

Subtask 3 demo

The task is to deploy new changes to song-service using Rolling Update.

First I've added `genre` field to response of GET endpoint of song-service:

```
public class SongMetadata {  
    4 usages  
    private String title;  
    4 usages  
    private String artist;  
    4 usages  
    private String album;  
    4 usages  
    private String duration;  
    4 usages  
    private String genre;  
    4 usages  
    private int resourceId;  
}
```

and pushed it to docker hub with tag `2.0.0`.

Then I updated song-service-deployment manifest:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: song-service-deployment
  namespace: my
spec:
  replicas: 2
  selector:
    matchLabels:
      app: song-service
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
  template:
```

and applied it.

Here what I had:

The screenshot shows a code editor with a file named `kubernetes-song-service.yml`. The manifest defines a Deployment for `song-service` in the `my` namespace. It uses a `RollingUpdate` strategy with `maxSurge: 1`. The container is `song-service-container` using the image `robertgaz/song-service:latest`. Below the manifest, a terminal window shows the output of `kubectl get pod -n my`.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: song-service-deployment
  namespace: my
spec:
  replicas: 2
  selector:
    matchLabels:
      app: song-service
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
  template:
    metadata:
      labels:
        app: song-service
    spec:
      containers:
      - name: song-service-container
        image: robertgaz/song-service:latest
```

Document 5/6 > spec: > template: > spec: > containers: > Item 1/1 > name: > song-service-contain...

Terminal: Local x Local (2) x Local (3) x + v

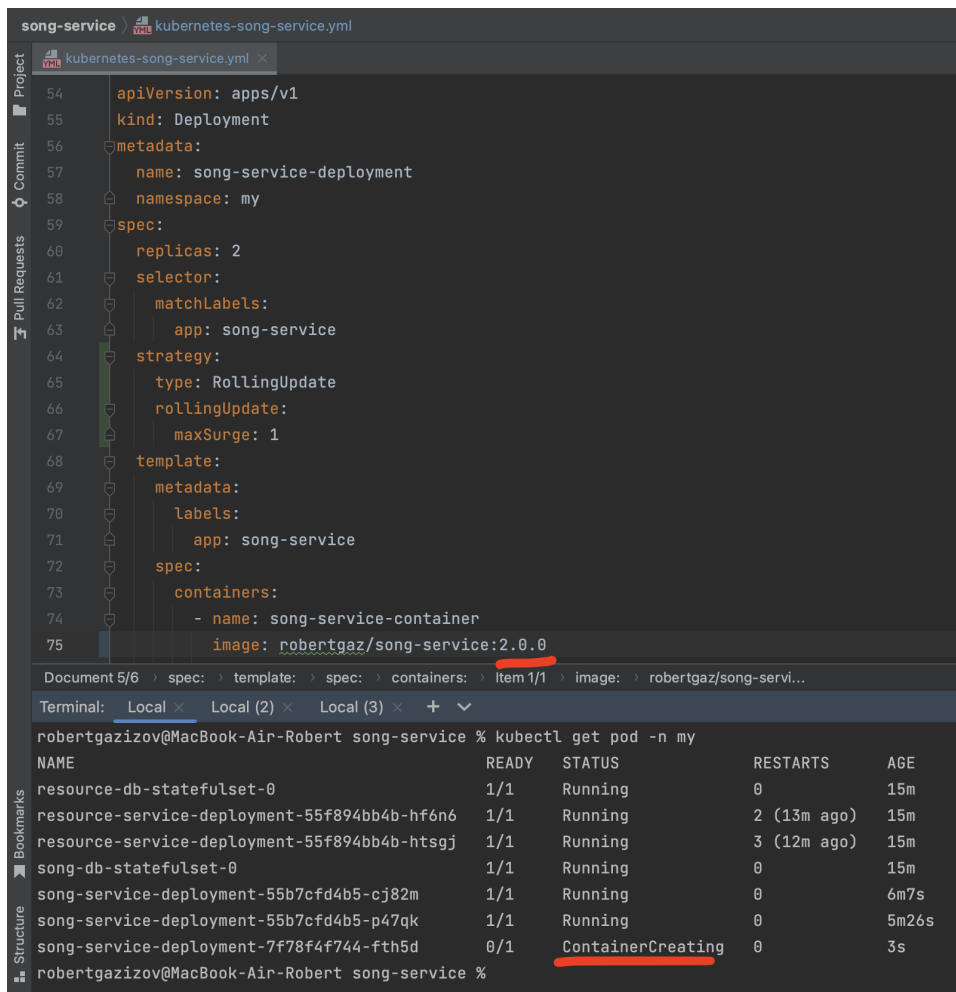
```
robertgazizov@MacBook-Air-Robert song-service % kubectl get pod -n my
NAME                                READY   STATUS    RESTARTS   AGE
resource-db-statefulset-0           1/1     Running   0           14m
resource-service-deployment-55f894bb4b-hf6n6  1/1     Running   2 (11m ago)  14m
resource-service-deployment-55f894bb4b-htsgj  1/1     Running   3 (10m ago)  14m
song-db-statefulset-0               1/1     Running   0           13m
song-service-deployment-55b7cfd4b5-cj82m     1/1     Running   0           4m27s
song-service-deployment-55b7cfd4b5-p47qk     1/1     Running   0           3m46s
robertgazizov@MacBook-Air-Robert song-service %
```

Then I updated tag of container in manifest:

This snippet shows the `template` section of the manifest after the update. The `image` field for `song-service-container` has been changed from `latest` to `2.0.0`.

```
template:
  metadata:
    labels:
      app: song-service
  spec:
    containers:
    - name: song-service-container
      image: robertgaz/song-service:2.0.0
      env:
```

Then I applied it and saw that kubernetes started to schedule new pods with updated container:



The screenshot shows a code editor with a file named `kubernetes-song-service.yml`. The file contains a Kubernetes Deployment configuration. The `image` field in the container specification is highlighted in red and underlined, showing `robertgaz/song-service:2.0.0`.

Below the code editor, a terminal window is open, showing the output of the command `kubectl get pod -n my`. The output is a table with columns: NAME, READY, STATUS, RESTARTS, and AGE.

NAME	READY	STATUS	RESTARTS	AGE
resource-db-statefulset-0	1/1	Running	0	15m
resource-service-deployment-55f894bb4b-hf6n6	1/1	Running	2 (13m ago)	15m
resource-service-deployment-55f894bb4b-htsgj	1/1	Running	3 (12m ago)	15m
song-db-statefulset-0	1/1	Running	0	15m
song-service-deployment-55b7cfd4b5-cj82m	1/1	Running	0	6m7s
song-service-deployment-55b7cfd4b5-p47qk	1/1	Running	0	5m26s
song-service-deployment-7f78f4f744-fth5d	0/1	ContainerCreating	0	3s


Finally, all pods were replaced:

The screenshot shows an IDE with a file named `kubernetes-song-service.yml` open. The manifest is a Deployment for `song-service` in the `my` namespace, using `image: robertgaz/song-service:2.0.0`. Below the manifest, a terminal window shows the command `kubectl get pod -n my` and its output, which lists several pods, including `song-service-deployment-7f78f4f744-fth5d` and `song-service-deployment-7f78f4f744-nkddz`, both in a `Running` state.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: song-service-deployment
  namespace: my
spec:
  replicas: 2
  selector:
    matchLabels:
      app: song-service
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
  template:
    metadata:
      labels:
        app: song-service
    spec:
      containers:
        - name: song-service-container
          image: robertgaz/song-service:2.0.0
          env:
```

```
robertgazizov@MacBook-Air-Robert song-service % kubectl get pod -n my
NAME                                     READY   STATUS    RESTARTS   AGE
resource-db-statefulset-0               1/1     Running   0          16m
resource-service-deployment-55f894bb4b-hf6n6  1/1     Running   2 (14m ago)  16m
resource-service-deployment-55f894bb4b-htsgj  1/1     Running   3 (13m ago)  16m
song-db-statefulset-0                   1/1     Running   0          16m
song-service-deployment-7f78f4f744-fth5d    1/1     Running   0          74s
song-service-deployment-7f78f4f744-nkddz    1/1     Running   0          53s
robertgazizov@MacBook-Air-Robert song-service %
```

And I were able to see `genre` field in the response from song-service GET endpoint:

 <http://127.0.0.1:50265/songs/1>

GET

▼

<http://127.0.0.1:50265/songs/2>

Send

▼

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

Cookies

Headers (5)

Test Results

200 OK

30 ms

276 B

Save Response ▼


Pretty



Raw

Preview

Visualize

JSON ▼



1

2

3

4

5

6

7

8

2

"title": null,

"artist": "My2007",

"album": "Orlanda",

"duration": "306.54290771484375",

"genre": "rock",

"resourceId": 2

3