

Data: 2024-06-11

Imię: Tomasz

Nazwisko: Wawer

Rok: 3

Grupa: GR1

Id studenta: 37285

Temat pracy: Jak różnią się podejścia do zarządzania aktualizacjami i rollbackami w Docker Sw

Domena: sprzęt

Źródła:

1. "Kubernetes Up and Running" - Kelsey Hightower, Brendan Burns, Joe Beda

2. "Docker Deep Dive" - Nigel Poulton

3. "Kubernetes: Up & Running: Dive into the Future of Infrastructure" - Kelsey Hightower, Brendan

4. Docker Documentation: <https://docs.docker.com/>

5. Kubernetes Documentation: <https://kubernetes.io/docs/>

6. "The Kubernetes Book" - Nigel Poulton

Treść:

Zarządzanie aktualizacjami i rollbackami jest krytycznym aspektem w środowiskach kontenerow

****1. Docker Swarm:****

Docker Swarm jest orkiestracyjnym narzędziem zaprojektowanym do prostego zarządzania klas

- **Aktualizacje:**

Docker Swarm oferuje "Rolling Update," co oznacza, że można stopniowo aktualizować konten

```
```yaml
version: '3.7'
services:
 my_service:
 image: my_image:latest
 deploy:
 update_config:
 parallelism: 2
 delay: 10s
...```
```

### **- \*\*Rollback:\*\***

Jeśli coś pójdzie nie tak podczas aktualizacji, Docker Swarm oferuje automatyczny rollback. Sw

```
```yaml
deploy:
  rollback_config:
    parallelism: 2
    delay: 10s
...```
```

****2. Kubernetes:****

Kubernetes jest zaawansowanym narzędziem orkiestracyjnym, które oferuje bogatszy zestaw fu

- **Aktualizacje:**

Kubernetes korzysta z mechanizmu Rolling Updates oraz Blue-Green Deployment lub Canary D

- ****Rolling Updates:**** Podobnie jak w Docker Swarm, Kubernetes pozwala na stopniowe aktual

```
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 name: my-deployment
spec:
 replicas: 3
 strategy:
 type: RollingUpdate
...```
```