



KubeCon



CloudNativeCon

Europe 2020

Virtual

*Stayin' Alive: PodDisruptionBudgets for **Maintenance** and **Upgrades***

Matthew Robson, Red Hat



KubeCon



CloudNativeCon

Europe 2020

Virtual



Matthew Robson

Principal Technical Account Manager

[@mattjrobson](#)

github.com/mrobson

What is a **PodDisruptionBudget**?



A **PodDisruptionBudget** is an **application owner** created object that defines the **minimum** number of **replicas** that must be **available** for that application to operate in a **stable** manner during a **voluntary disruption**

Why PodDisruptionBudget's?



Owned by the application team

Supportive of the operations team

Defines availability requirements

Respected by the eviction API

Deployment, ReplicationController, StatefulSet and
ReplicaSet objects

Involuntary disruptions

Explicit deletions and scaling

PodDisruptionBudgets for single replicas are burdensome

Drain can hang indefinitely

Do not overlap selectors

Meaningful **Name**

matchLabels corresponding to the controllers
.spec.selector

Appropriate **minAvailable**

or

Appropriate **maxUnavailable**

The Definition



KubeCon



CloudNativeCon

Europe 2020

Virtual

```
# cat django-ws-pdb.yaml
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: django-ws-pdb
spec:
  selector:
    matchLabels:
      app: django-ws
  minAvailable: 2 or 50%
  or
  maxUnavailable: 1 or 50%
```

```
# kubectl get deployment django-ws -o yaml
apiVersion: apps/v1
kind: Deployment
  labels:
    app: django-ws
    framework: django
spec:
  replicas: 3
  selector:
    matchLabels:
      app: django-ws
```



```
# kubectl describe poddisruptionbudget django-ws-pdb
```

```
Name:          django-ws-pdb
```

```
Namespace:     disruption
```

```
Min available: 2
```

```
Selector:      app=django-ws
```

```
Status:
```

```
  Allowed disruptions: 1
```

```
  Current:            3
```

```
  Desired:            2
```

```
  Total:              3
```


An Eviction



```
# kubectl drain worker-1.k8s.tamlab.redhat.com
node/worker-1.k8s.tamlab.redhat.com cordoned
pod/django-ws-68dbbdb8ff-v7dvf evicted
node/worker-1.k8s.tamlab.redhat.com evicted
```

```
# kubectl describe pdb django-ws-pdb
```

```
Name:          django-ws-pdb
```

```
Namespace:     disruption
```

```
Min available: 2
```

```
Selector:      app=django-ws
```

```
Status:
```

```
  Allowed disruptions: 0
```

```
  Current:             2
```

```
  Desired:             2
```

```
  Total:               2
```

Eviction Denied



```
# kubectl drain worker-2.k8s.tamlab.redhat.com --timeout=10s  
node/worker-2.k8s.tamlab.redhat.com cordoned
```

```
error when evicting pod "django-ws-68dbbdb8ff-7brhf" (will retry after  
5s): Cannot evict pod as it would violate the pod's disruption budget
```

```
error: unable to drain node "worker-2.k8s.tamlab.redhat.com", aborting
```

There are pending nodes to be drained:

```
worker-2.k8s.tamlab.redhat.com
```

```
error: Drain did not complete within 10s
```

Encourage application owners to define their **operating requirements** with **PodDisruptionBudgets**

Leverage **voluntary eviction** for cluster **maintenance** and **upgrades**

Remember the **caveats** and **watch out** for **bad practices**



KubeCon



CloudNativeCon

Europe 2020



Virtual



KEEP CLOUD NATIVE

CONNECTED

