

corewire

Kubernetes

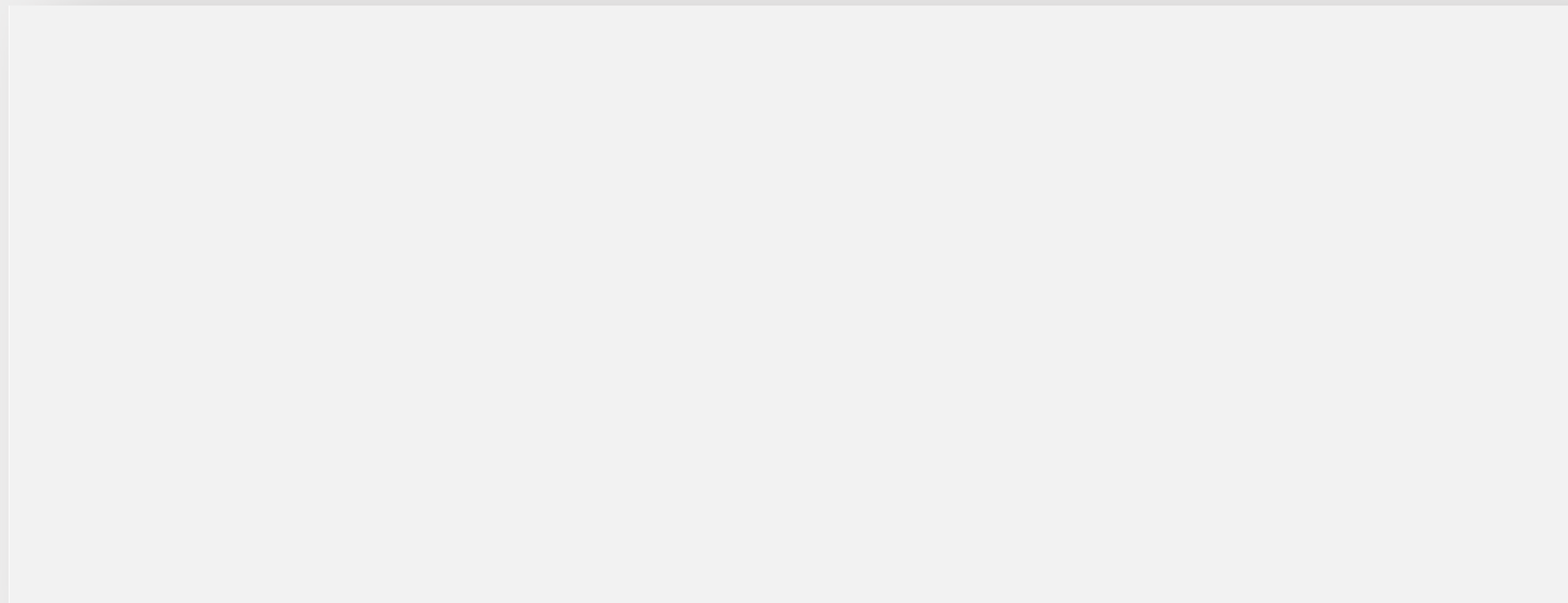
Erste Schritte

Folien-Hinweis

- `Space`, `Page down`: Nächste Folie
- `Page up`: Vorherige Folie
- `ESC`, `o`: Übersicht

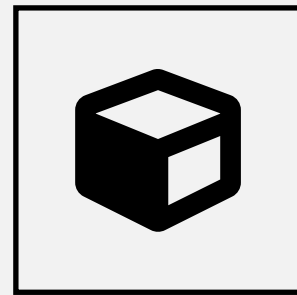
[Zur Kapitelübersicht](#)

Pod



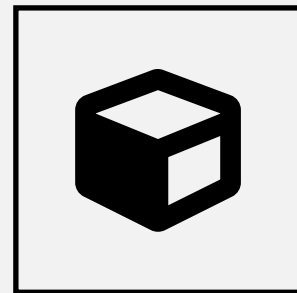
Pod

Pod

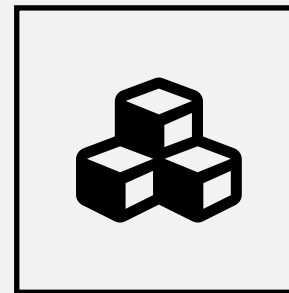


Pod

Pod

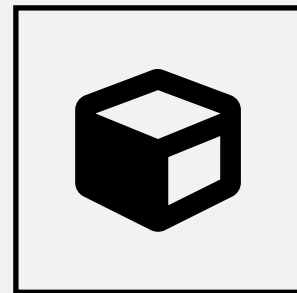


Pod

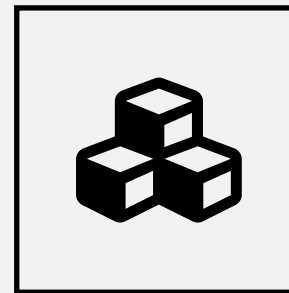


Pod

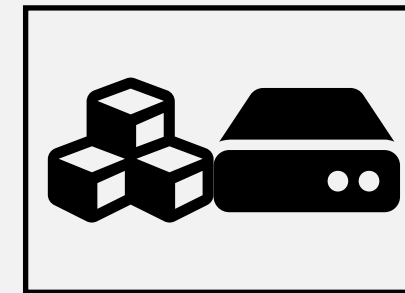
Pod



Pod

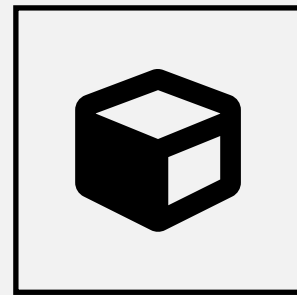


Pod



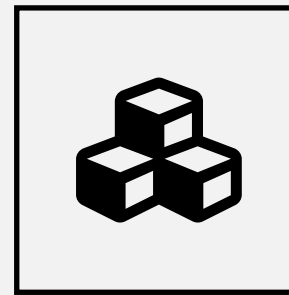
Pod

Pod



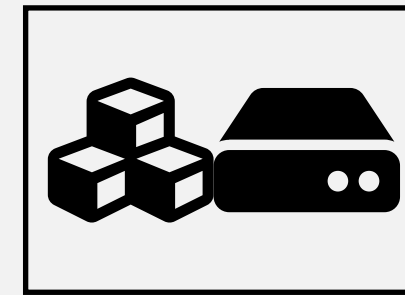
z.B. 10.10.10.1

Pod



z.B. 10.10.10.2

Pod



z.B. 10.10.10.3

Pod

- kleinste Einheit in einem Cluster
- besteht aus einem oder mehreren Containern
- Container teilen sich den Netzwerk Namespace

Podbeschreibung in YAML

```
# my-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-pod-name
spec:
  containers:
  - name: my-container-name
    image: nginx
```

YAML

- `Key: Value` Mapping
- Endet auf `.yaml` oder `.yml`
- Hierarchie durch gleichmäßige Einrückung (Leerzeichen)

YAML

```
# Kommentar  
Key1: Value  
Key2:  
  Subkey: "String"  
  Subkey2: "#kein Kommentar"  
List:  
  - Key1  
  - "String"
```

YAML vs. Json

```
# Kommentar  
Key1: Value  
Key2:  
  Subkey: "String"  
  Subkey2: "#kein Kommentar"  
List:  
  - Key1  
  - "String"
```

```
{  
  "Key1": Value,  
  "Key2": {  
    "Subkey": "String",  
    "Subkey2": "#kein Kommentar"  
  },  
  "List": [  
    "Key1",  
    "String"  
  ]  
}
```

Podbeschreibung in YAML

```
# my-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-pod-name
spec:
  containers:
  - name: my-container-name
    image: nginx
```

Pods starten

Pod starten

```
kubectl apply -f my-pod.yaml
```

Alternativ:

```
kubectl run --image=nginx my-pod-name
```

Pods starten

Pod starten

```
kubectl apply -f my-pod.yaml
```

Alternativ:

```
kubectl run --image=nginx my-pod-name  
kubectl run --image=nginx my-pod-name --dry-run=client -o=yaml > pod.yaml
```

Weitere Pod-Befehle

Pods anzeigen

```
kubectl get pods
```

Podbeschreibung anzeigen

```
kubectl describe pod my-pod-name
```

Pod löschen

```
kubectl delete pod my-pod-name
```

Demo: Pods starten

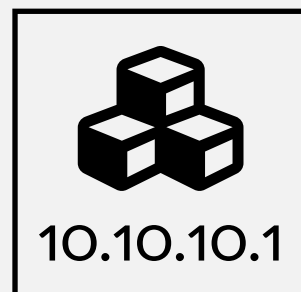
Labels

```
# my-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-pod-name
  labels:
    app: my-app
    environment: production
    tier: frontend
spec:
  containers:
  - name: my-container-name
    image: nginx
```

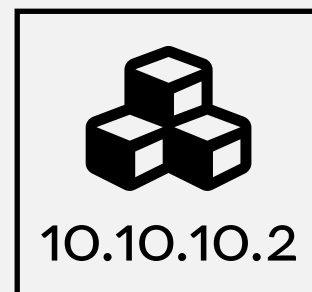
Labels

- Beliebige Key-Value-Paare
- Helfen bei der Organisation (von Pods)
- Werden als Selektoren genutzt

Services



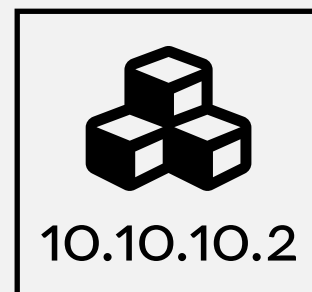
Services



Services

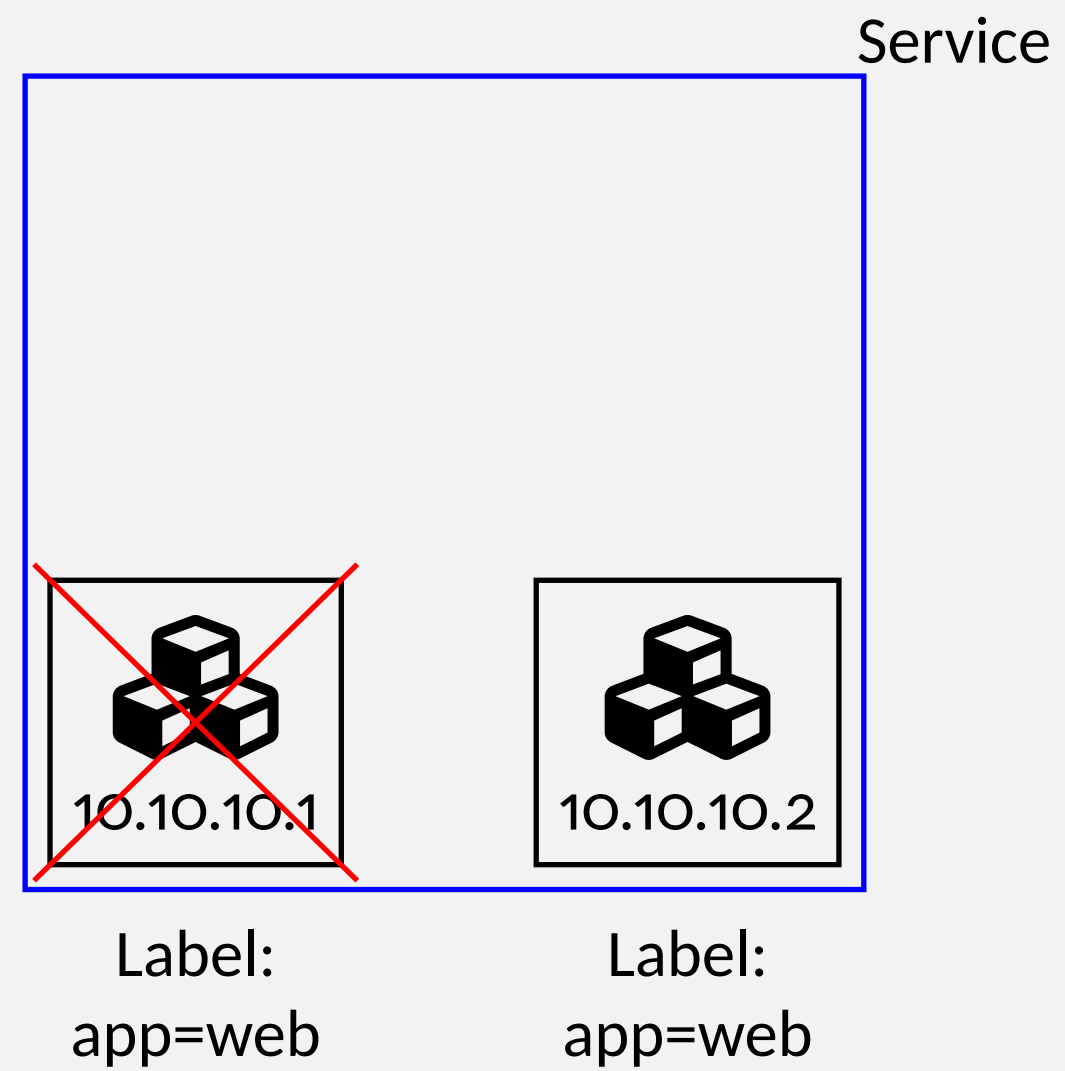


Label:
app=web

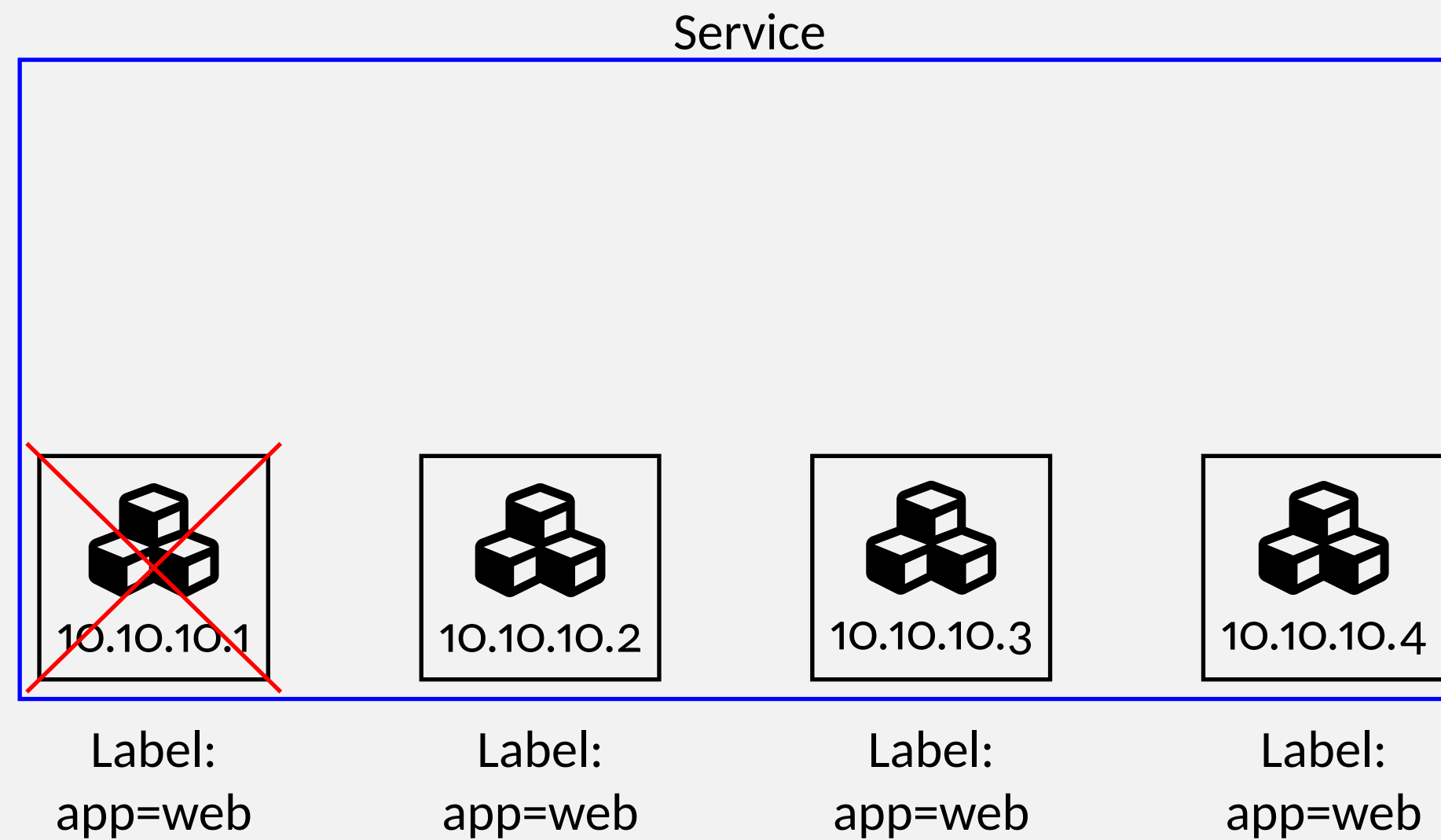


Label:
app=web

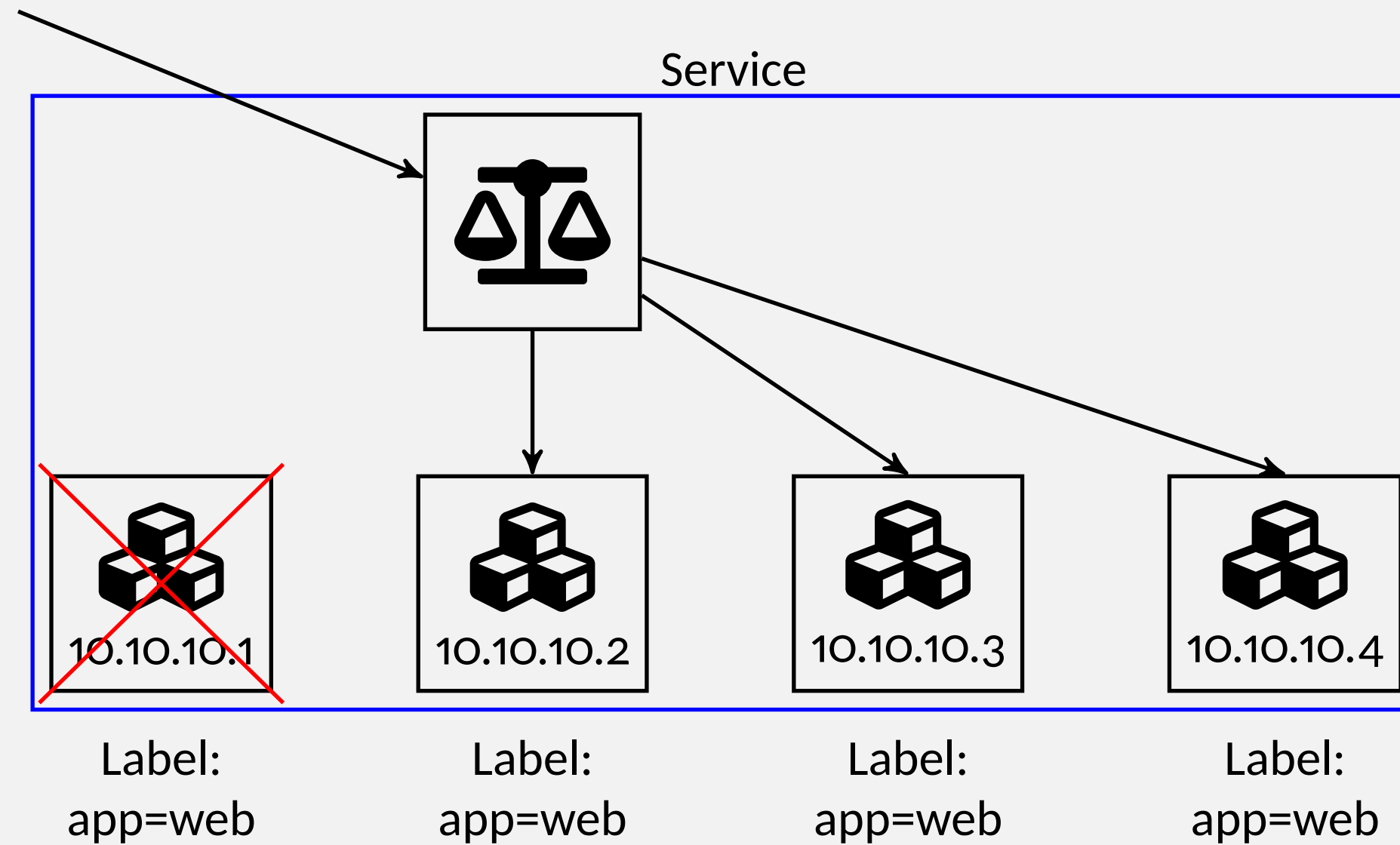
Services



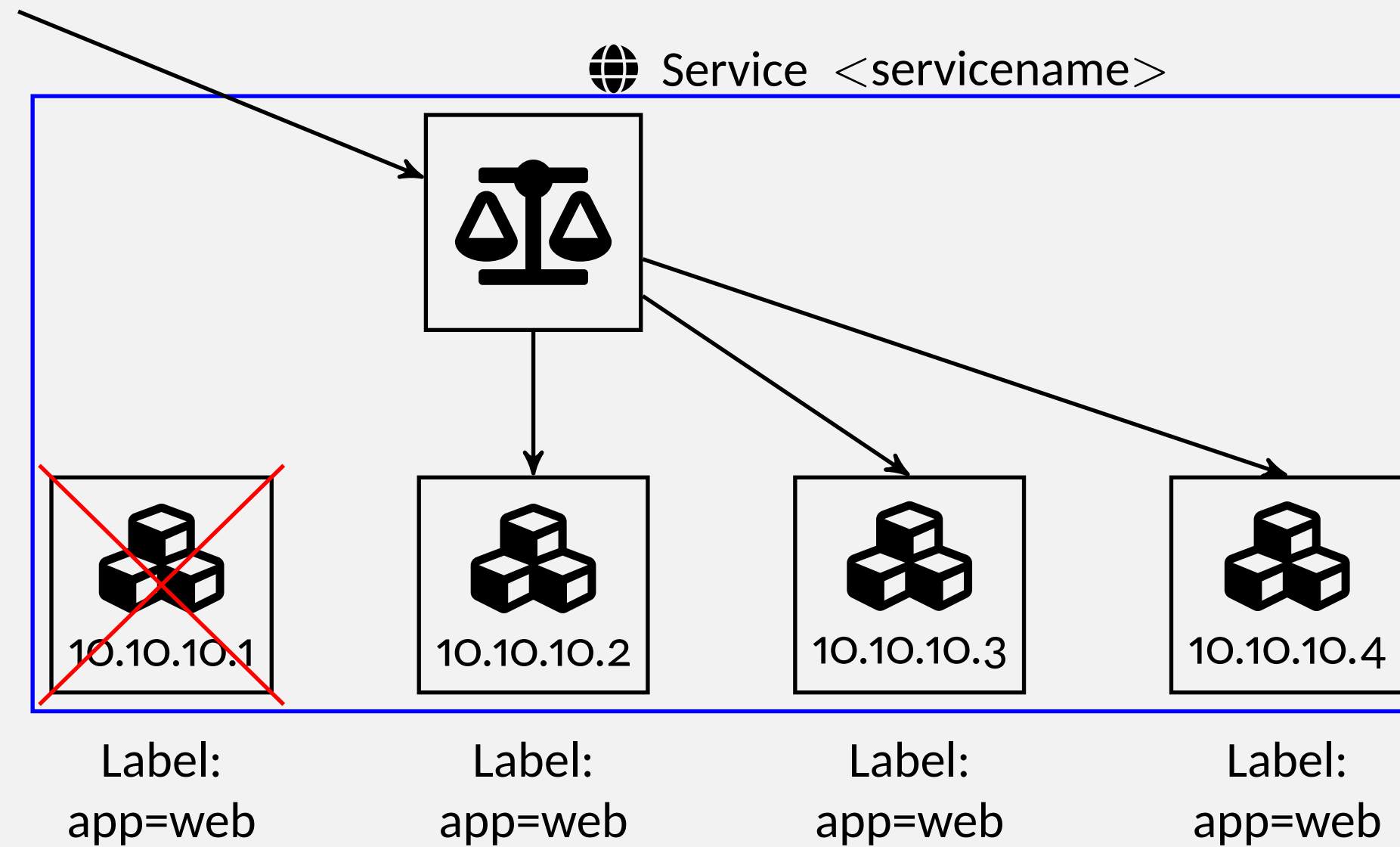
Services



Services



Services



Services

- Pods werden bei Update ersetzt und bekommen neue IP
- Services
 - nutzen Label als Pod-Selektoren
 - bieten Loadbalancing und DNS-Endpunkt

Servicebeschreibung in YAML

```
# my-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Lab

- Gehen Sie auf: <https://learn.corewire.de>
- Navigieren Sie nach:
 - Helm
 - Aufgaben
 - Erste Schritte

Annotations

- Beliebige Key-Value-Paare
- Für Metadaten:
 - Build-Informationen, Release Notes
 - Timestamps, Versionen
 - Konfiguration für externe Tools

Labels vs. Annotations

- Performance des Api-Servers
 - Labels werden indiziert, Annotations nicht
- Logische Trennung:
 - Labels: zum Organisieren, Filtern und Auswählen von Ressourcen
 - Annotations: nicht-essentielle, zusätzliche Metadaten

Zur Kapitelübersicht

- Vorheriges Kapitel: [Kubernetes Einführung](#)
- Nächstes Kapitel: [Pod Management](#)