Pod, ReplicaSet, Deployment, Service und Ingress mittels einer YAML Datei.

In [2]:

```
! wget https://raw.githubusercontent.com/mc-b/misegr/master/bpmn/bpmn-frontend.yaml >/d
ev/null 2>&1
! cat bpmn-frontend.yaml
```

```
apiVersion: v1
kind: Service
metadata:
  name: bpmn-frontend
  labels:
    app: bpmn-frontend
    group: web
    tier: frontend
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
  selector:
    app: bpmn-frontend
---
apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: bpmn-frontend
spec:
  replicas: 5
  selector:
    matchLabels:
      app: bpmn-frontend
  template:
    metadata:
      labels:
        app: bpmn-frontend
        group: web
        tier: frontend
    spec:
      containers:
      - name: bpmn-frontend
        image: misegr/bpmn-frontend:latest
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 80
          name: bpmn-frontend
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: bpmn-frontend
  labels:
    app: bpmn-frontend
    tier: frontend
spec:
  rules:
  - http:
      paths:
      - path: /frontend
        backend:
          serviceName: bpmn-frontend
          servicePort: 80
```

In [3]:

```
! kubectl apply -f https://raw.githubusercontent.com/mc-b/misegr/master/bpmn/bpmn-frontend.yaml --namespace depl
```

```
service/bpmn-frontend created
deployment.apps/bpmn-frontend created
ingress.extensions/bpmn-frontend created
```

Ausgabe der Erzeugten Ergebnisse und die YAML Datei welche den Erzeugten Ressourcen beschreibt.

Ab `spec.containers` kommt erst der Pod.

In [4]:

```
! kubectl get pod,deployment,replicaset,service,ingress --namespace depl
```

```
NAME                                    READY   STATUS              RESTARTS
AGE
pod/bpmn-frontend-746b849978-dwh8b      0/1     ContainerCreating   0
2s
pod/bpmn-frontend-746b849978-fttnb      0/1     ContainerCreating   0
2s
pod/bpmn-frontend-746b849978-jg5jw      0/1     ContainerCreating   0
2s
pod/bpmn-frontend-746b849978-qg9rh      1/1     Running             0
2s
pod/bpmn-frontend-746b849978-sbzwq      0/1     ContainerCreating   0
2s

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.extensions/bpmn-frontend      1/5     5            1           2s

NAME                                               DESIRED   CURRENT   READY
AGE
replicaset.extensions/bpmn-frontend-746b849978     5         5         1
2s

NAME                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
AGE
service/bpmn-frontend    ClusterIP   10.109.140.139   <none>        80/TCP
3s

NAME                                HOSTS   ADDRESS   PORTS   AGE
ingress.extensions/bpmn-frontend    *                 80      2s
```

Schauen wir uns zuerst die Ausgabe des laufenden Pods an, diesmal über den URL des Kubernetes API Servers und den via Ingress Ressourcen angehängten Prefix.

Der Prefix `frontend` ist fix in der Ingress Ressource hinterlegt, `index.html` ergibt sich wie der Container aufgebaut ist (Apache Server mit einer HTML Datei).

In [5]:

```
! echo $(kubectl config view -o=jsonpath='{ .clusters[0].cluster.server }' | sed -e "s/6443/30443/")/frontend/index.html
```

```
https://192.168.137.100:30443/frontend/index.html
```

Wir wollen jedoch nicht die letzte ( `latest` ) Version von `bpmn-frontend` sondern die Version `V1.0` , deshalb führen wir einen Rolling Update durch, bzw. Ändern die Versionsnummer hinter dem Imagenamen.

In [6]:

```
! kubectl set image deployment/bpmn-frontend bpmn-frontend=misegr/bpmn-frontend:V1.0 --
namespace depl
```

```
deployment.extensions/bpmn-frontend image updated
```

Die Änderungen können wir uns Anzeigen lassen:

In [7]:

```
! kubectl describe deployment/bpmn-frontend --namespace depl
```

```
Name:                   bpmn-frontend
Namespace:              depl
CreationTimestamp:      Fri, 04 Oct 2019 11:59:19 +0000
Labels:                 app=bpmn-frontend
                        group=web
                        tier=frontend
Annotations:            deployment.kubernetes.io/revision: 2
                        kubectl.kubernetes.io/last-applied-configuration:
                          {"apiVersion":"apps/v1beta2","kind":"Deploymen
t","metadata":{"annotations":{},"name":"bpmn-frontend","namespace":"dep
l"},"spec":{"replicas...
Selector:               app=bpmn-frontend
Replicas:               5 desired | 3 updated | 7 total | 5 available | 2
unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=bpmn-frontend
           group=web
           tier=frontend
  Containers:
   bpmn-frontend:
    Image:        misegr/bpmn-frontend:V1.0
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    ReplicaSetUpdated
OldReplicaSets:  bpmn-frontend-746b849978 (3/3 replicas created)
NewReplicaSet:   bpmn-frontend-578dccbf9d (3/4 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  7s    deployment-controller  Scaled up replic
a set bpmn-frontend-746b849978 to 5
  Normal  ScalingReplicaSet  3s    deployment-controller  Scaled up replic
a set bpmn-frontend-578dccbf9d to 2
  Normal  ScalingReplicaSet  3s    deployment-controller  Scaled down repl
ica set bpmn-frontend-746b849978 to 4
  Normal  ScalingReplicaSet  3s    deployment-controller  Scaled up replic
a set bpmn-frontend-578dccbf9d to 3
  Normal  ScalingReplicaSet  1s    deployment-controller  Scaled down repl
ica set bpmn-frontend-746b849978 to 3
  Normal  ScalingReplicaSet  1s    deployment-controller  Scaled up replic
a set bpmn-frontend-578dccbf9d to 4
```

Neu sollte im Titel die Versionnummer `V1.0` angezeigt werden.

In [8]:

```
! echo $(kubectl config view -o=jsonpath='{ .clusters[0].cluster.server }' | sed -e "s/6443/30443/")/frontend/index.html
```

```
https://192.168.137.100:30443/frontend/index.html
```

Zum Aufräumen genügt es den Namespace zu löschen

# RollOut

Wenn die neue Version der Software nicht wie erwartet funktoniert, können wir zur vorherigen Version zurückkehren.

Dies ist möglich, weil Kubernetes den Rollout-Verlauf von Deployment in Form von Revisionen speichert .

In [9]:

```
! kubectl rollout history deployment/bpmn-frontend --namespace depl
```

```
deployment.extensions/bpmn-frontend
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

Es sollten zwei Revisionen angezeigt werden. Die aktuelle ist die Revision 2.

Um auf die vorherige Revision zurück zu kehren, verwenden wir:

In [10]:

```
! kubectl rollout undo deployment/bpmn-frontend --namespace depl
```

```
deployment.extensions/bpmn-frontend rolled back
```

In [11]:

```
! kubectl rollout history deployment/bpmn-frontend --namespace depl
```

```
deployment.extensions/bpmn-frontend
REVISION  CHANGE-CAUSE
2         <none>
3         <none>
```

Aufräumen

In [ ]:

```
! kubectl delete namespace depl
```

```
namespace "depl" deleted
```

In [ ]:

```
```