



JAKUB KRZYWDA
@jakubkrzywda



How to perform **BLUE-GREEN DEPLOYMENTS** using only Kubernetes Primitives?

Cloud Native Technologies



basic level

Blue-green deployment refers to running **two** application **environments** in parallel **in a production** cluster.

The first environment (**blue**) is running the **stable** application version and the second environment (**green**) is running the **new** version.



By **default**, Kubernetes performs a **rolling update** of a deployment.

The old version is **replaced** by the new one during the rollout.

However, in case of some applications we want to keep the old version “**on stand-by**” for a while after the new rollout.



Luckily, it is **possible** to
perform **blue-green**
deployments using only
Kubernetes primitives!

Here I show you how to do it
in five easy steps:



1.

Create a Blue Deployment

```
replicas: 3  
...  
labels:  
  app: myapp  
  track: blue  
image: myapp:v1
```



1.

Create a Blue Deployment

Notice two labels,
app and track, their
values and the
image tag.

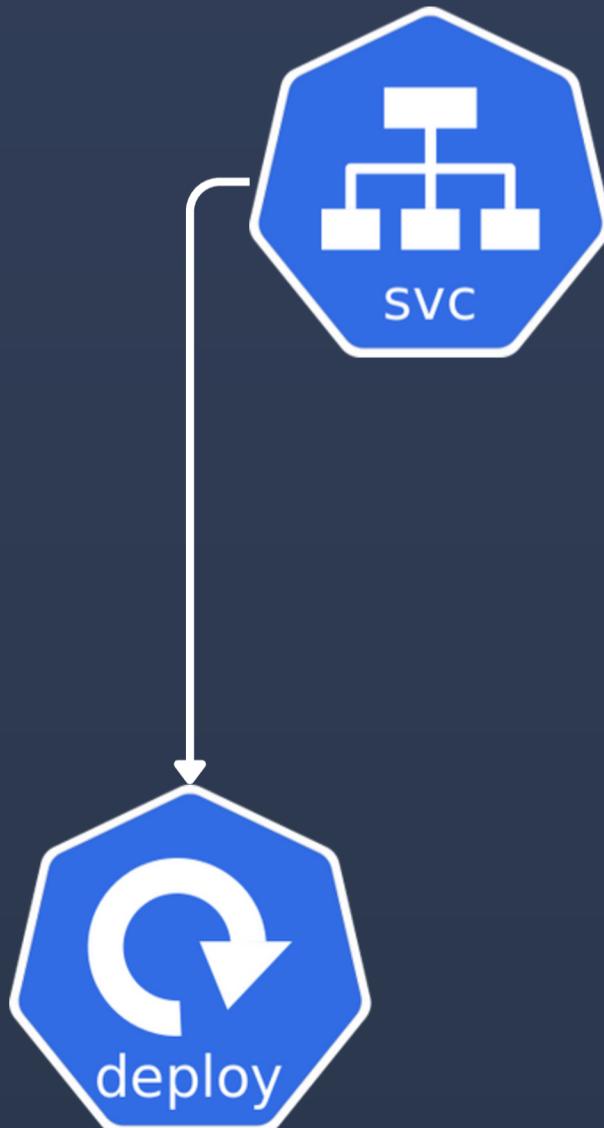
```
replicas: 3
...
labels:
  app: myapp
  track: blue
image: myapp:v1
```



2.

Expose it with a Service

```
replicas: 3  
...  
labels:  
  app: myapp  
  track: blue  
image: myapp:v1
```



```
name: myservice  
selector:  
  app: myapp  
  track: blue
```

2.

Expose it with a Service

Service selector uses both labels – app and track. Therefore, it precisely matches the blue deployment!



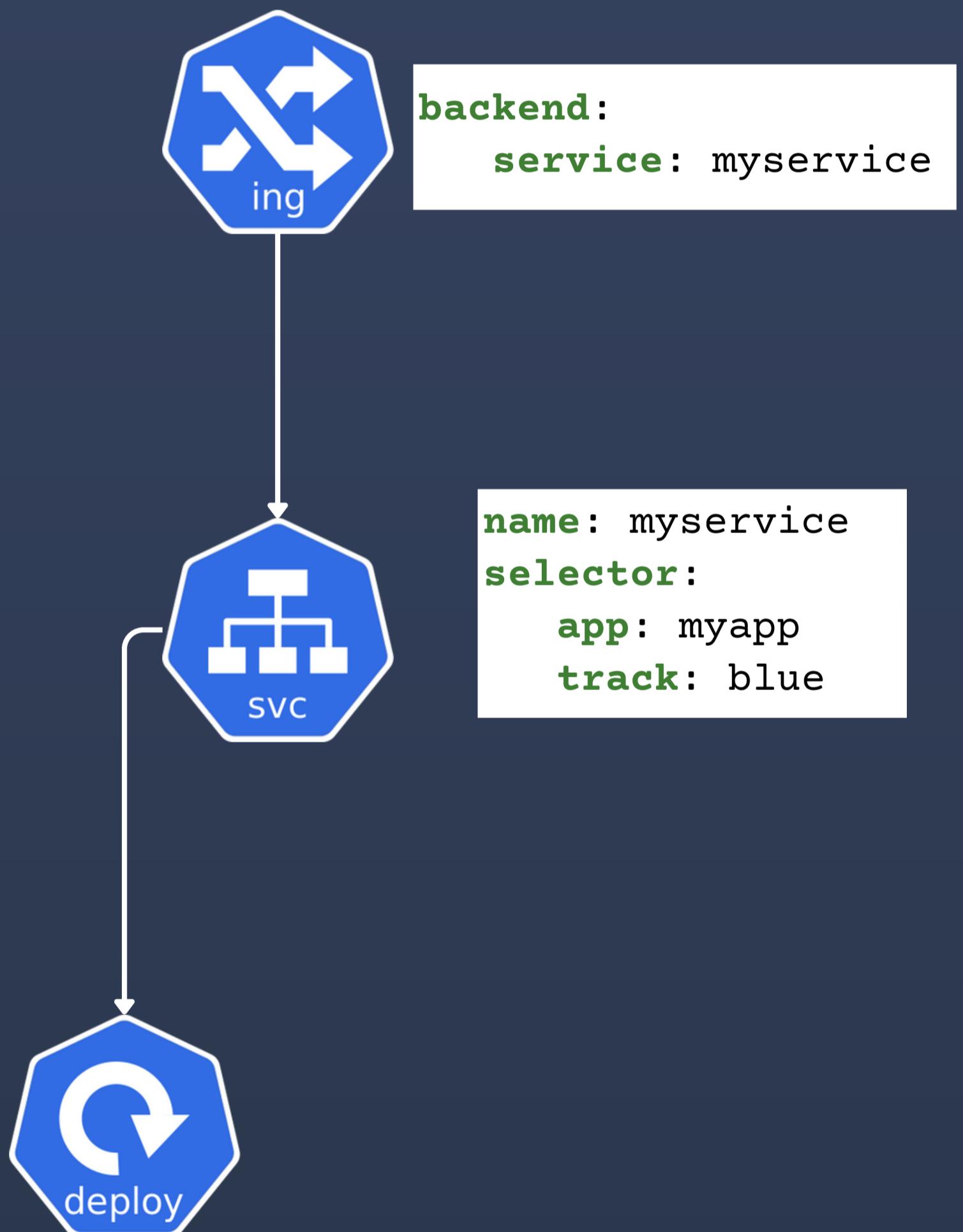
```
name: myservice  
selector:  
  app: myapp  
  track: blue
```

```
replicas: 3  
...  
labels:  
  app: myapp  
  track: blue  
image: myapp:v1
```



3.

Add an Ingress



3.

Add an Ingress

Ingress is optional
but useful to expose
application outside the
cluster.



```
backend:  
  service: myservice
```



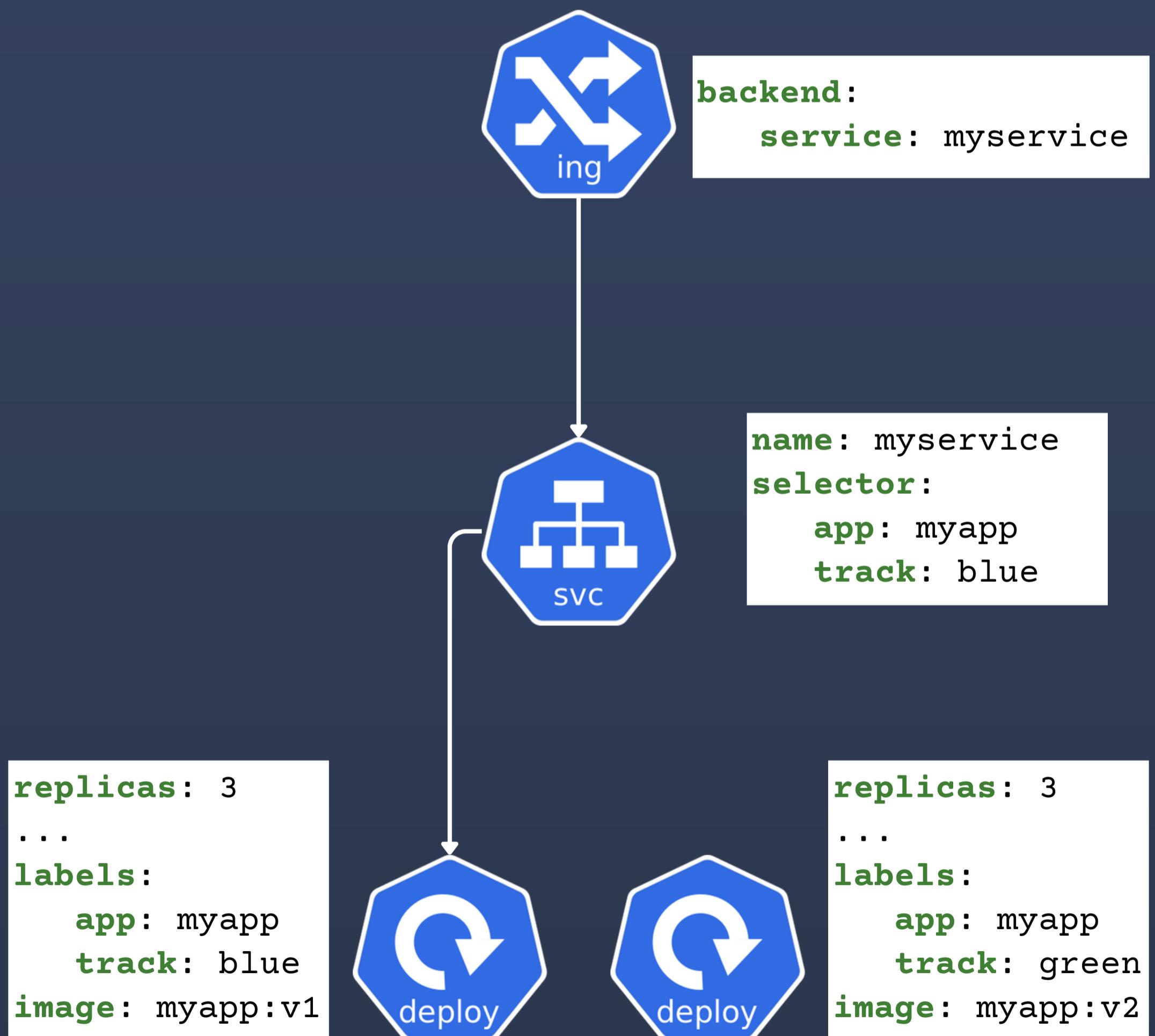
```
name: myservice  
selector:  
  app: myapp  
  track: blue
```

```
replicas: 3  
...  
labels:  
  app: myapp  
  track: blue  
image: myapp:v1
```



4.

Add a Green Deployment



4.

Add a Green Deployment



```
backend:  
  service: myservice
```

Notice different track labels and image tags.
The number of replicas is identical since the green deployment should take over the whole workload.

```
replicas: 3  
...  
labels:  
  app: myapp  
  track: blue  
image: myapp:v1
```



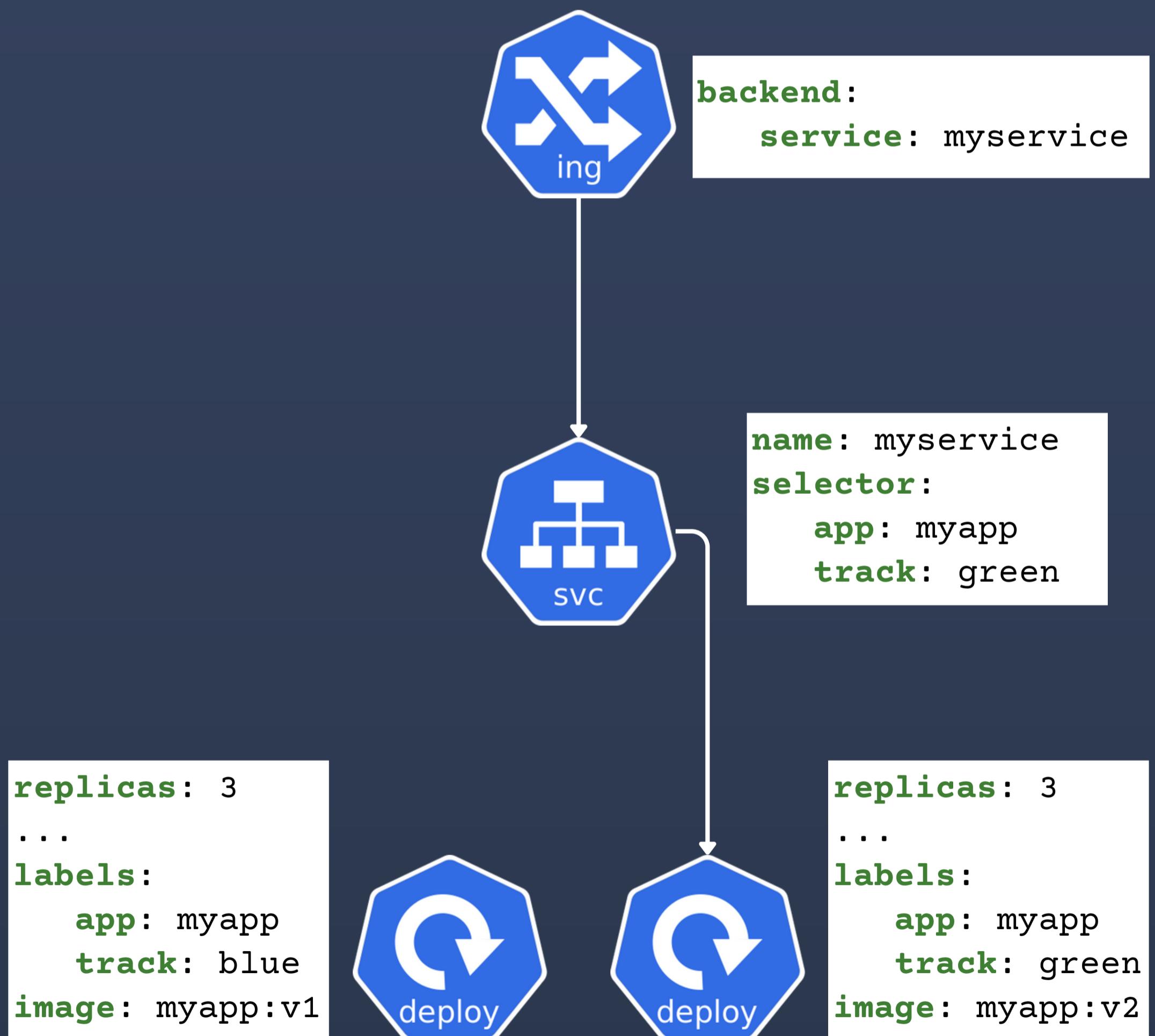
```
name: myservice  
selector:  
  app: myapp  
  track: blue
```



```
replicas: 3  
...  
labels:  
  app: myapp  
  track: green  
image: myapp:v2
```

5.

Switch to Green Deployment



5.

Switch to Green Deployment

Notice a new value of the track service selector – green. Therefore, the service matches only the green deployment now!

```
replicas: 3  
...  
labels:  
  app: myapp  
  track: blue  
image: myapp:v1
```



```
backend:  
  service: myservice
```



```
name: myservice  
selector:  
  app: myapp  
  track: green
```



```
replicas: 3  
...  
labels:  
  app: myapp  
  track: green  
image: myapp:v2
```



JAKUB KRZYWDA

@jakubkrzywda

THAT'S IT FOR TODAY!

My name is **Jakub Krzywda**.

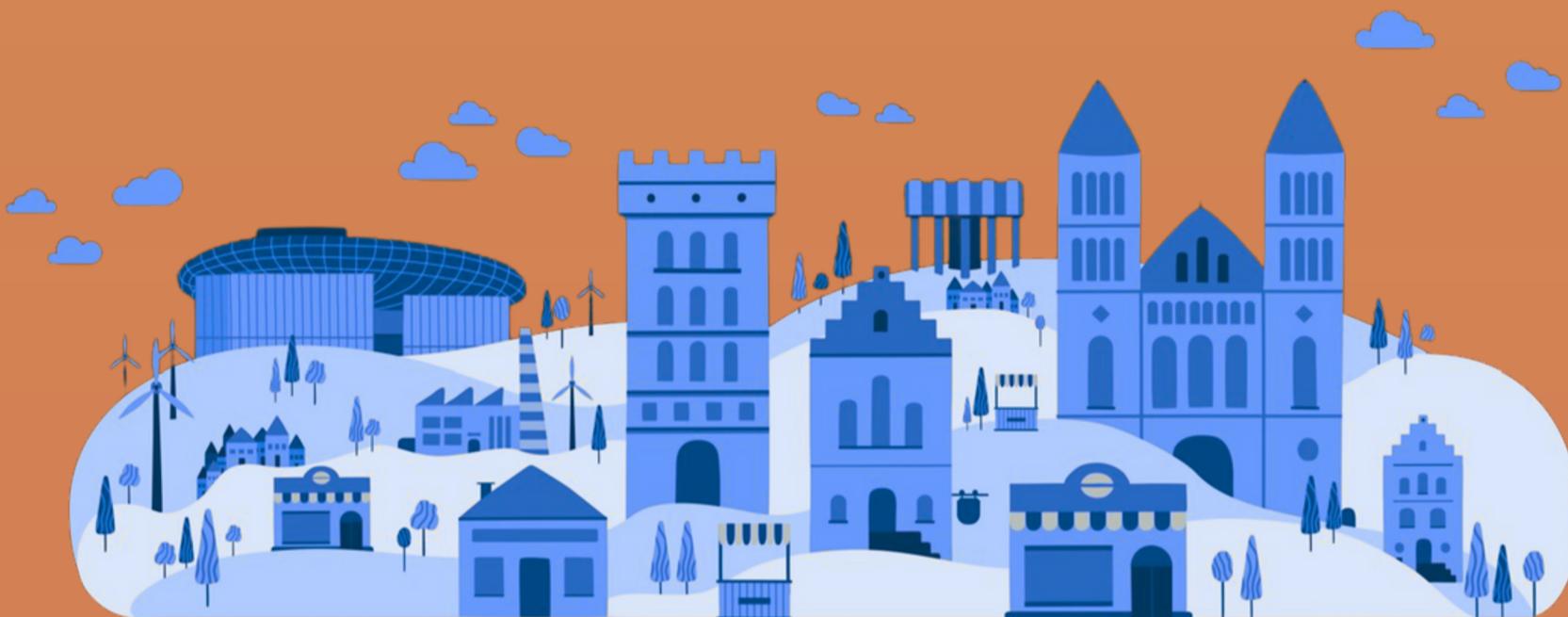
I'm a Cloud Native **Training Manager** and Senior Engineer.

I post about **Kubernetes**, **Cloud Native** technologies and **DevOps** practices.



JAKUB KRZYWDA
@jakubkrzywda

UPCOMING TRAINING



Kubernetes Administration (LFS458) Training

Lund, Sweden

October 14-17

Accelerate your
preparations for
the exam!

