

corewire

Docker & Kubernetes Troubleshooting

Kubernetes Developer Troubleshooting

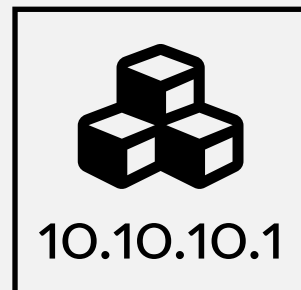
Folien-Hinweis

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

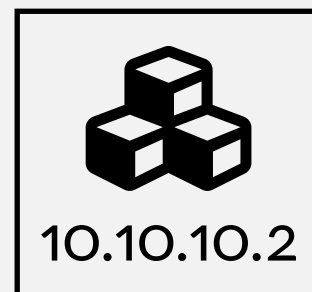
[Zur Kapitelübersicht](#)

Kubernetes Auffrischung

Services



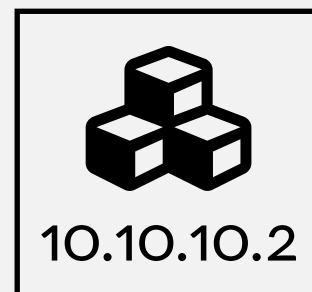
Services



Services

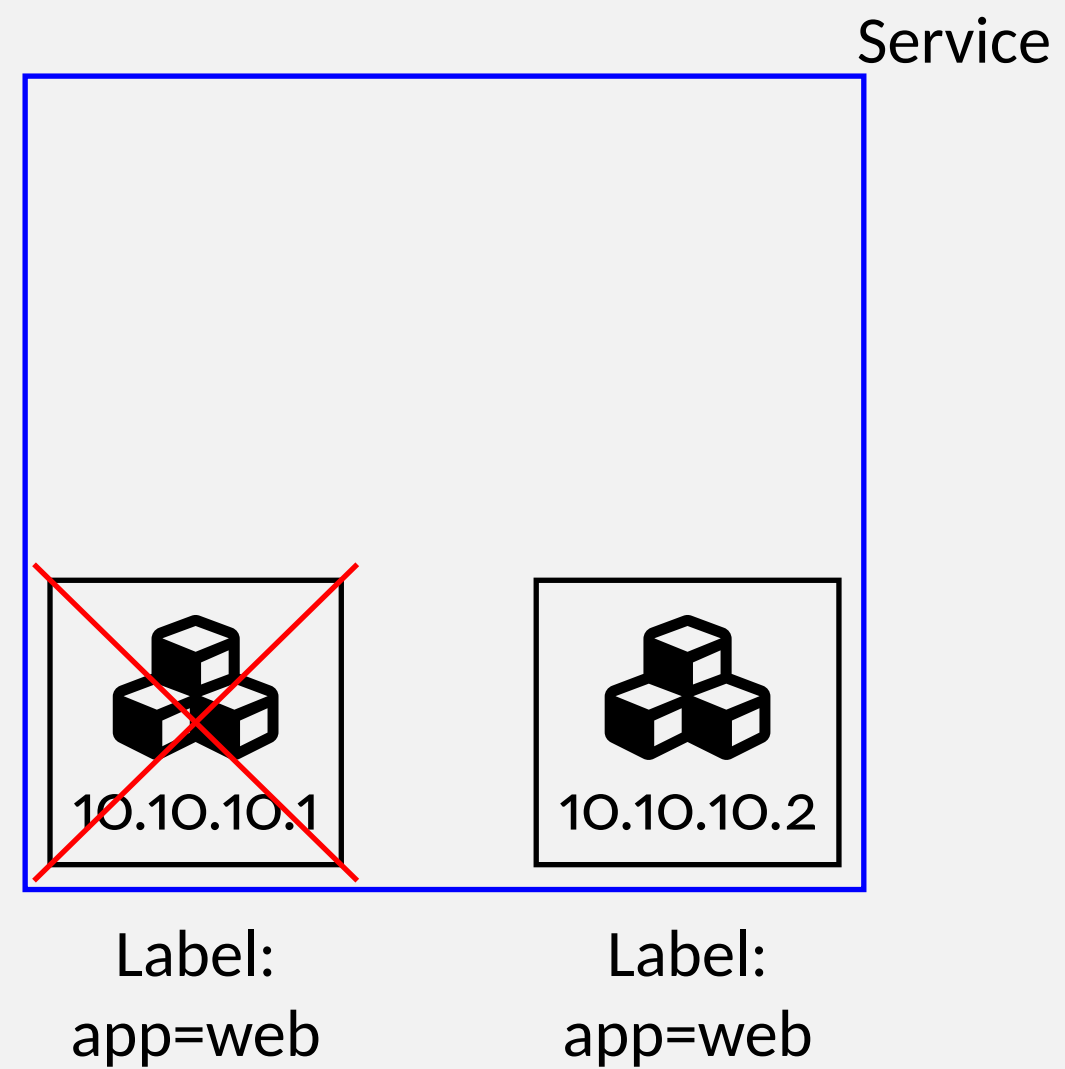


Label:
app=web

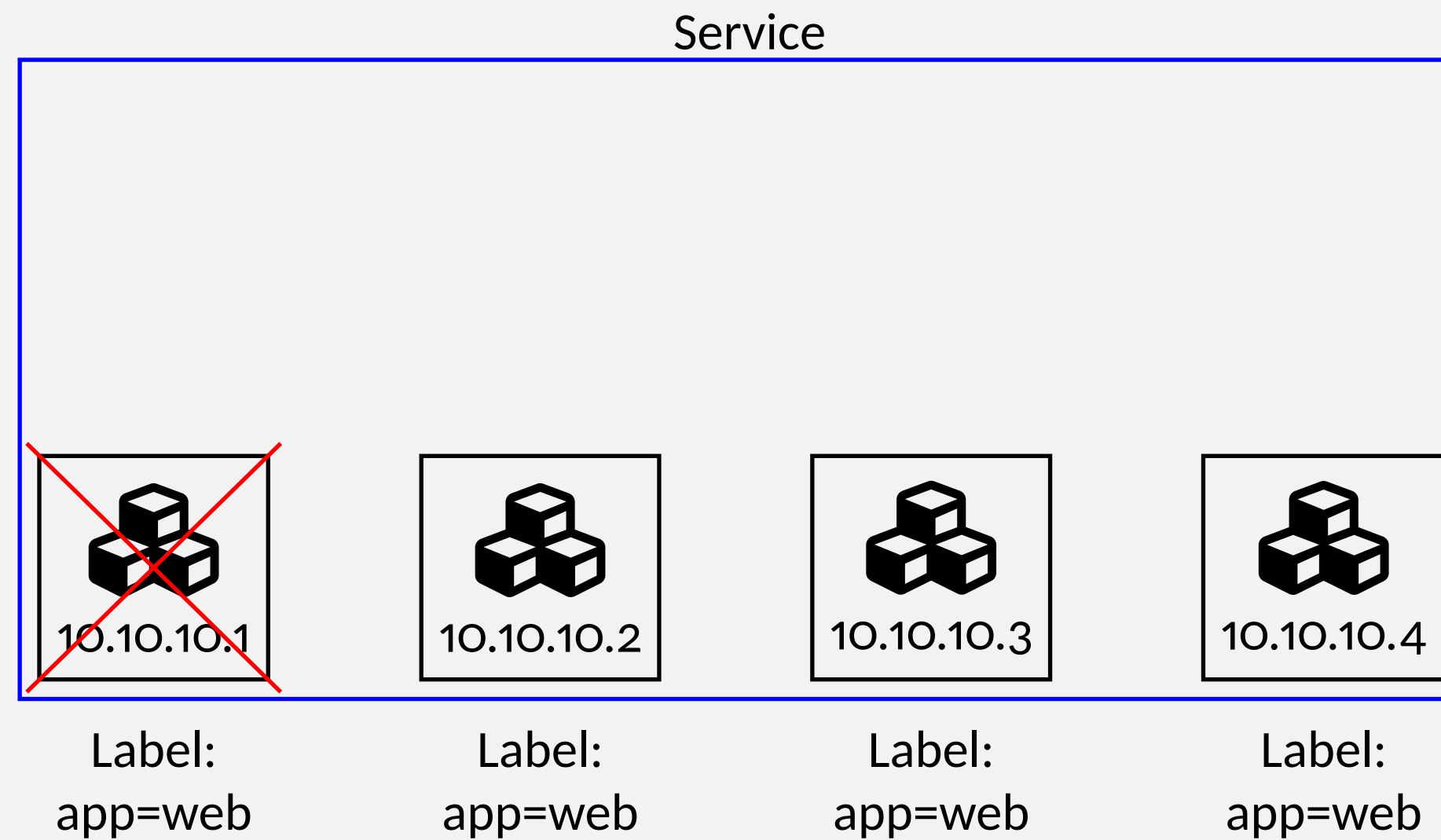


Label:
app=web

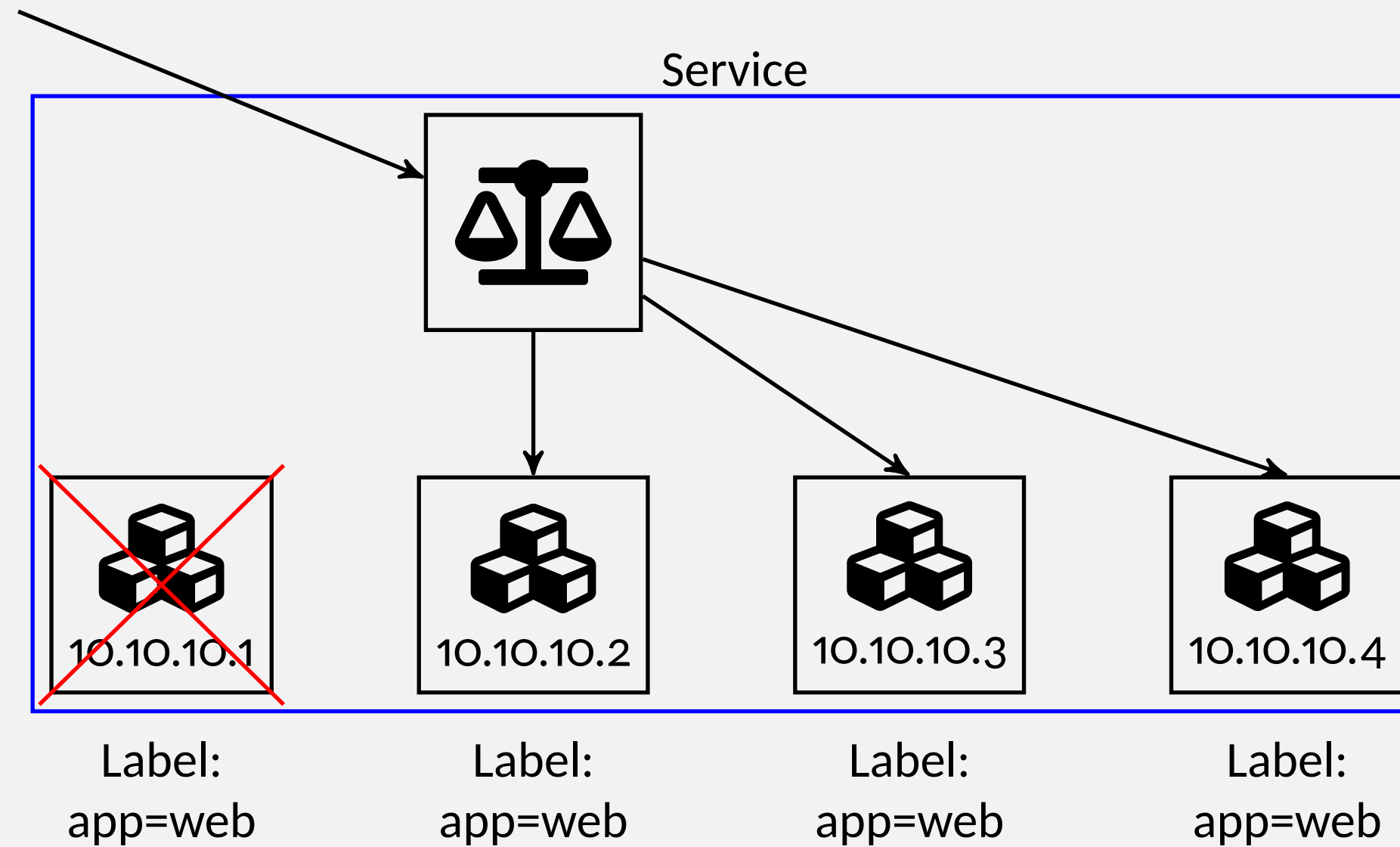
Services



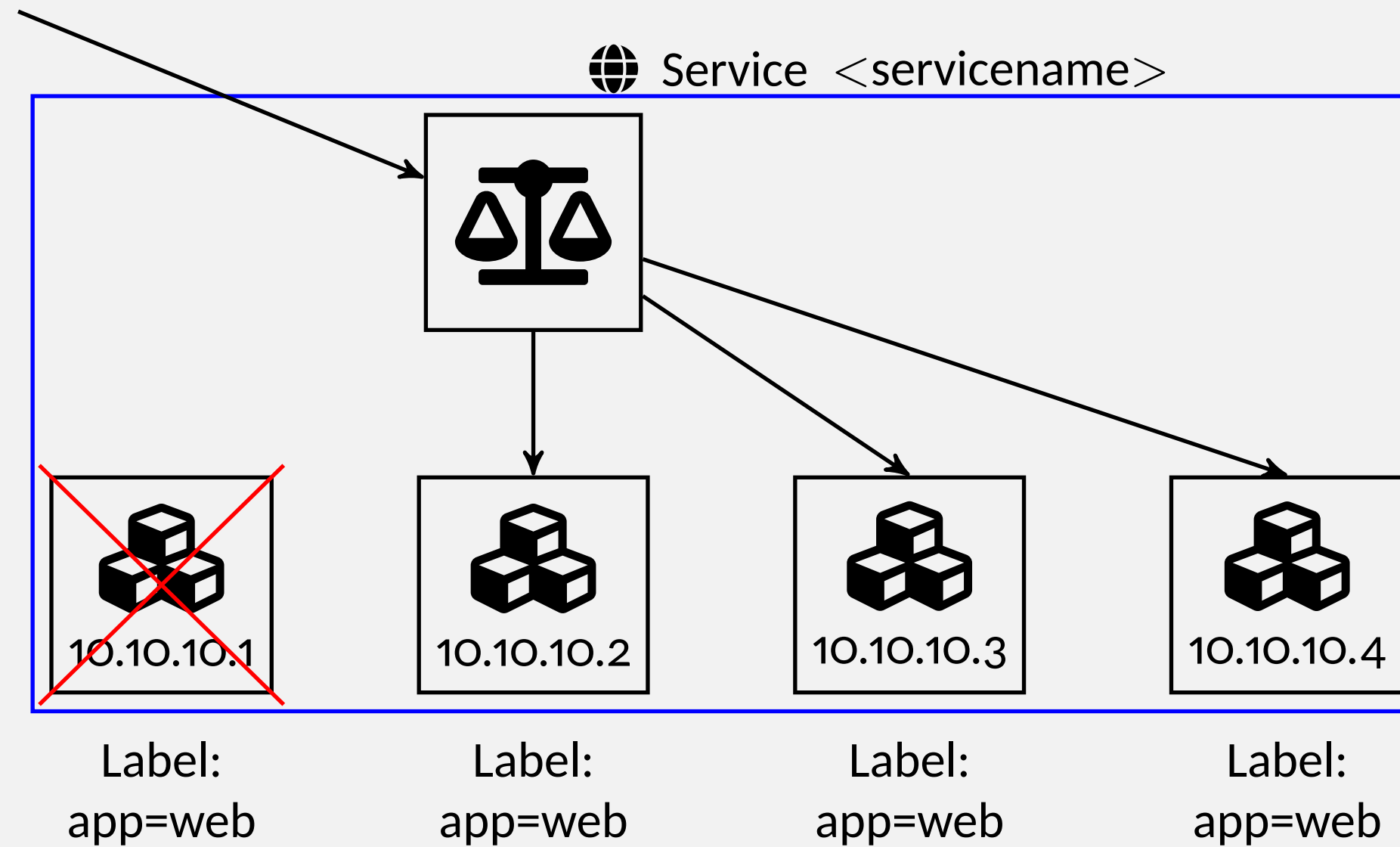
Services



Services



Services

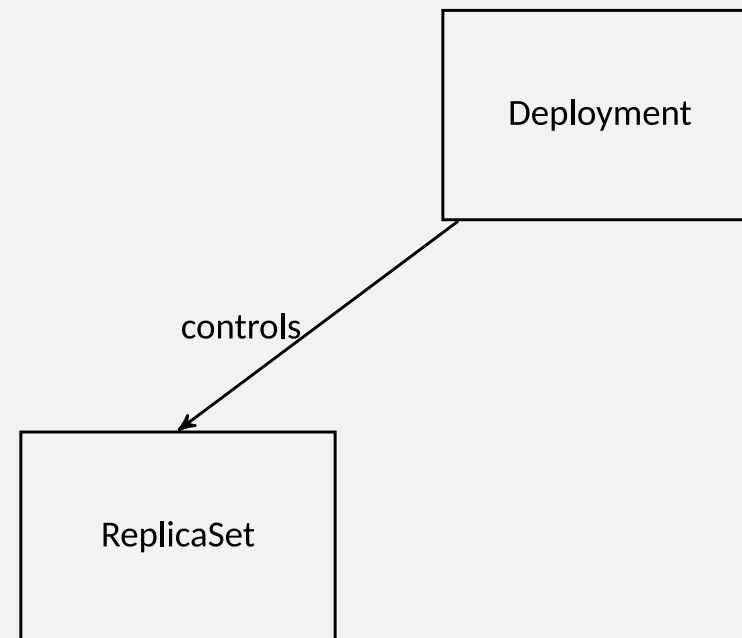


Deployment, ReplicaSet, Service

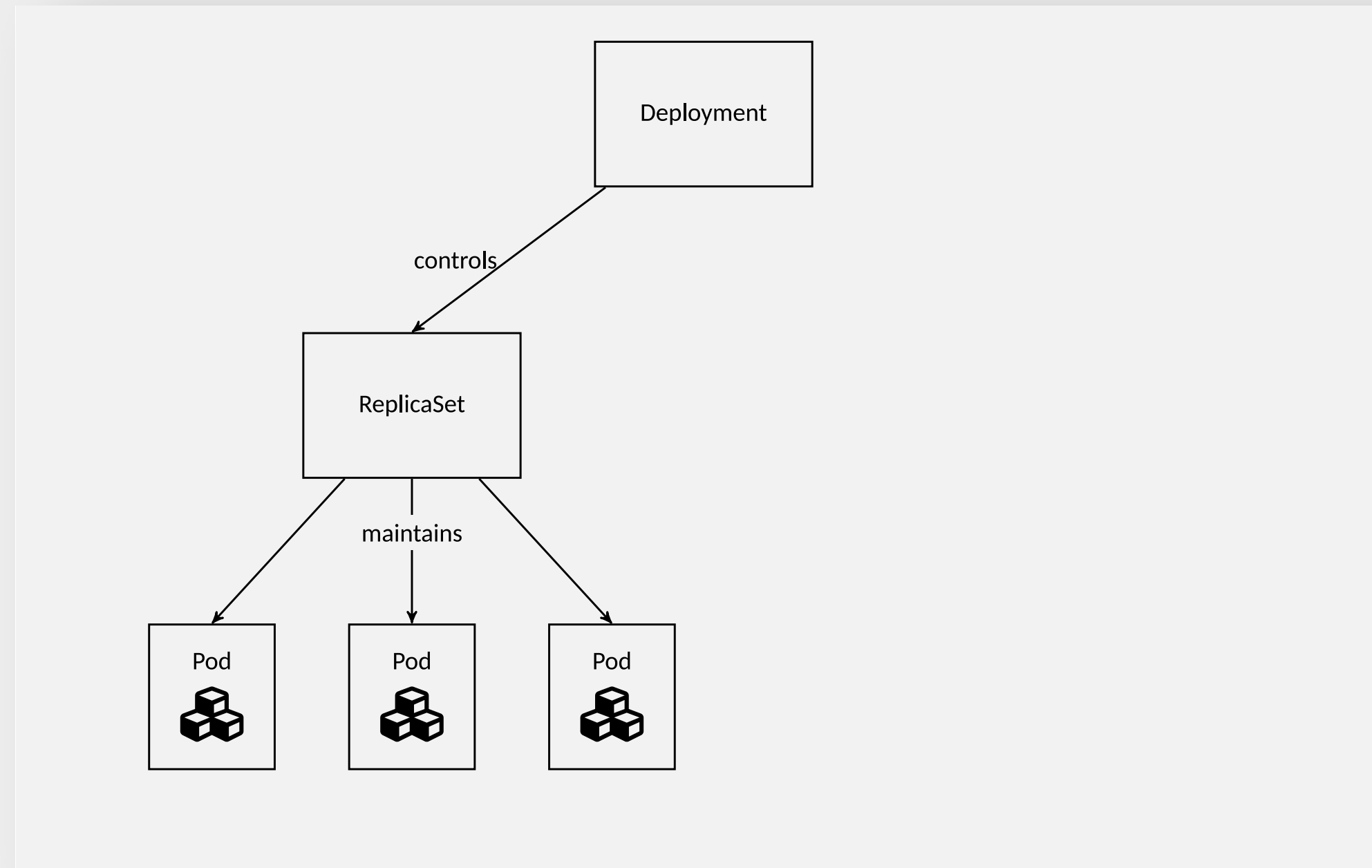


Deployment

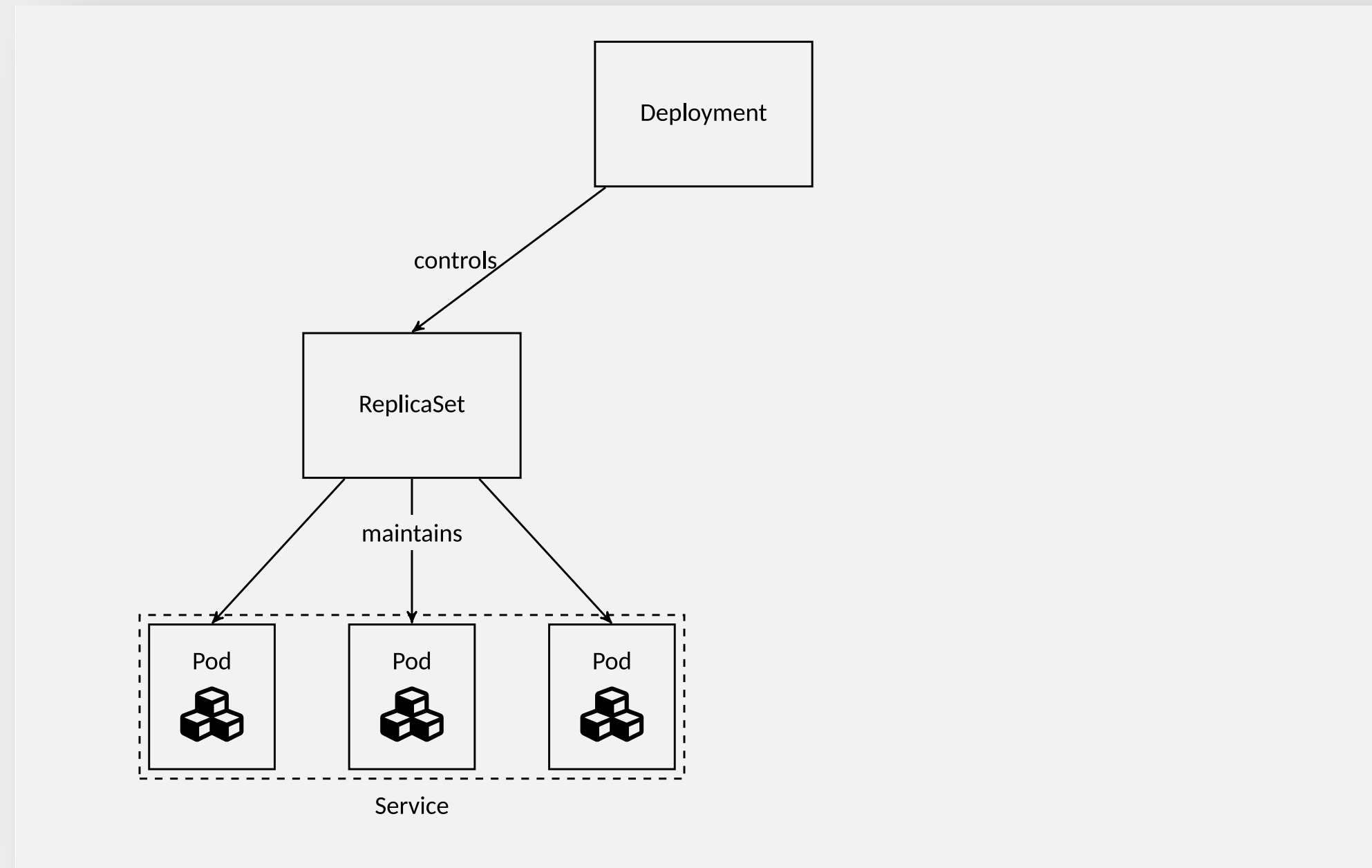
Deployment, ReplicaSet, Service



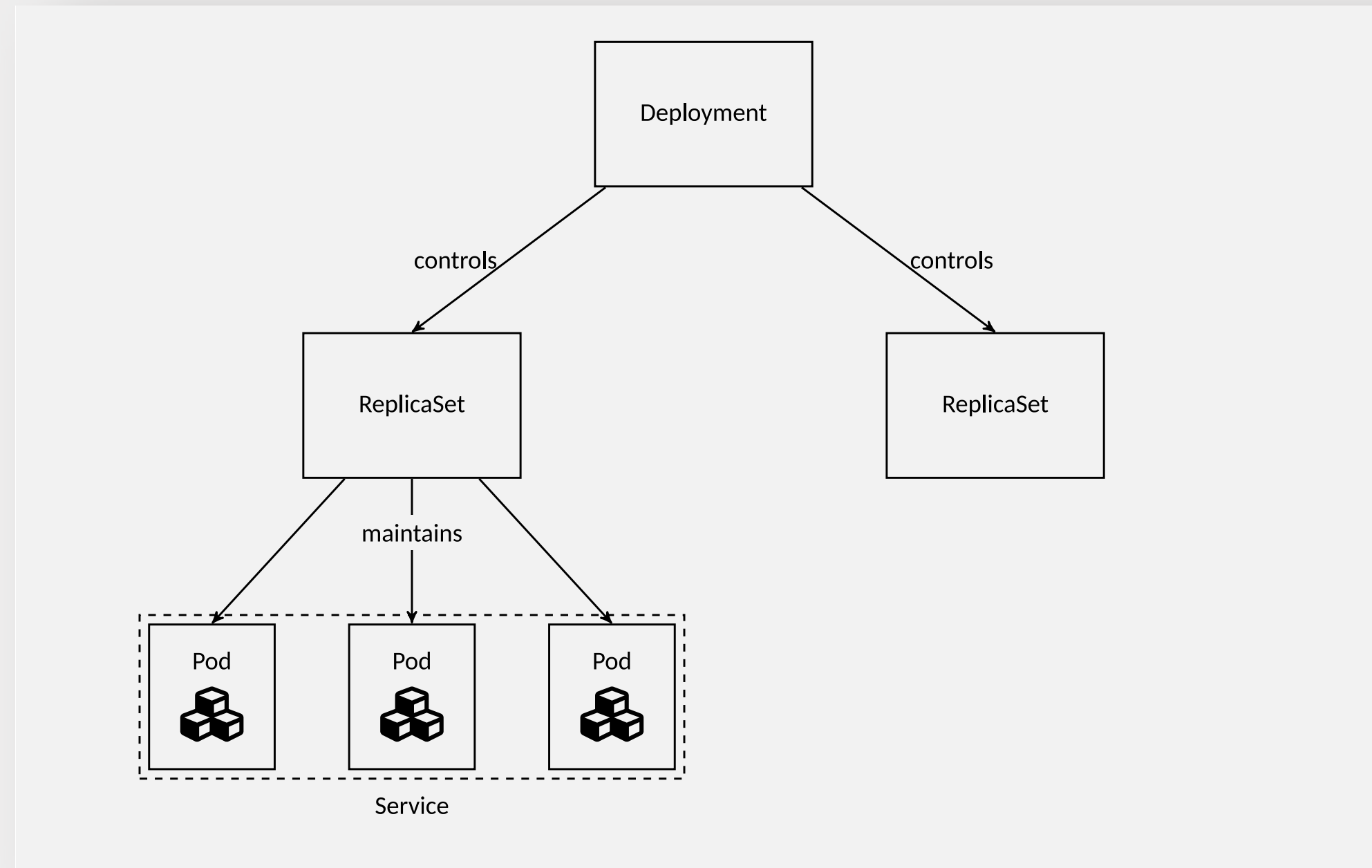
Deployment, ReplicaSet, Service



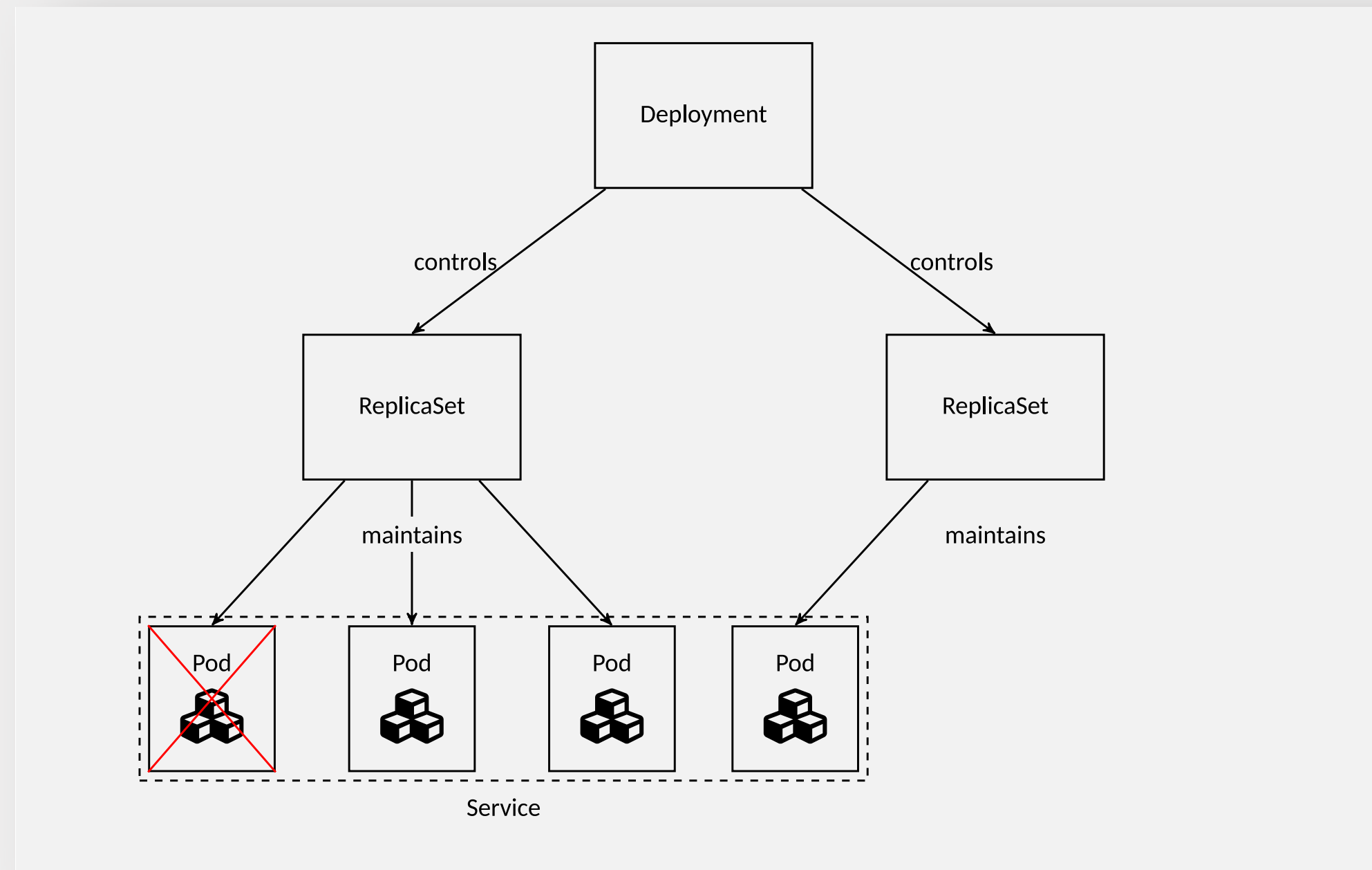
Deployment, ReplicaSet, Service



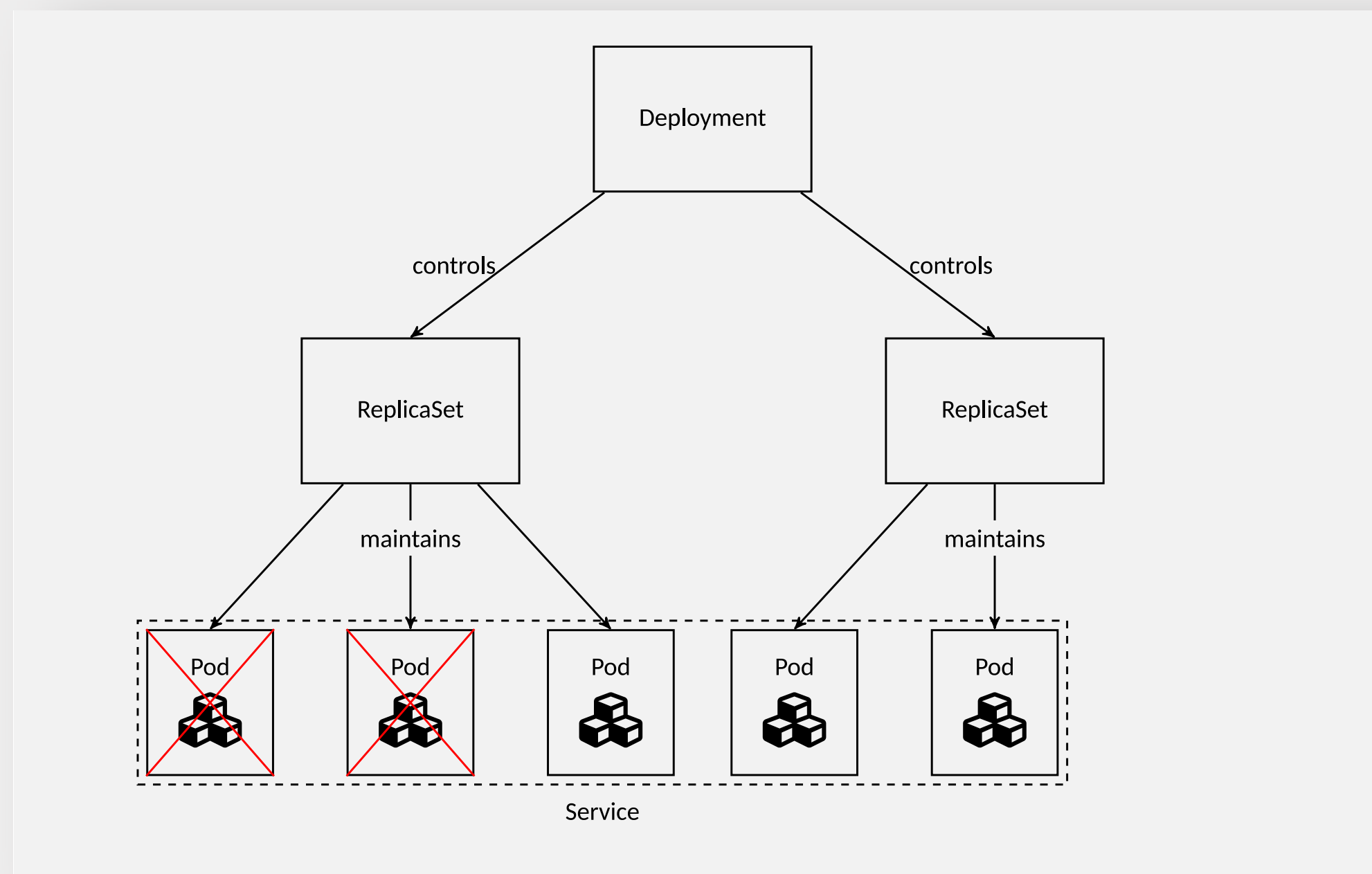
Deployment, ReplicaSet, Service



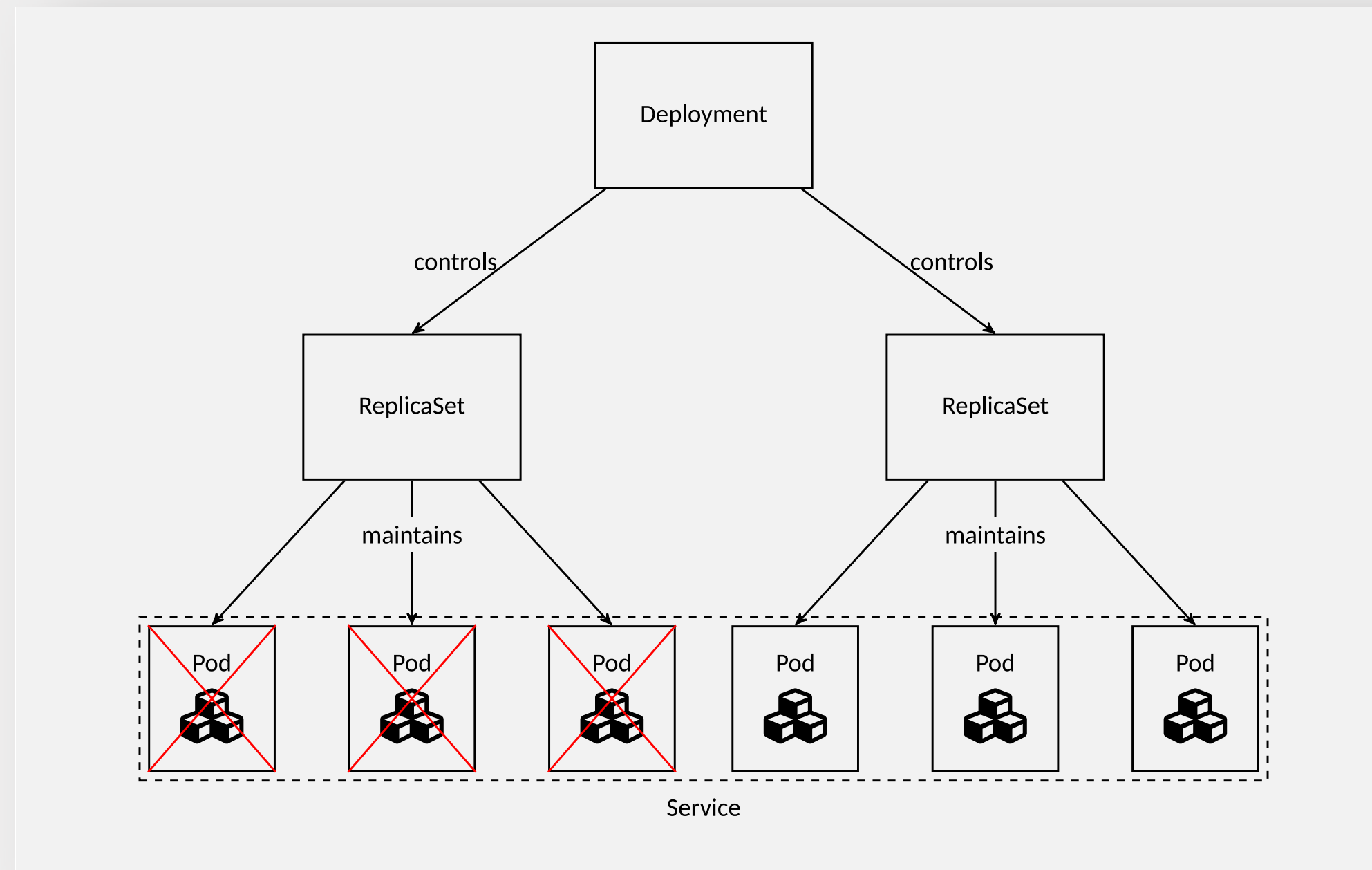
Deployment, ReplicaSet, Service



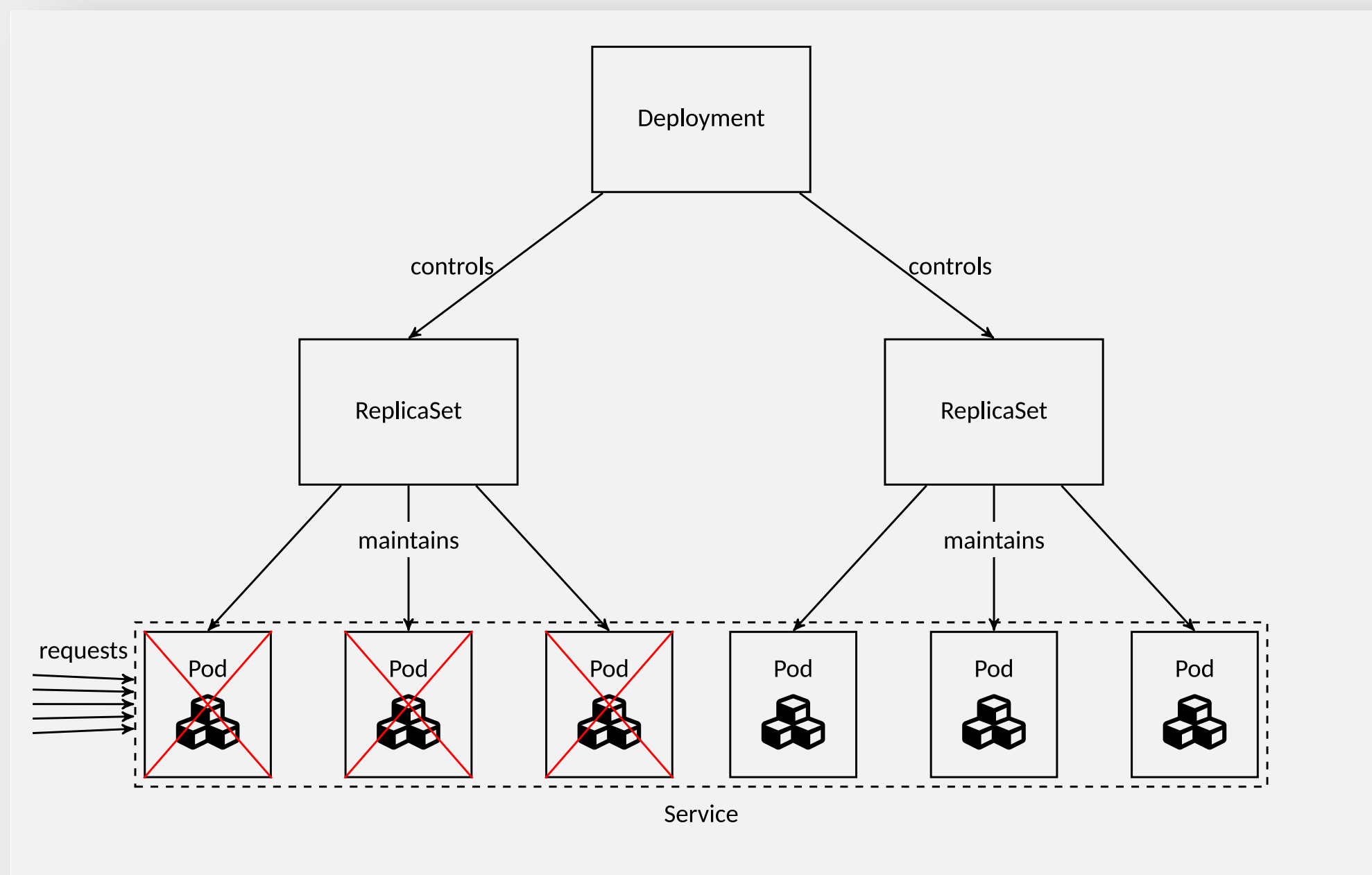
Deployment, ReplicaSet, Service



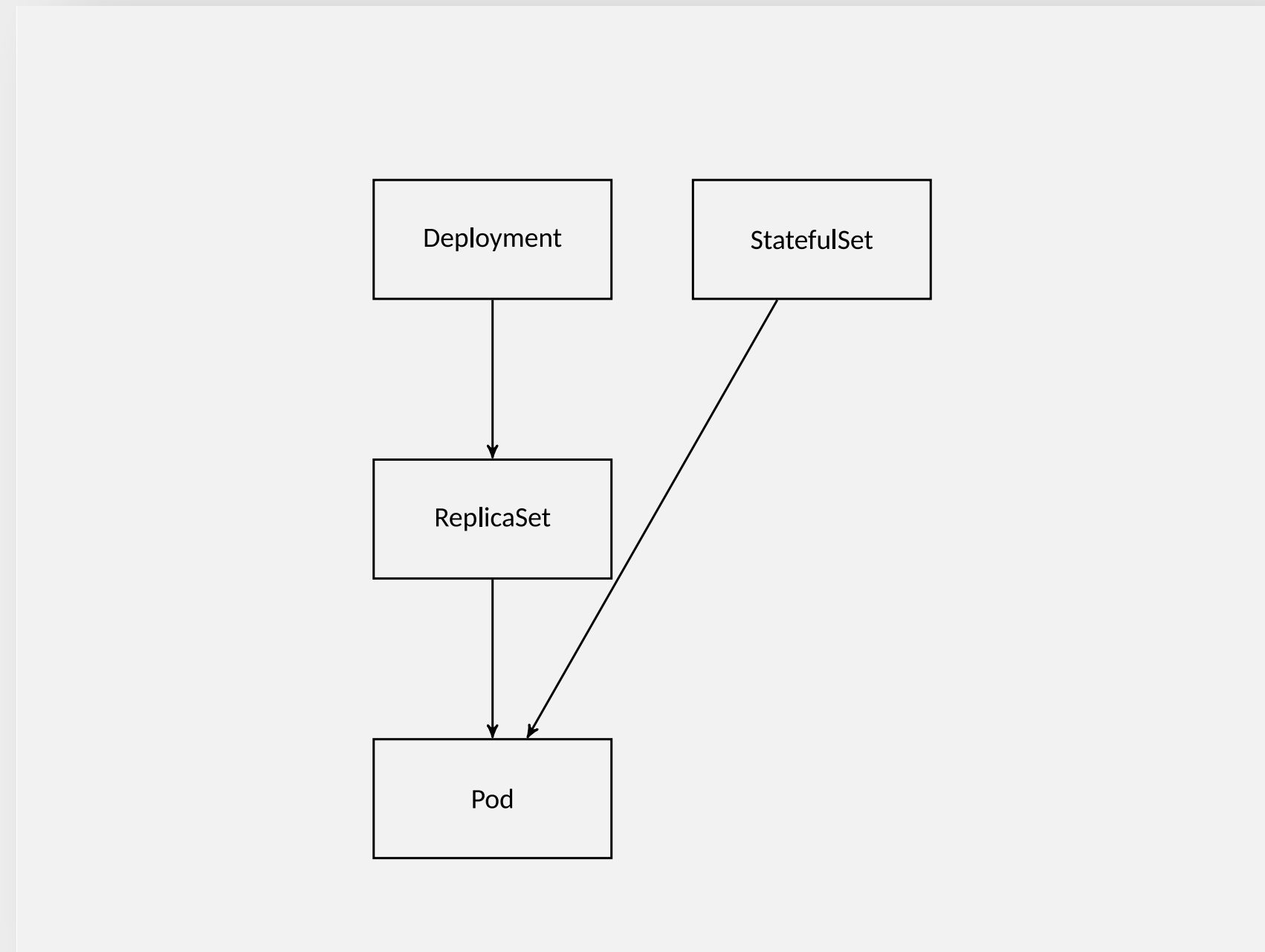
Deployment, ReplicaSet, Service



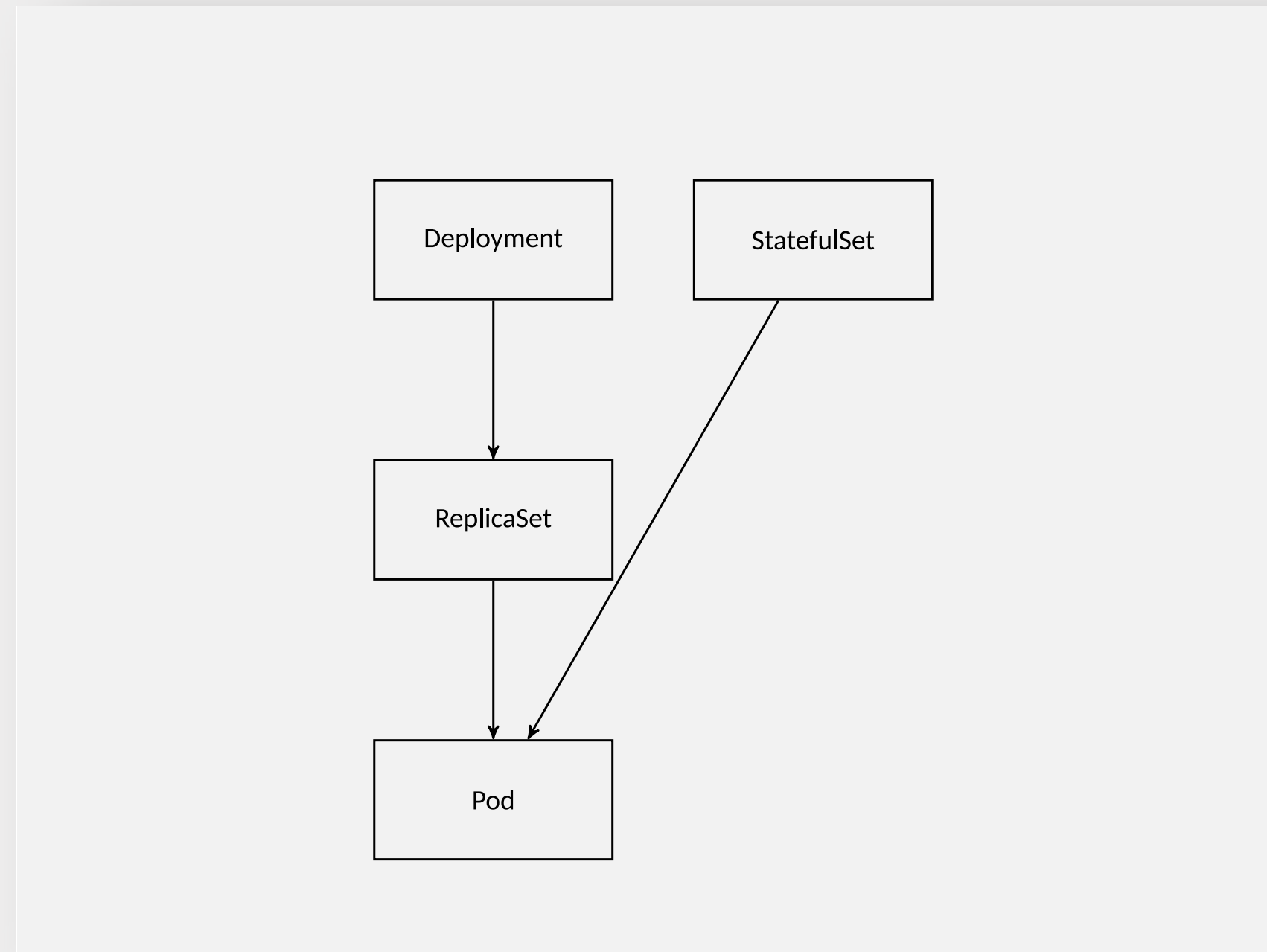
Deployment, ReplicaSet, Service



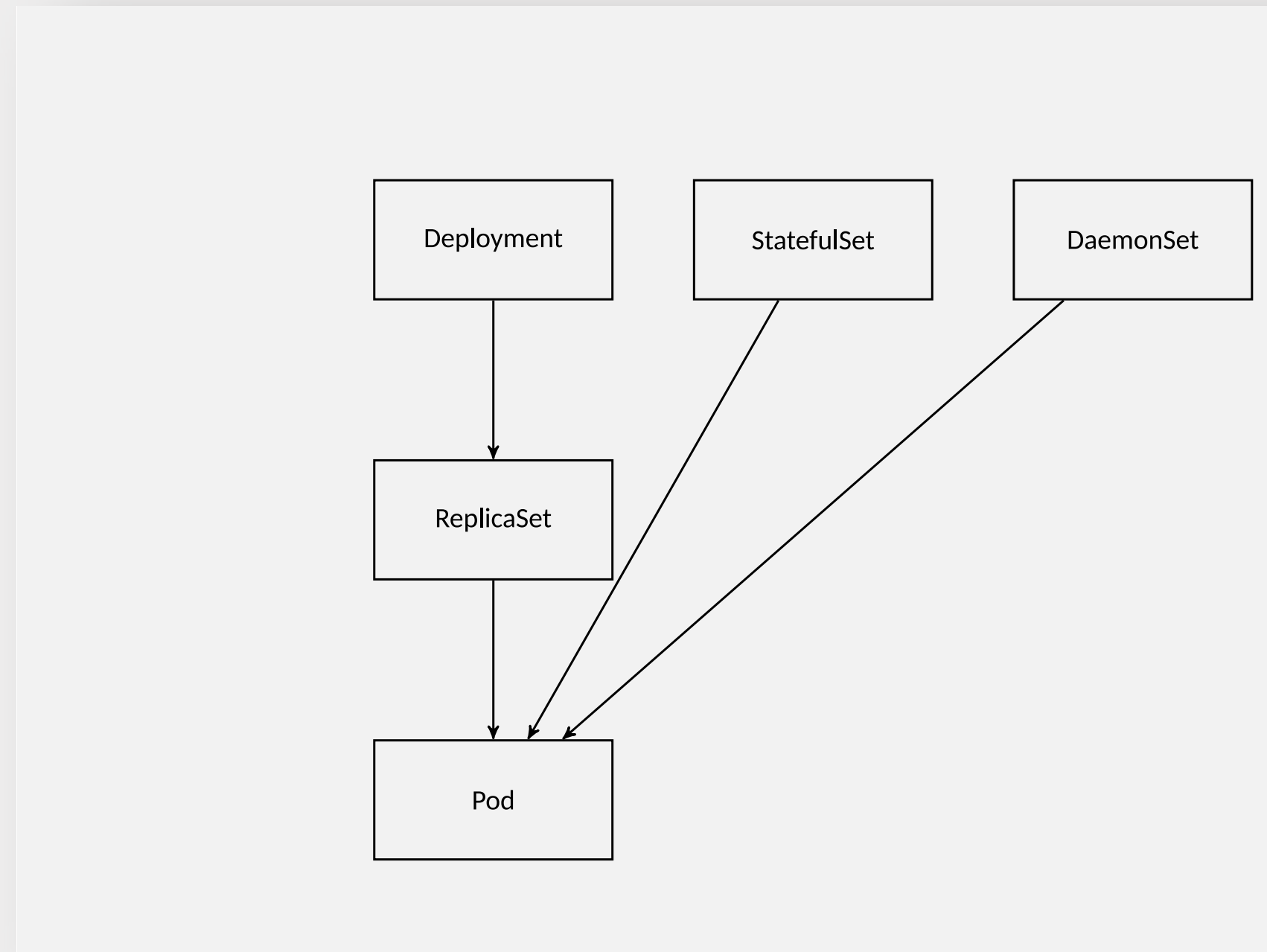
Pod Creation durch StatefulSet



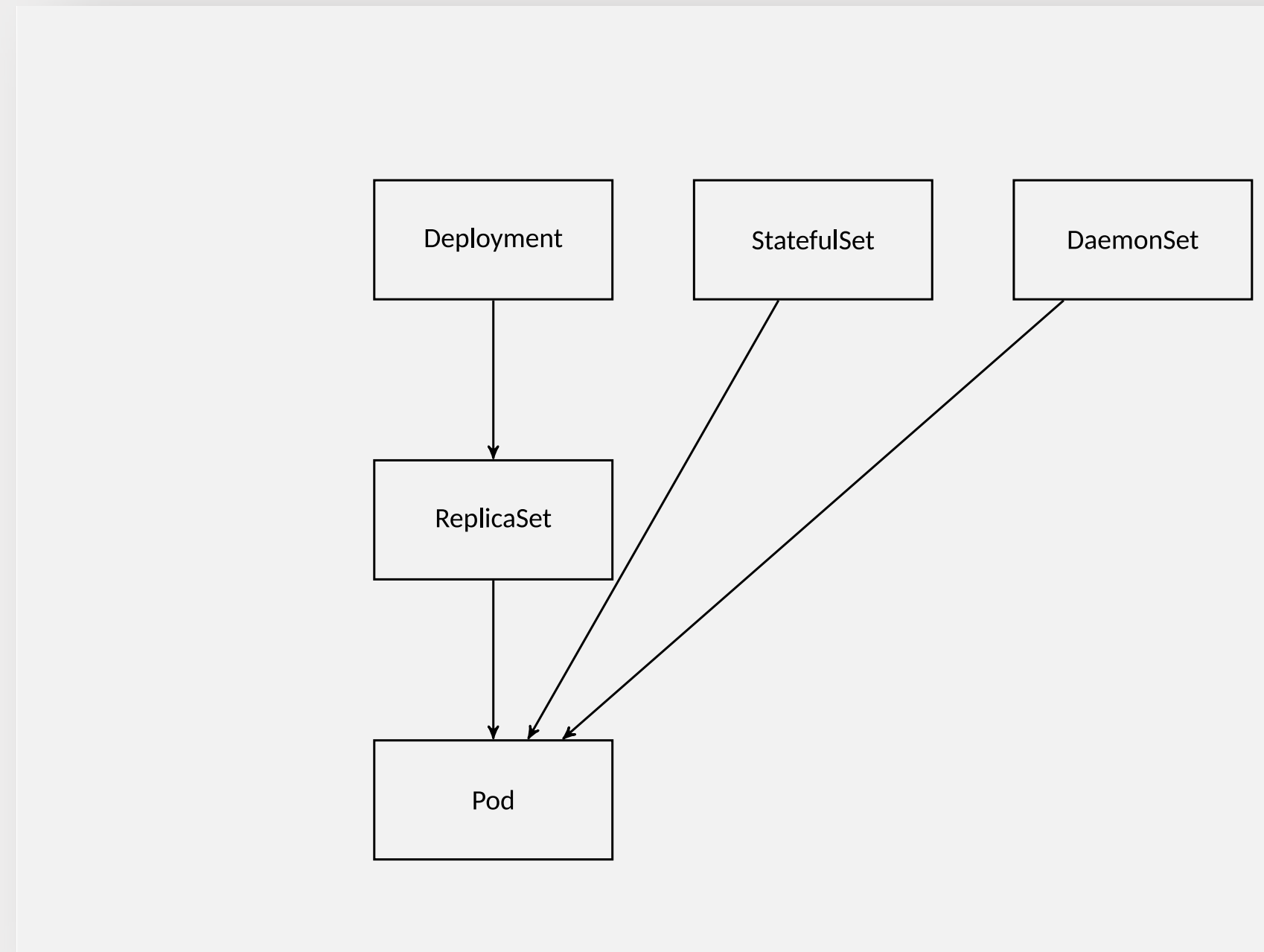
Pod Creation durch DaemonSets



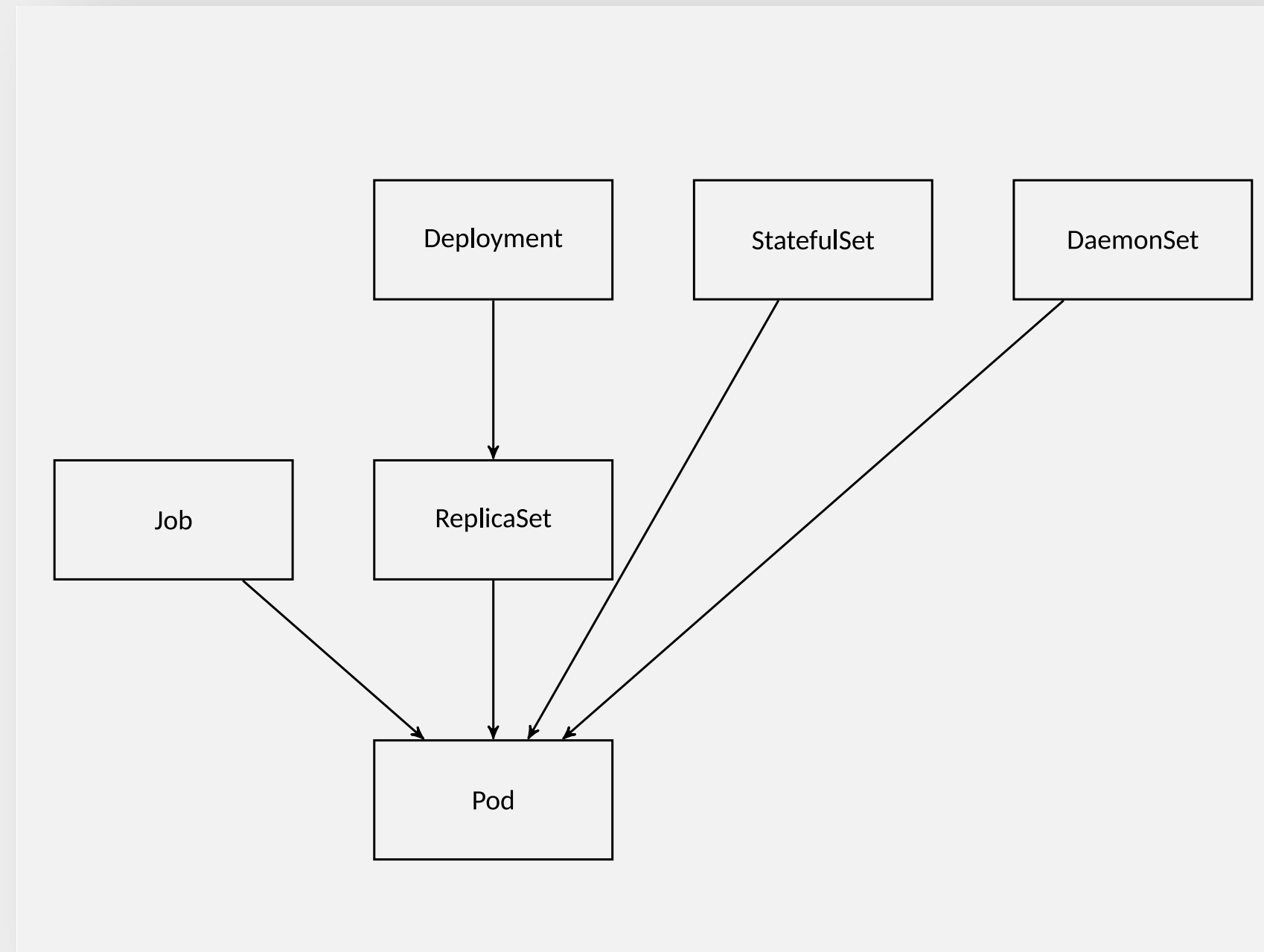
Pod Creation durch DaemonSets



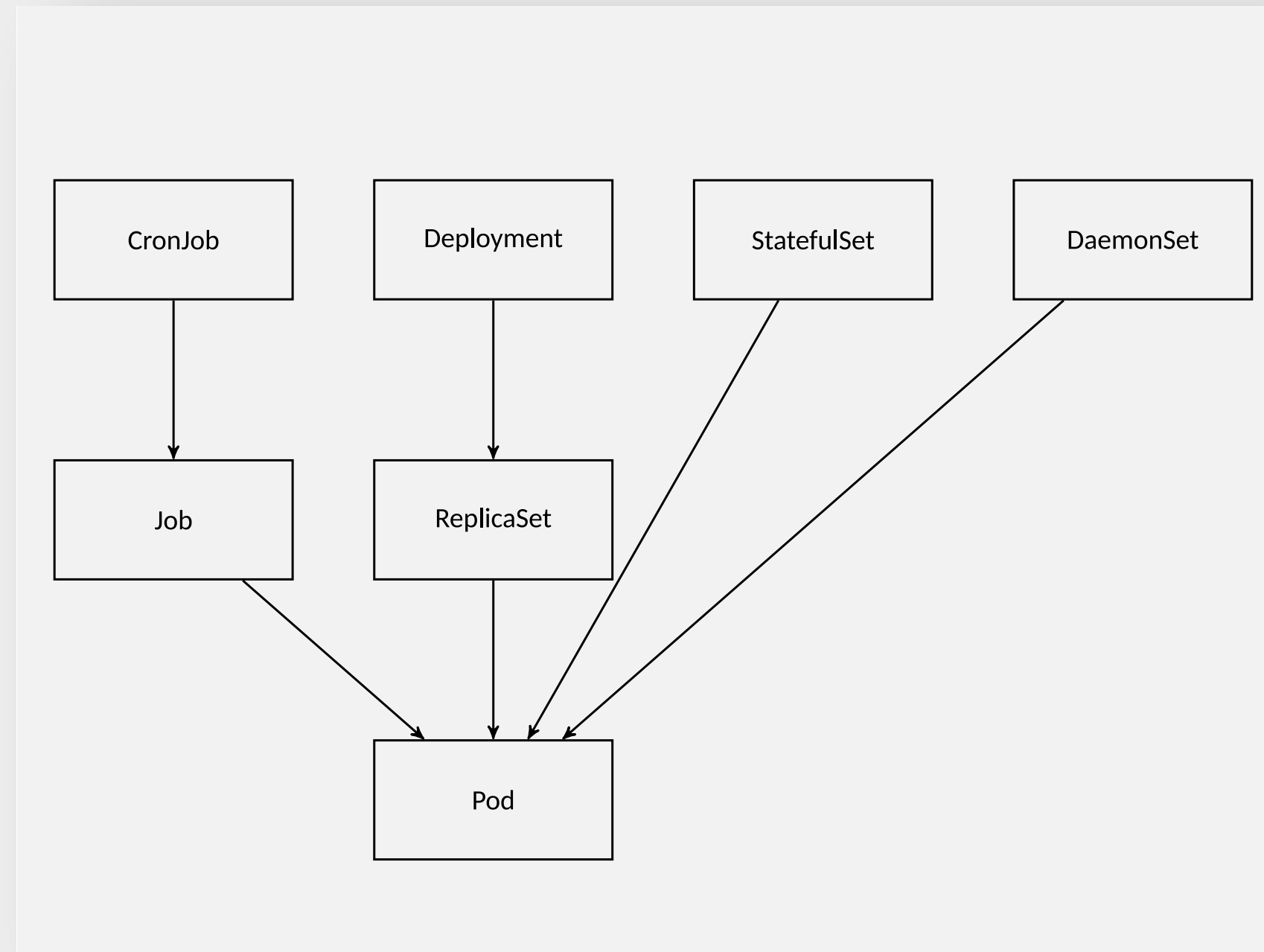
Pod Creation by Jobs/CronJobs



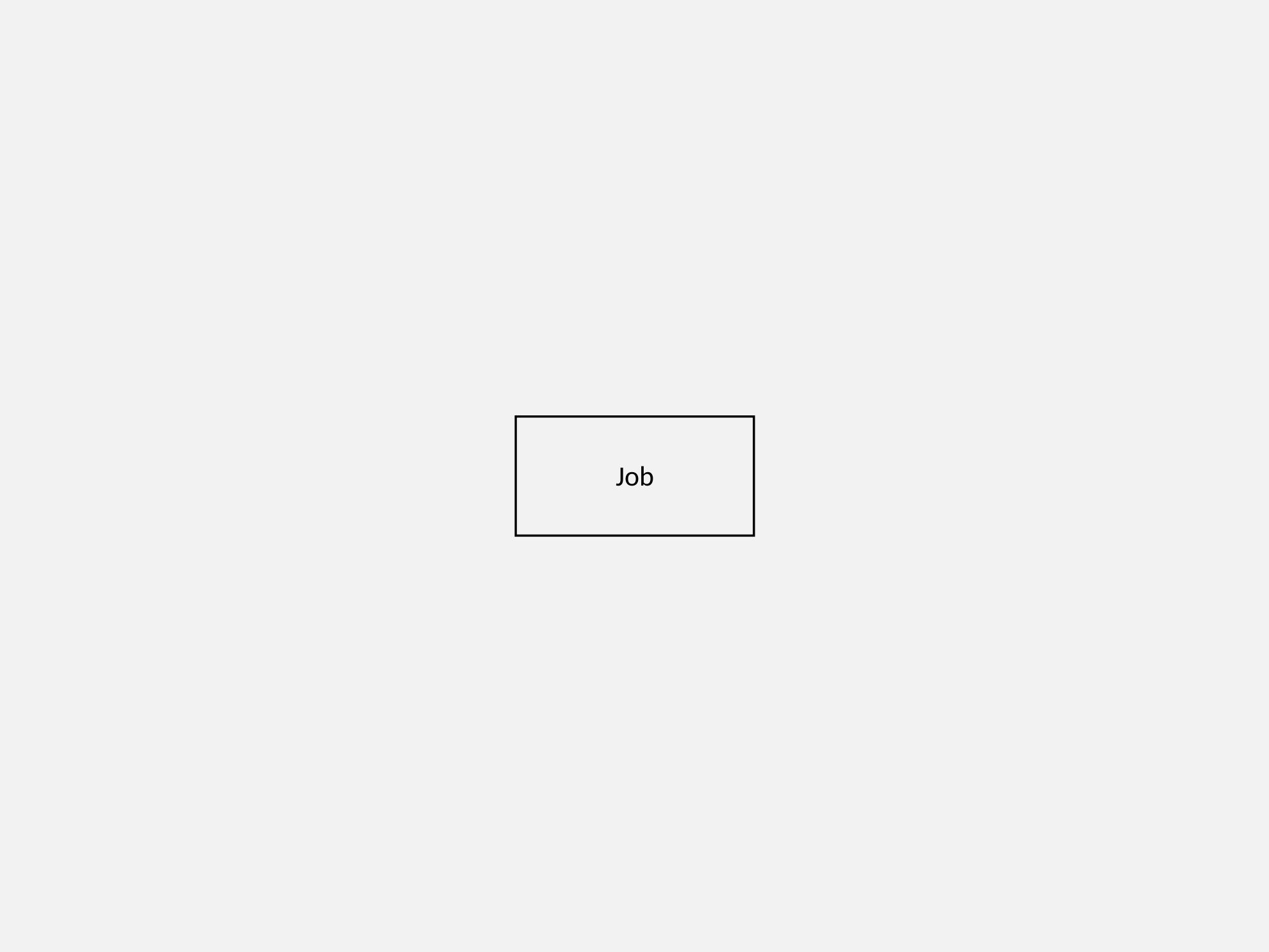
Pod Creation by Jobs/CronJobs



Pod Creation by Jobs/CronJobs

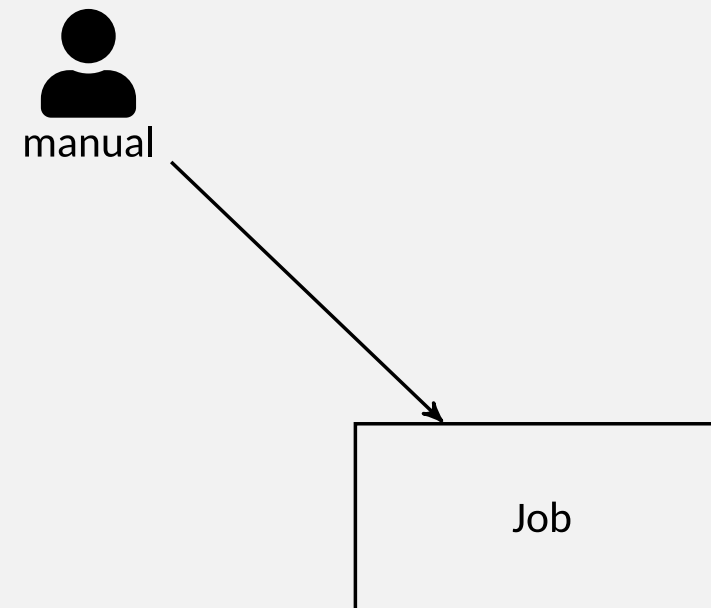


Jobs

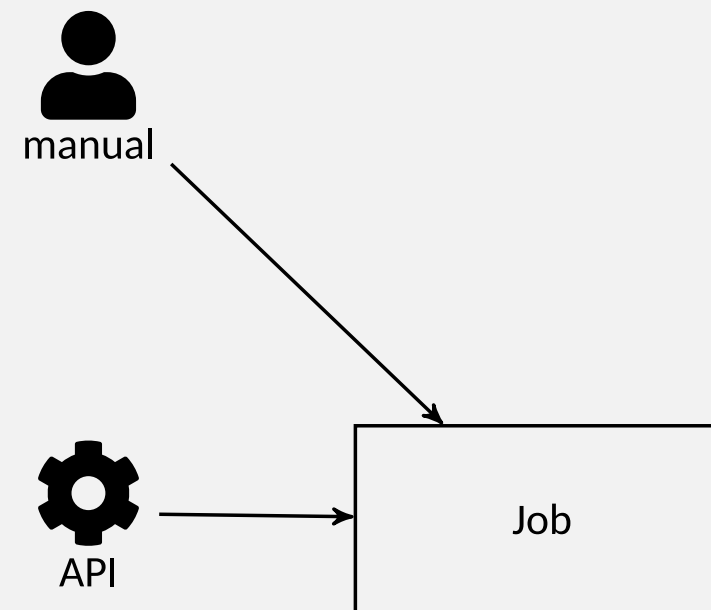


Job

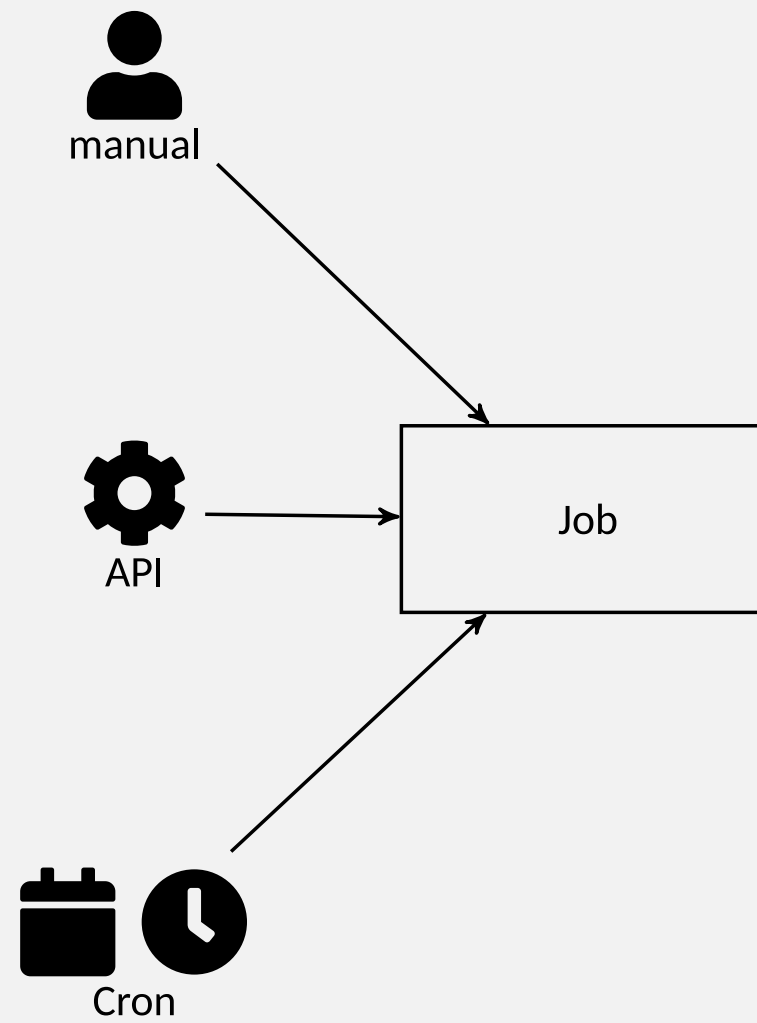
Jobs



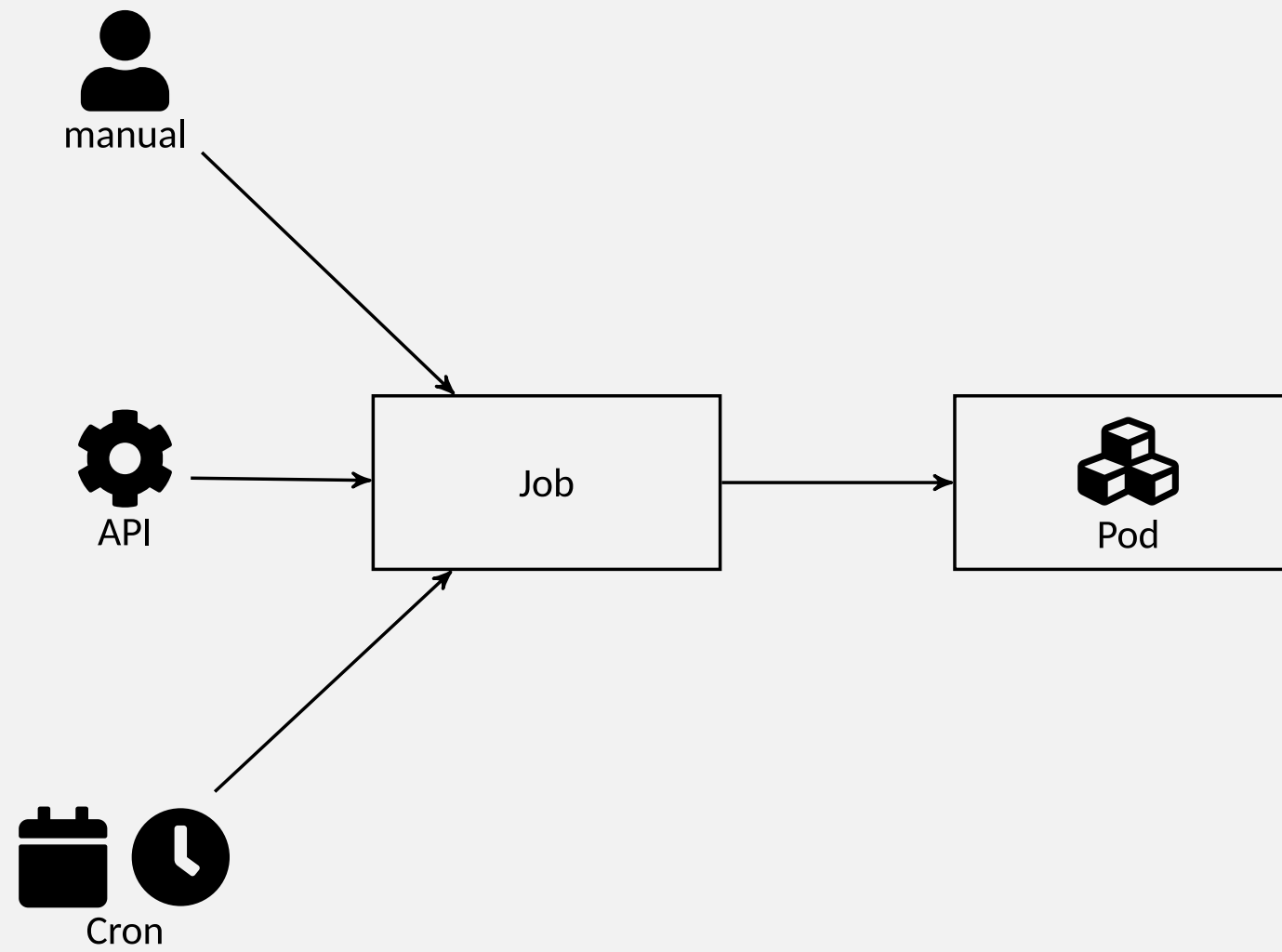
Jobs



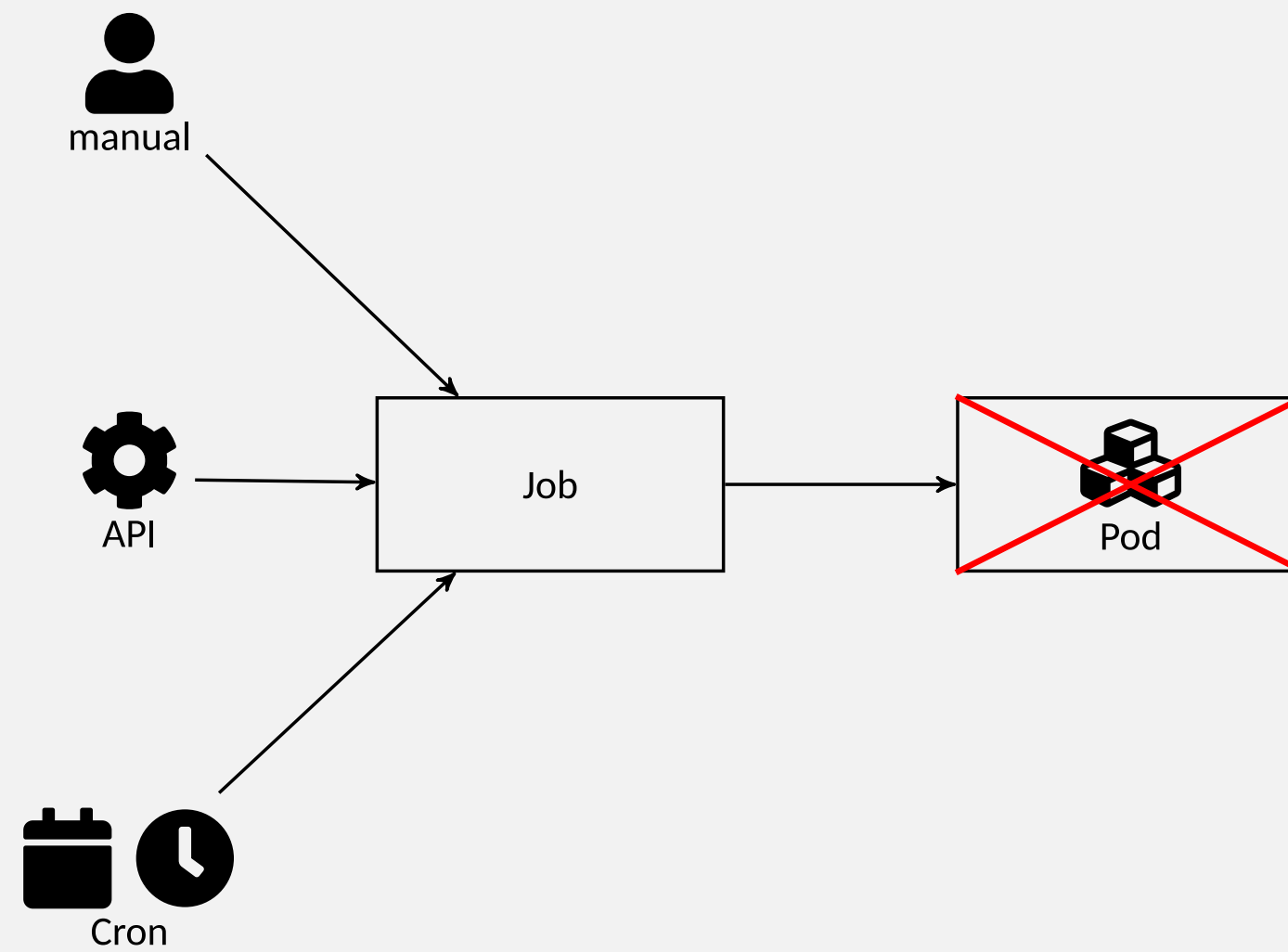
Jobs



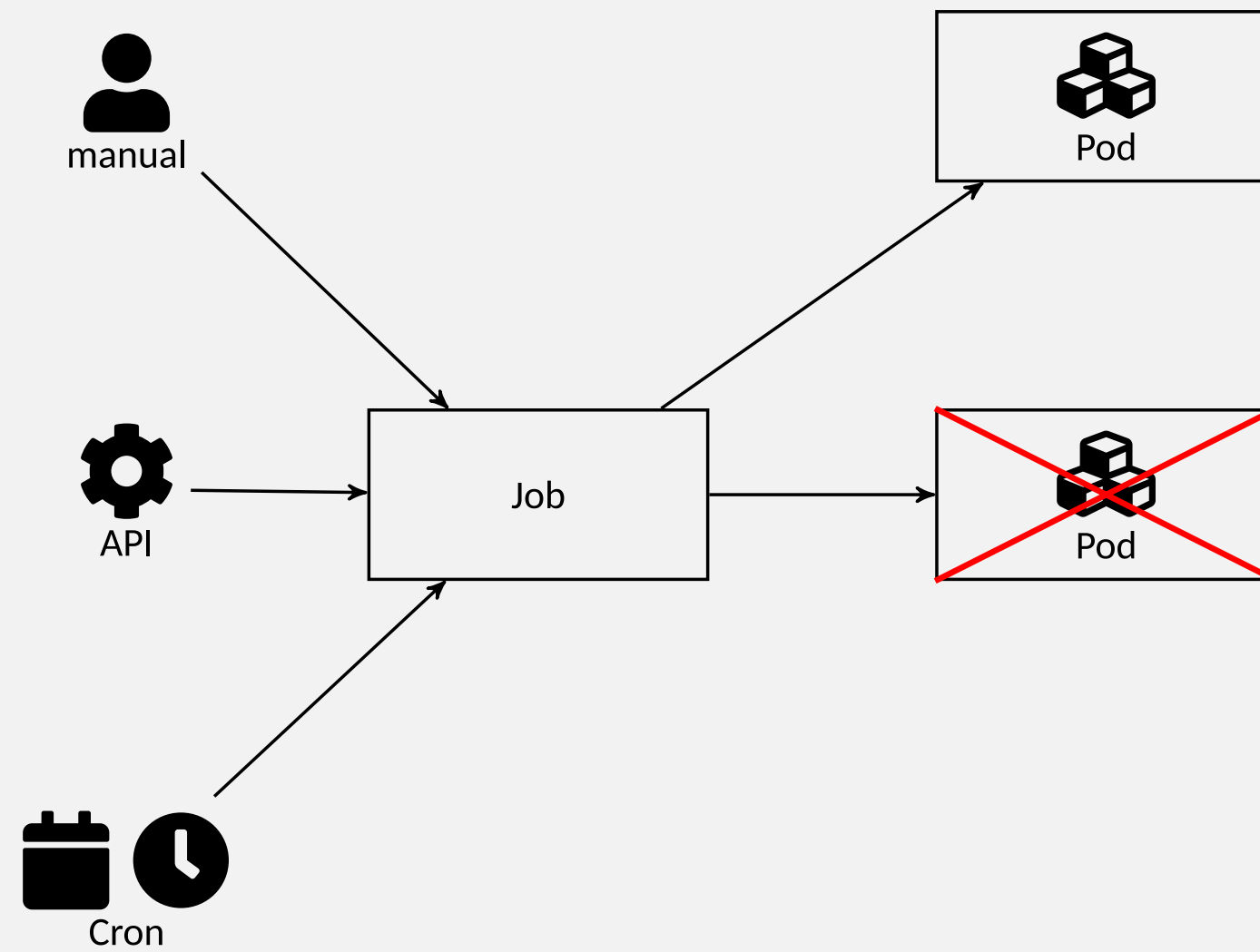
Jobs



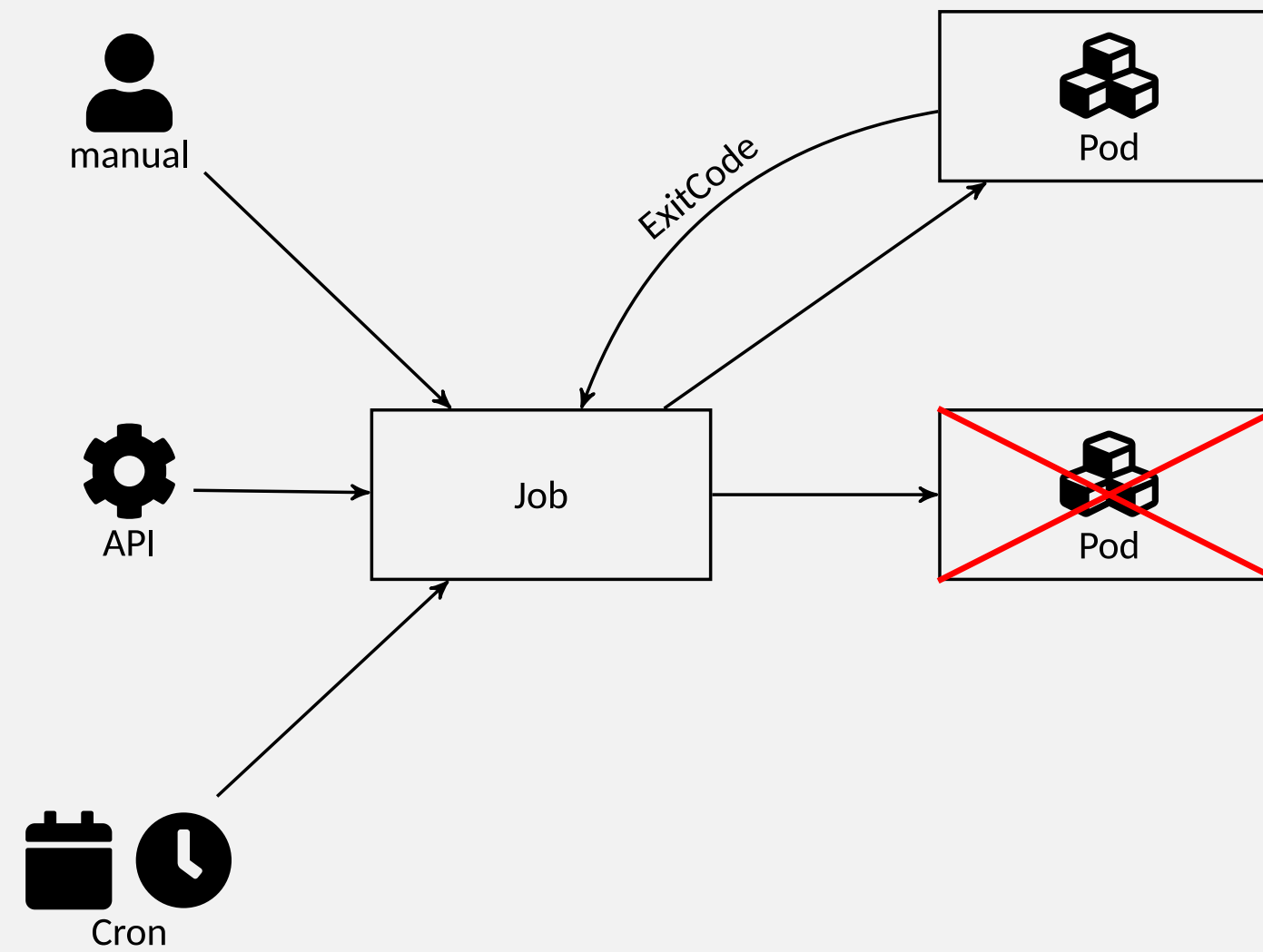
Jobs



Jobs

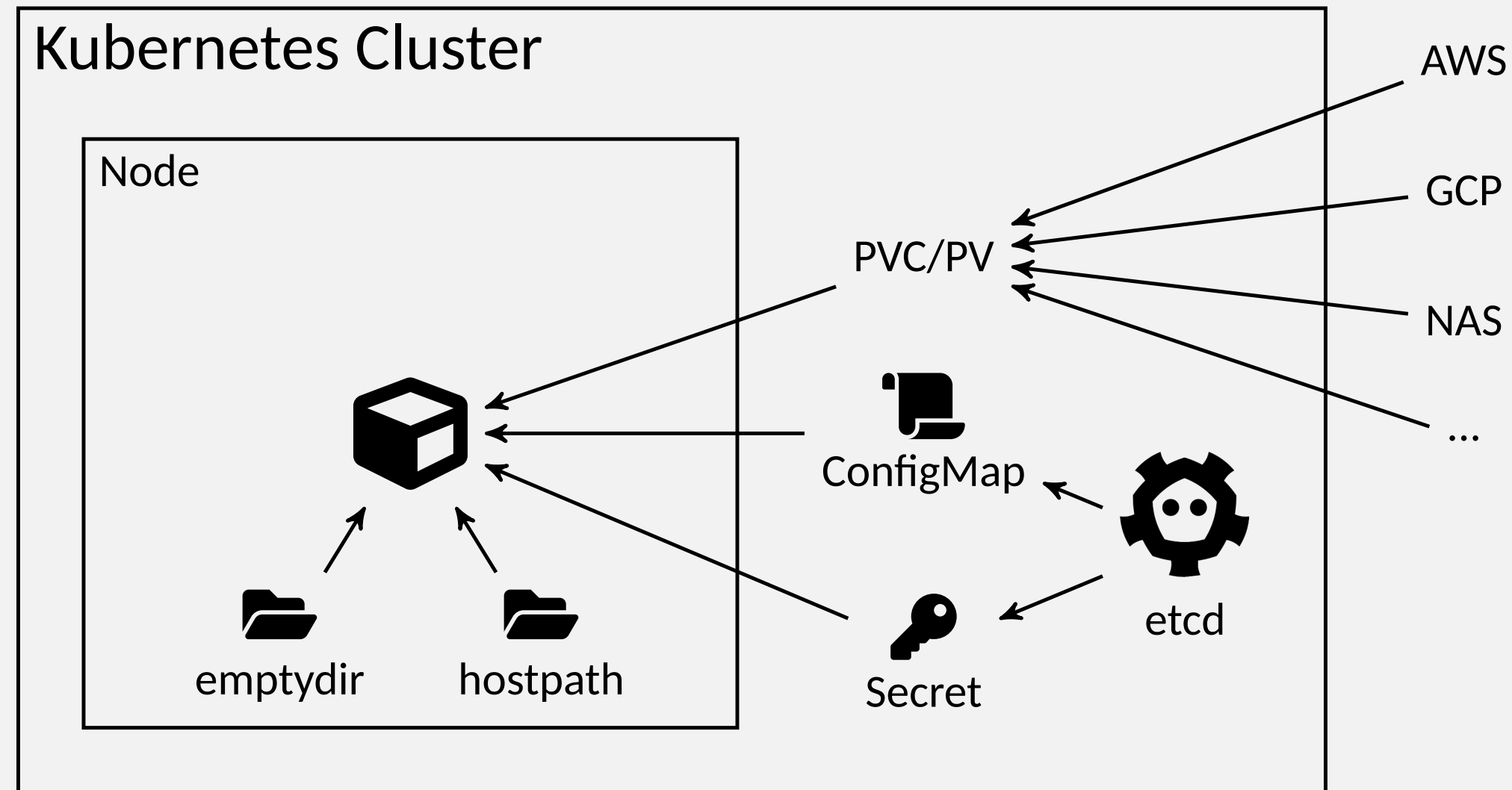


Jobs



Volumes

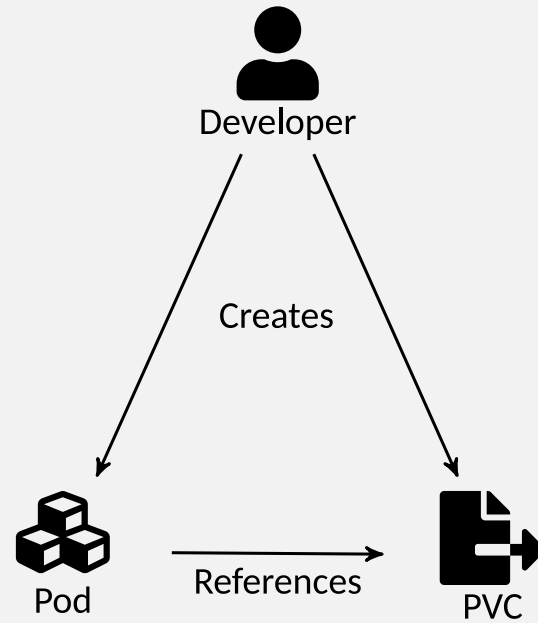
Volumes - Übersicht



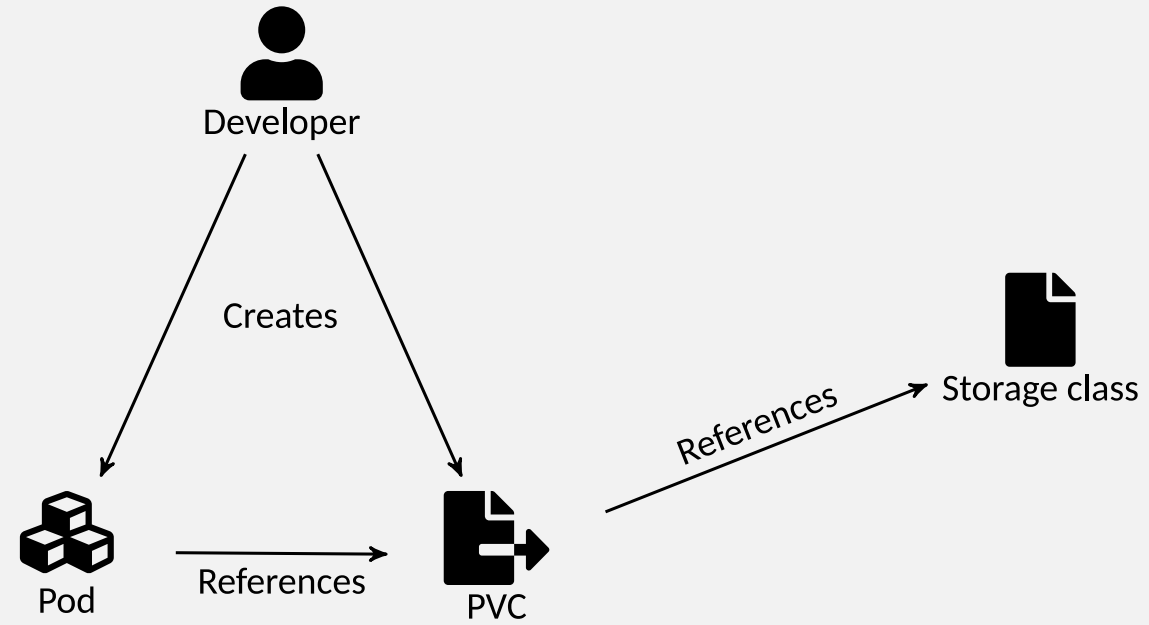
PVC Interna



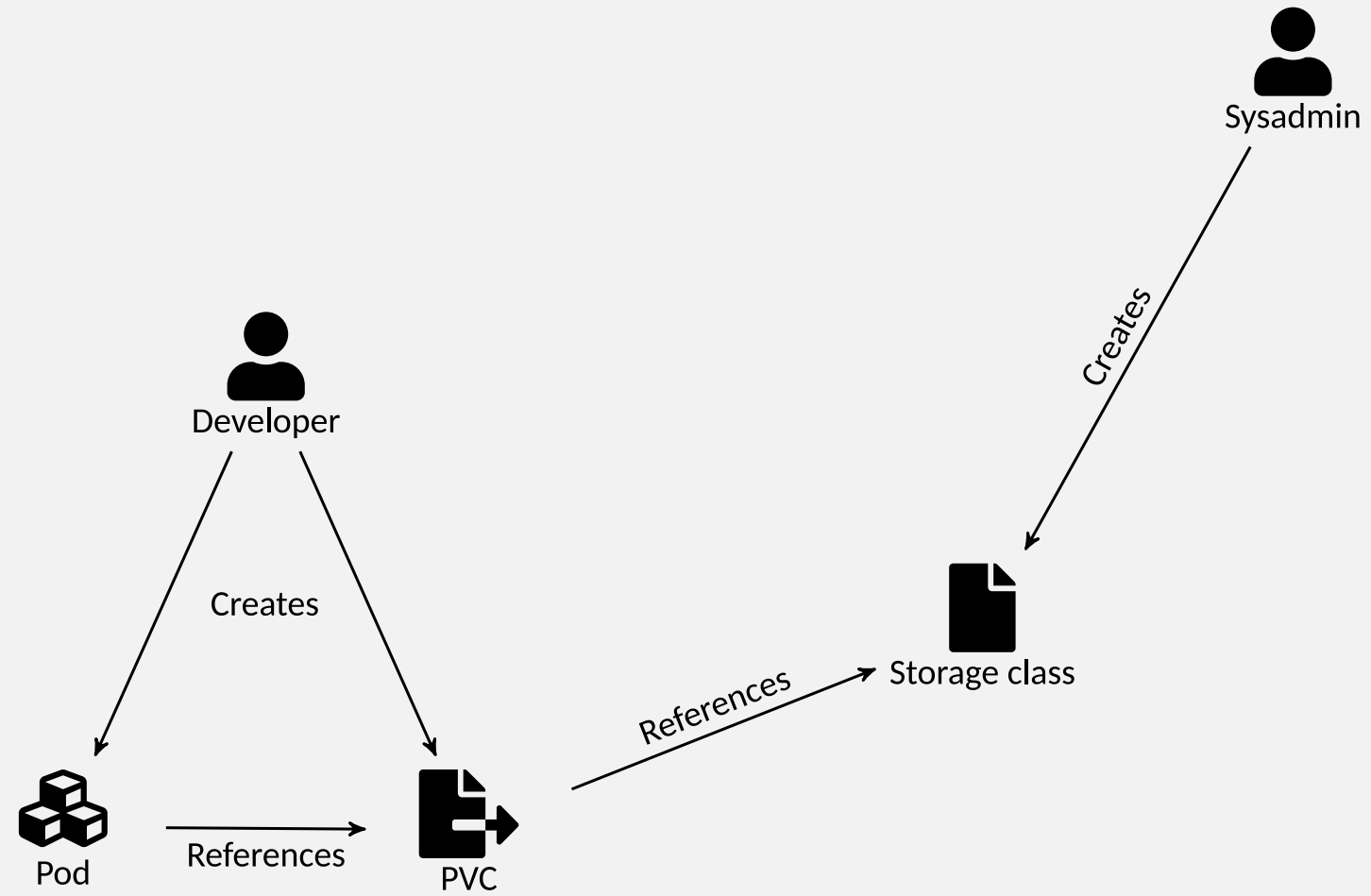
PVC Interna



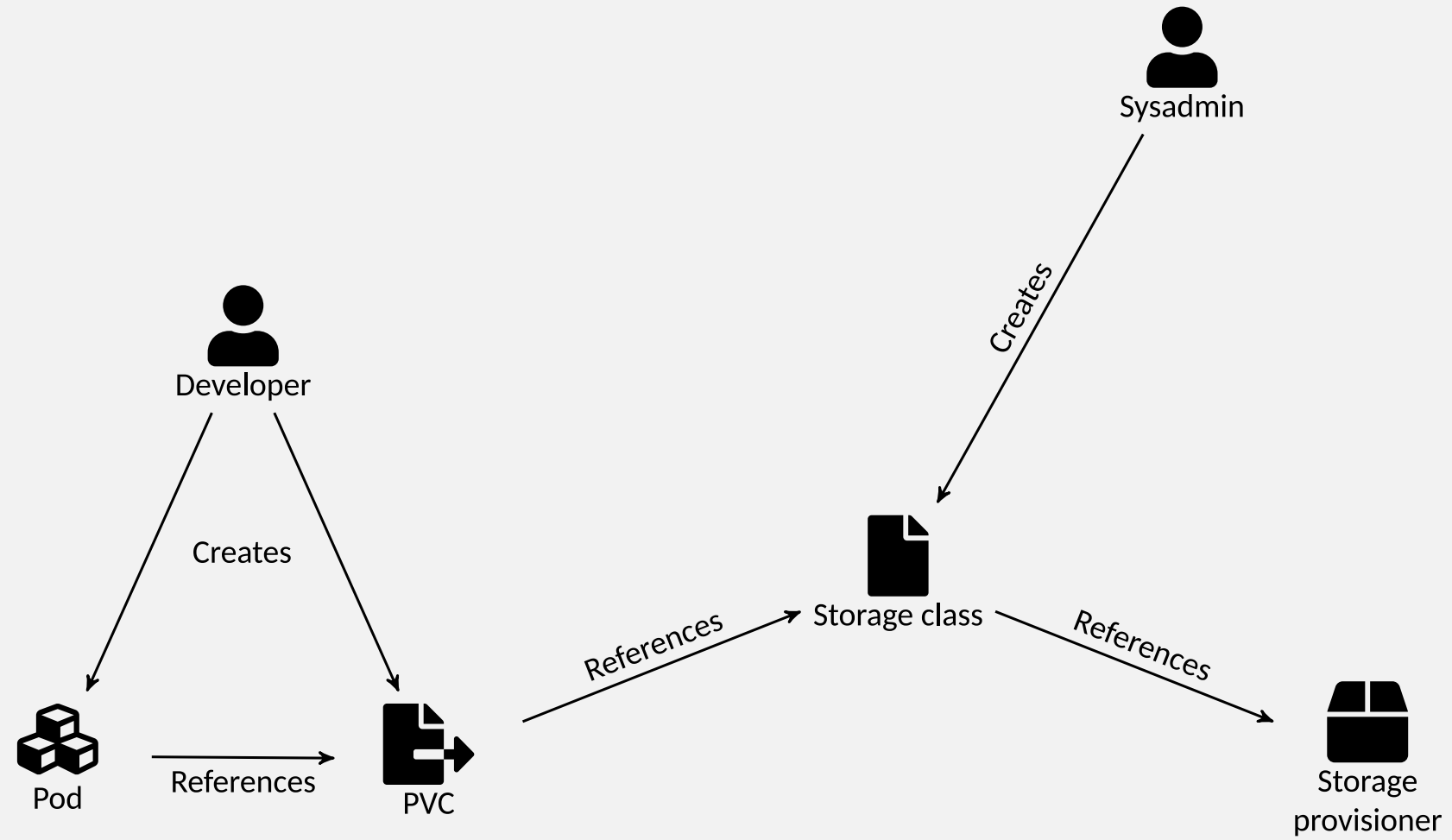
PVC Interna



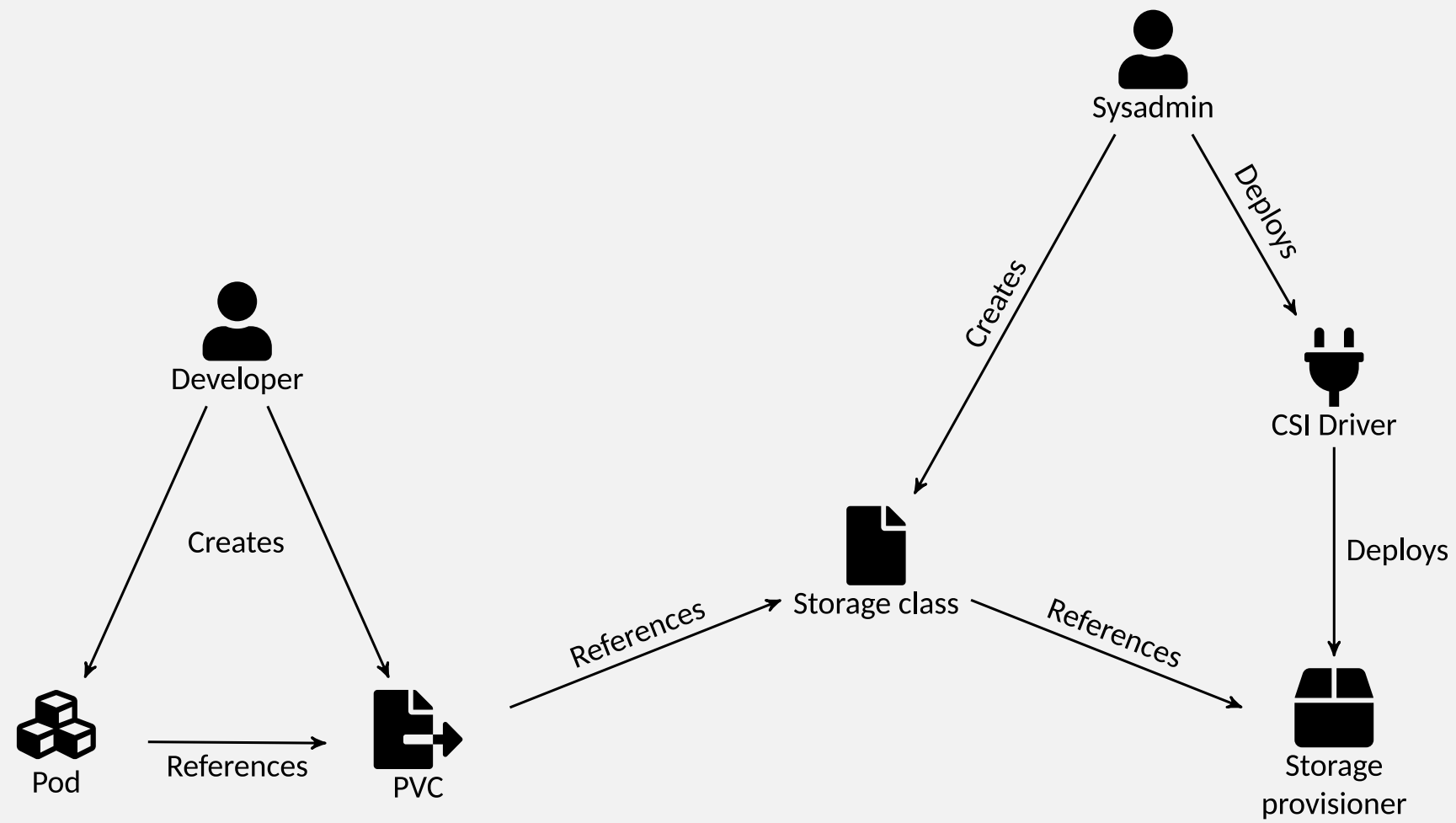
PVC Interna



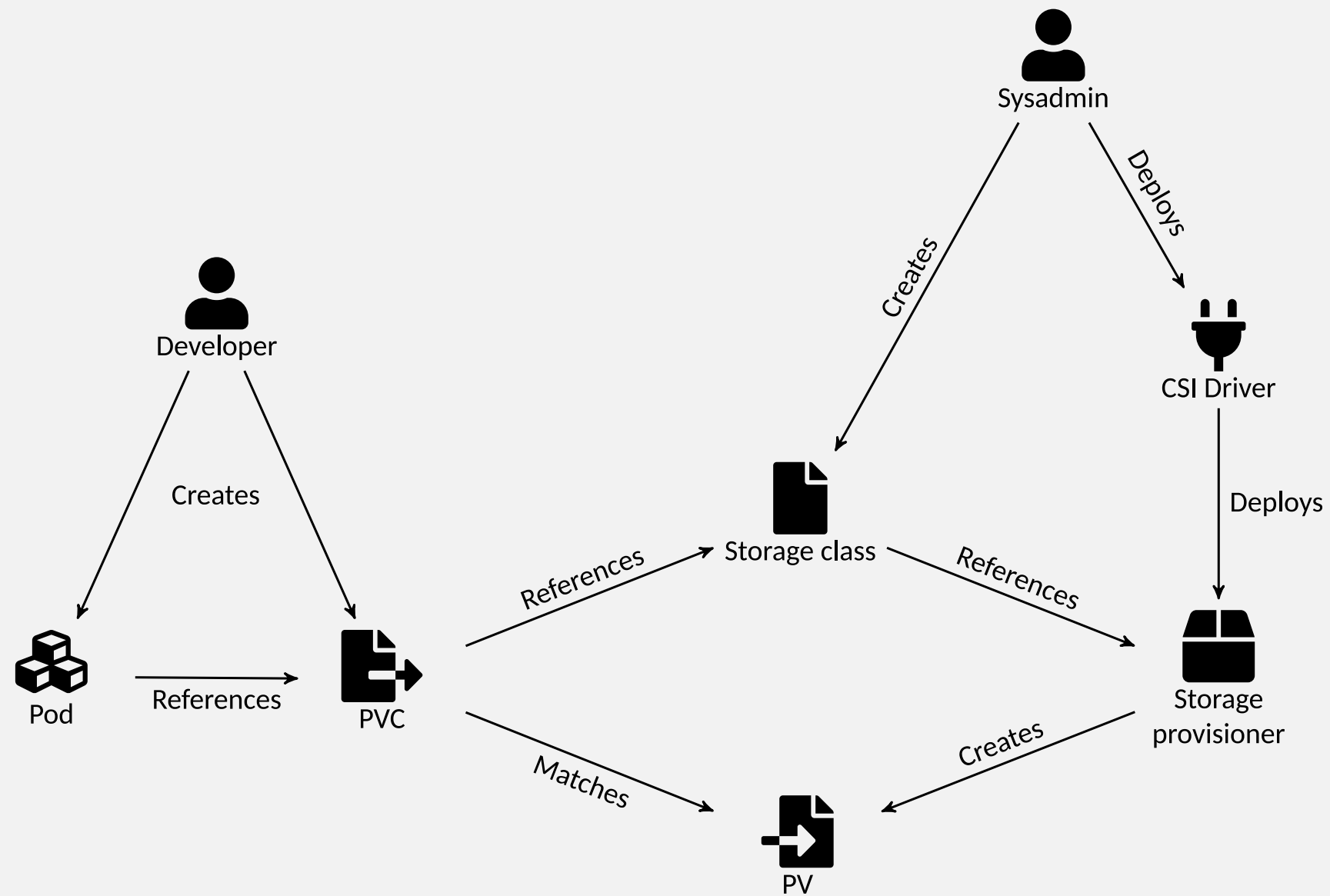
PVC Interna



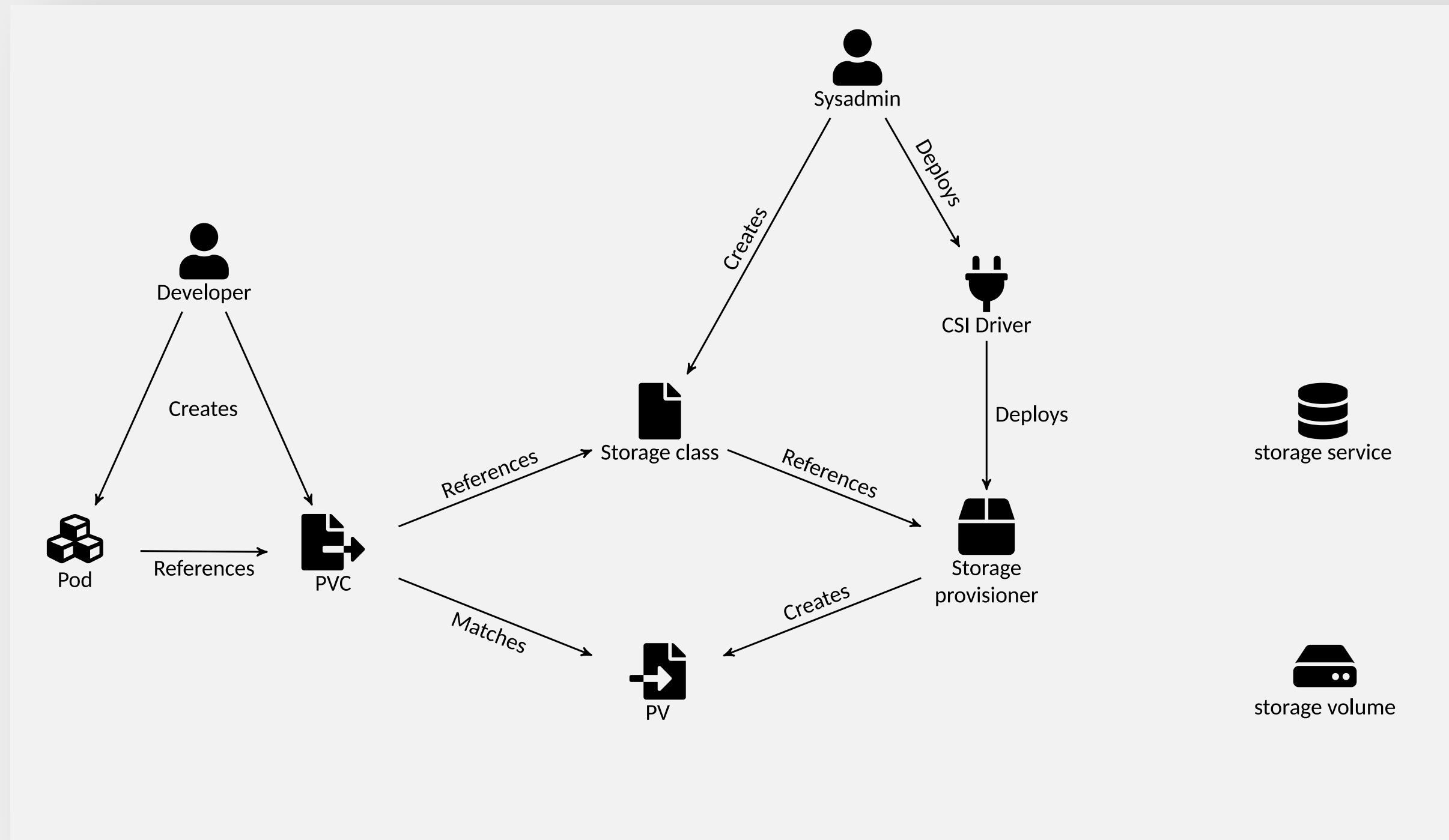
PVC Interna



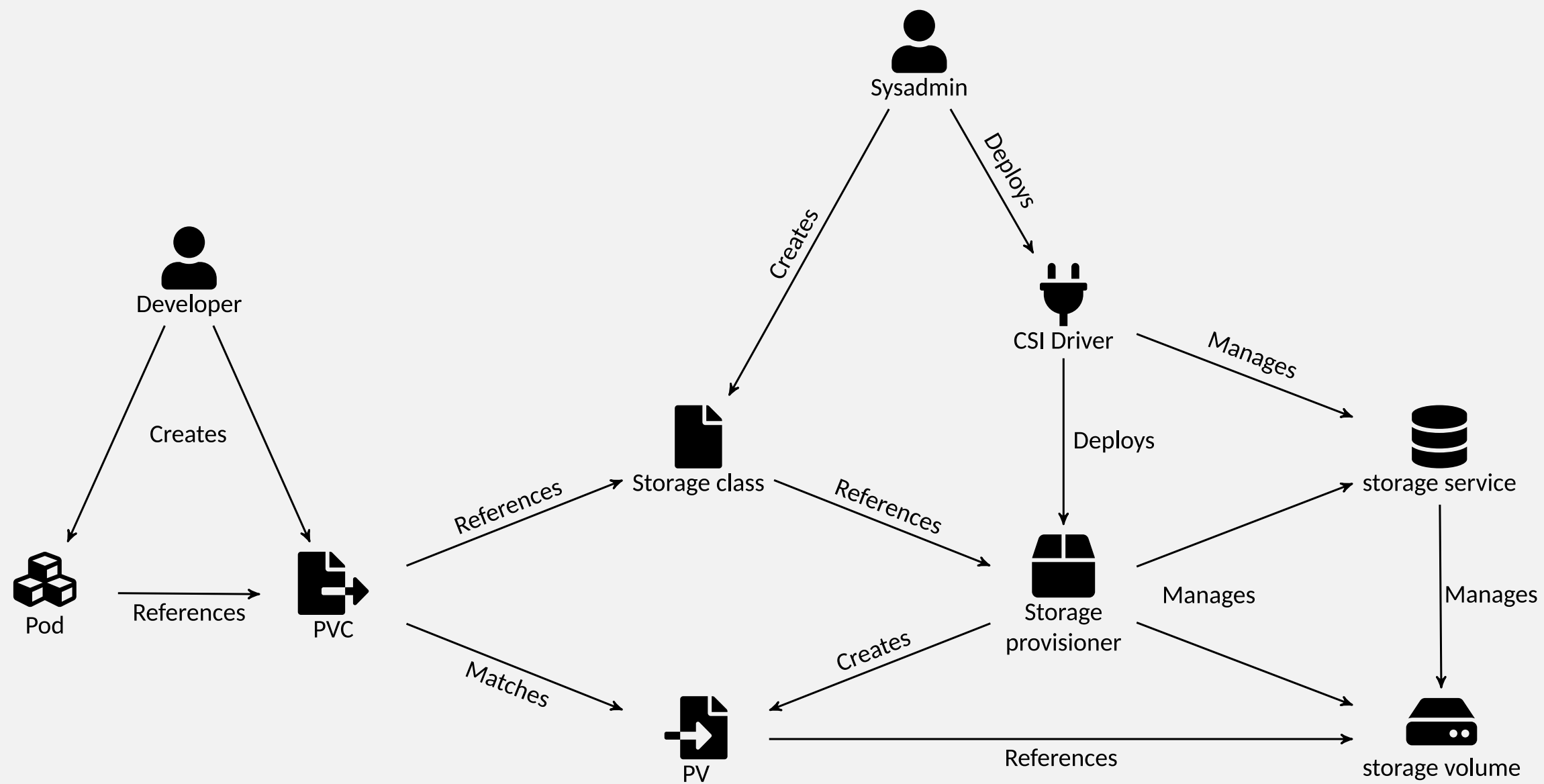
PVC Interna



PVC Interna

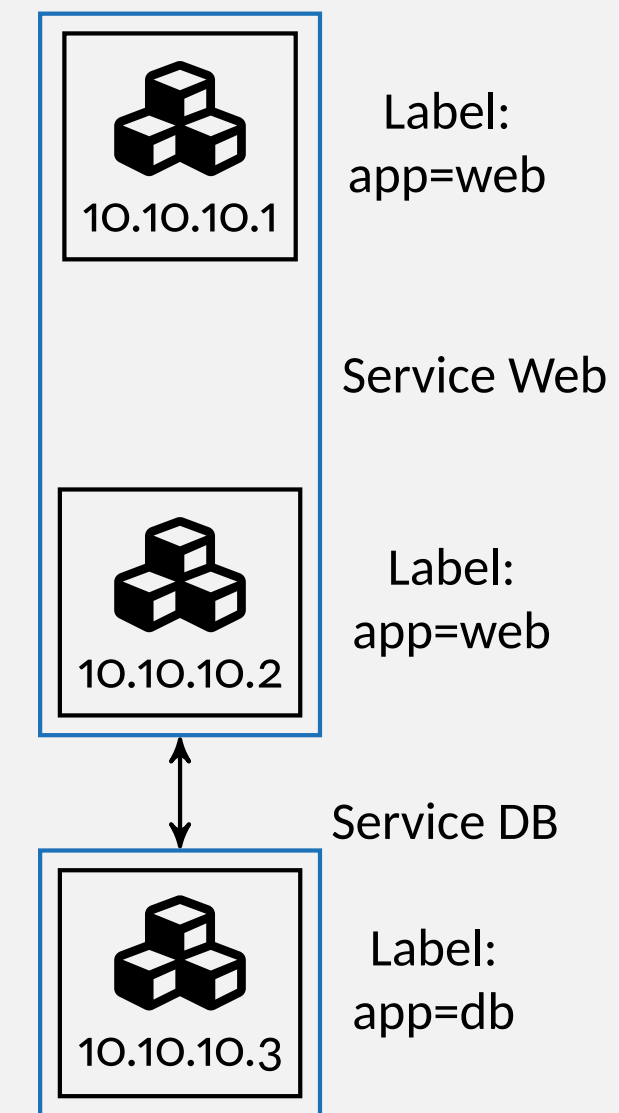


PVC Interna

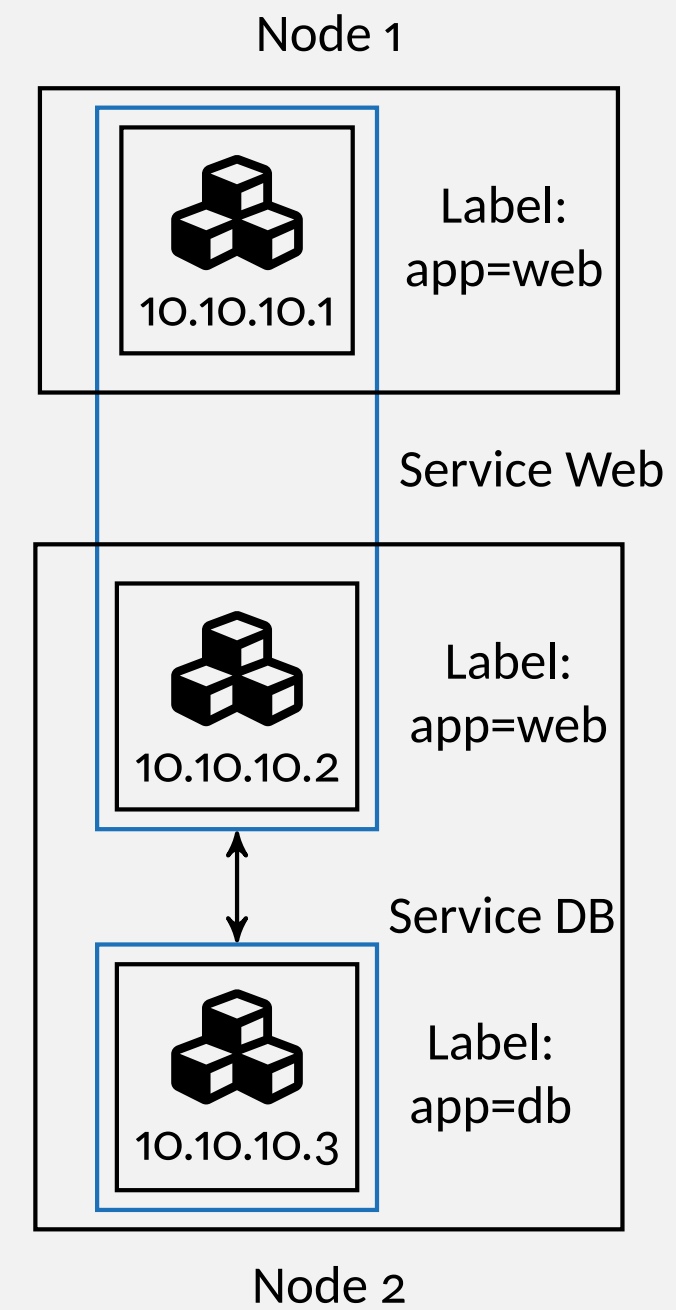


Kubernetes Netzwerk

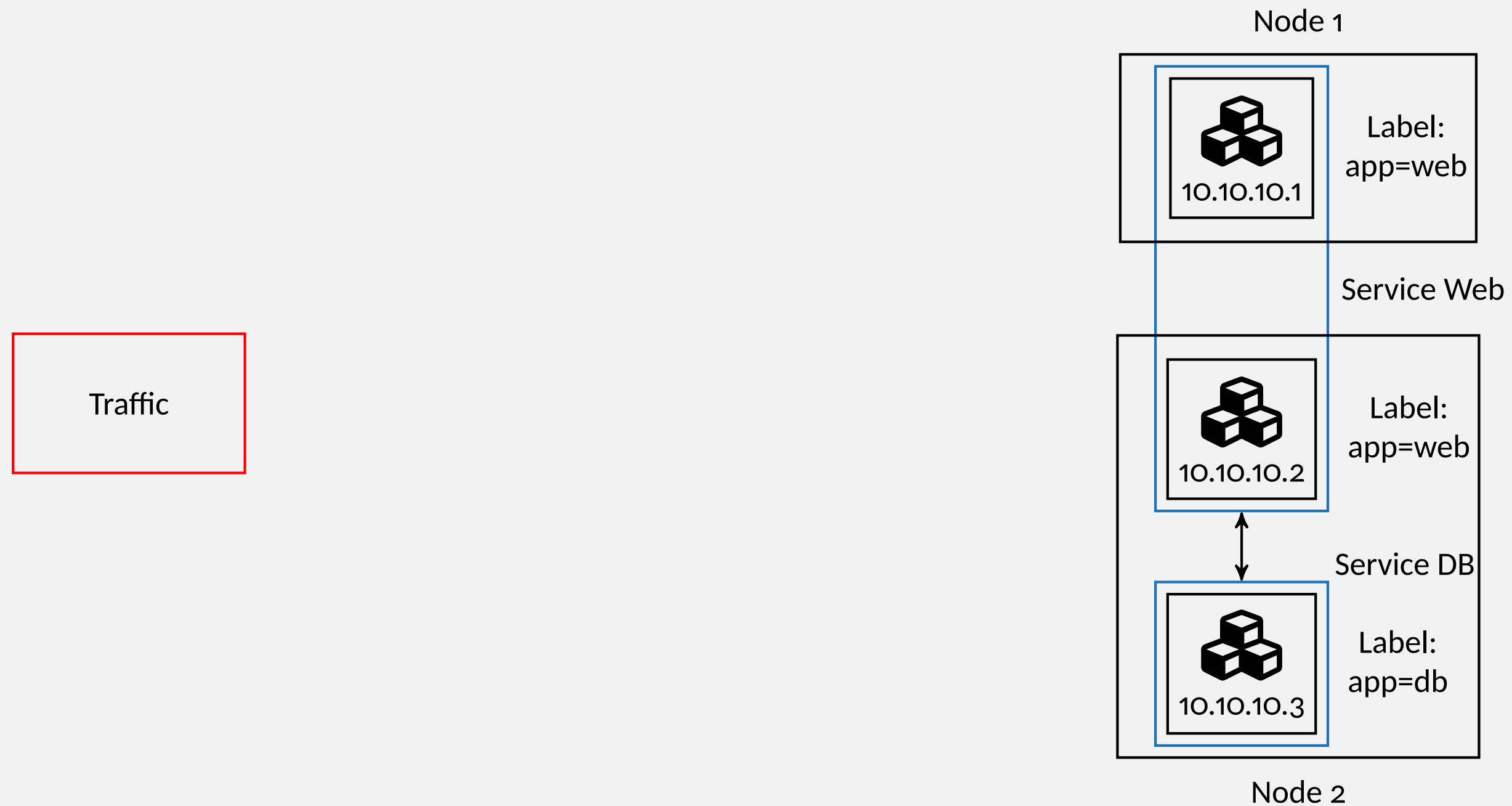
Zugriff von außerhalb



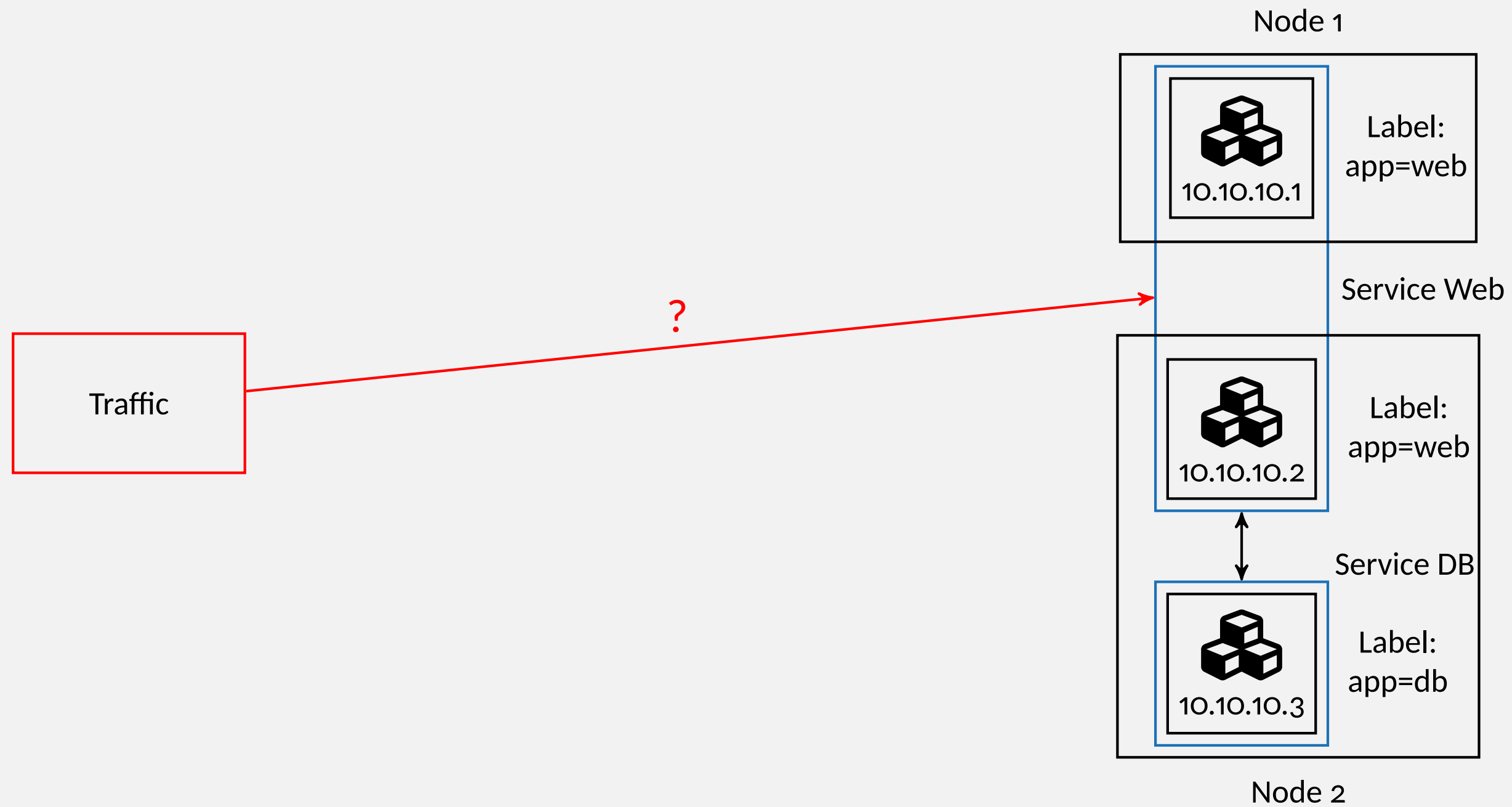
Zugriff von außerhalb



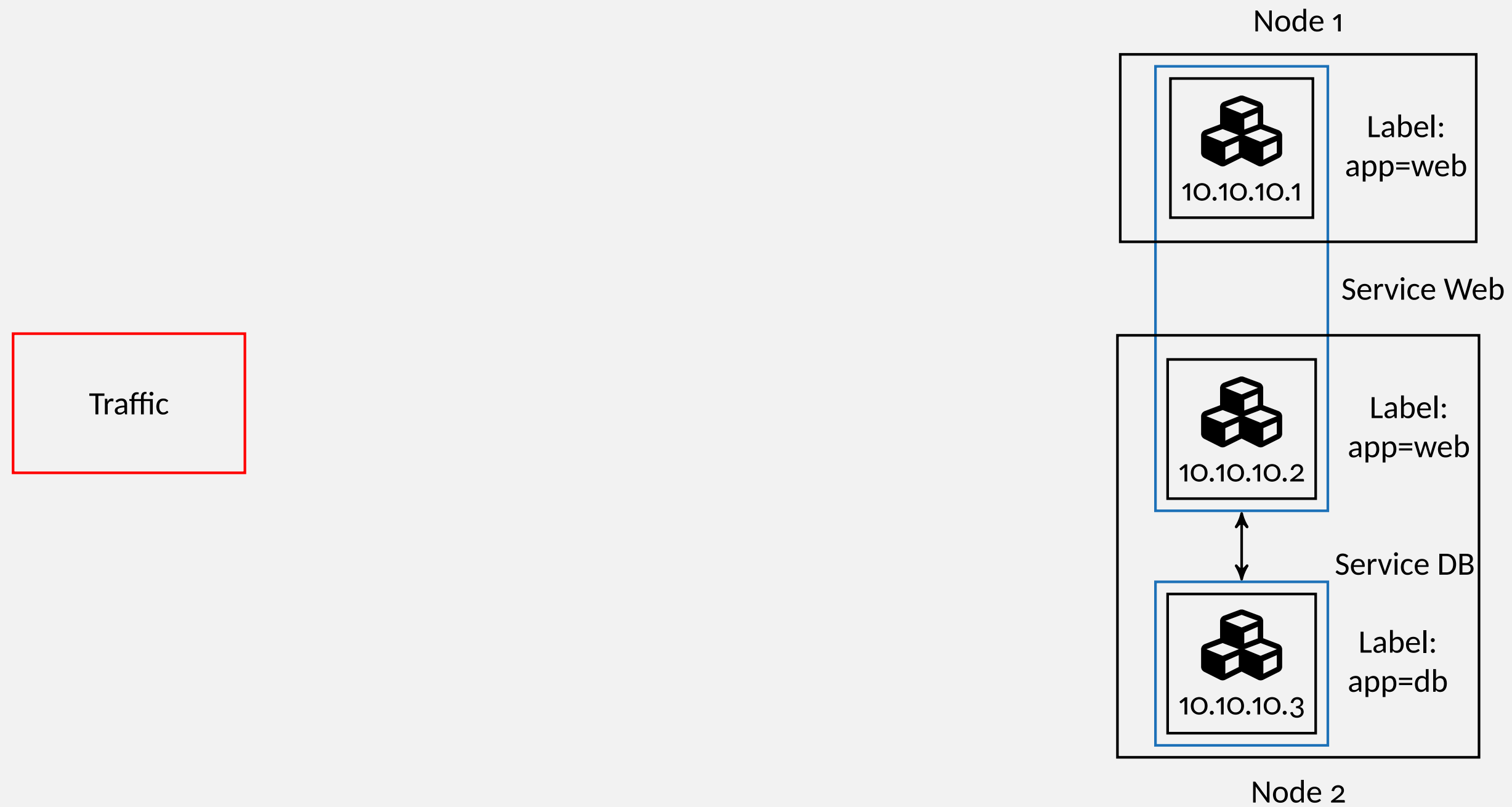
Zugriff von außerhalb



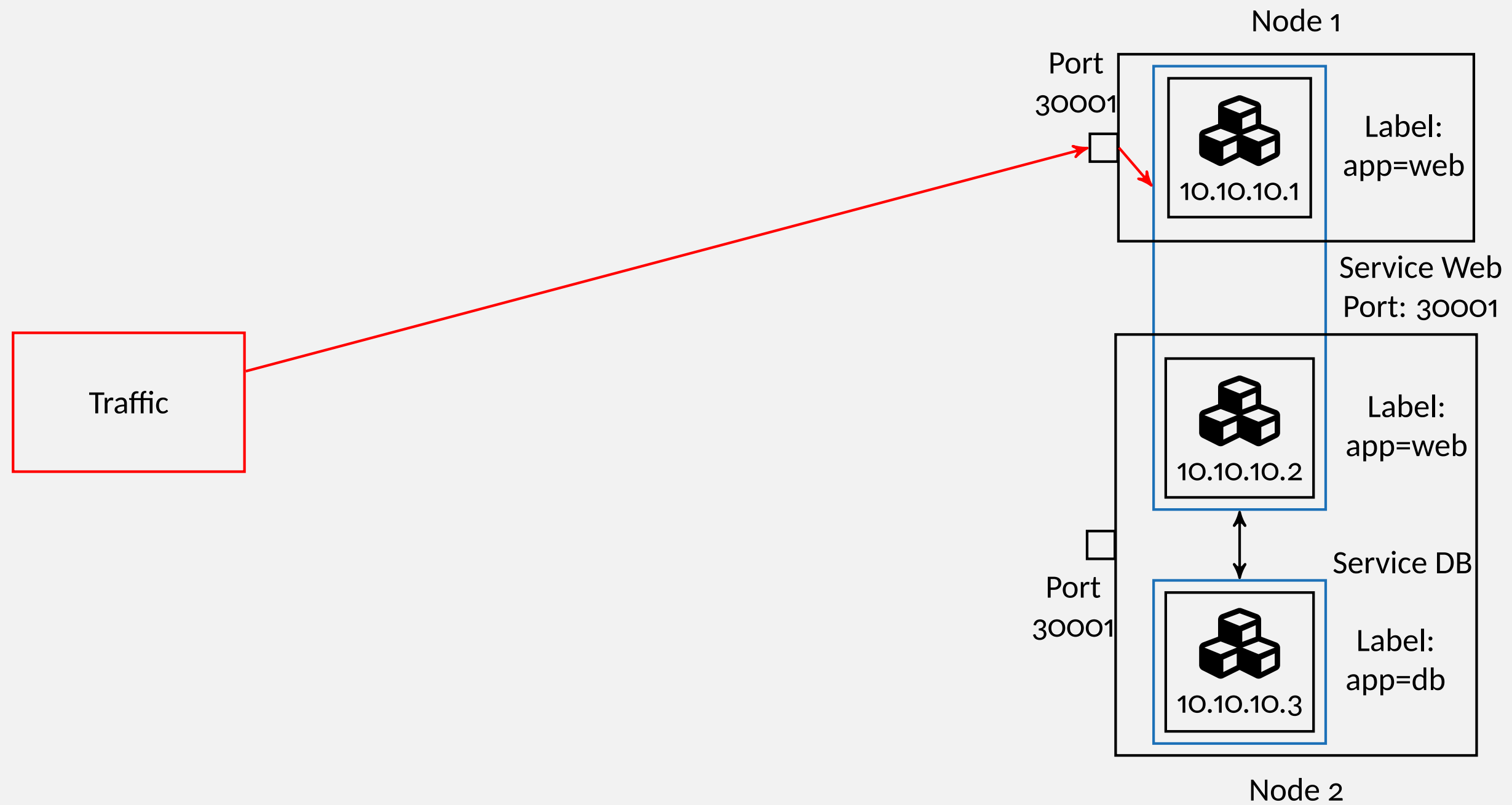
Zugriff von außerhalb



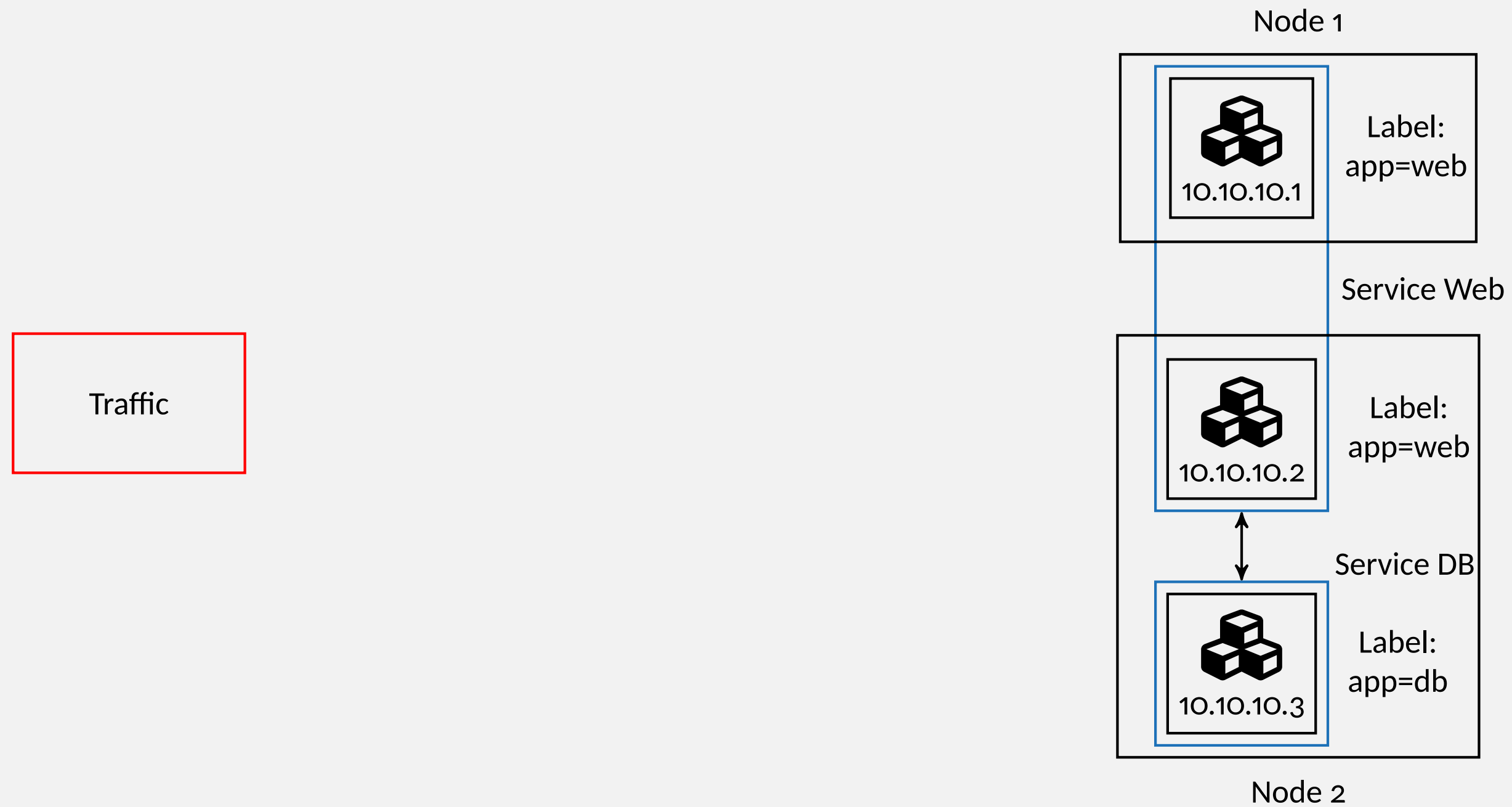
NodePort



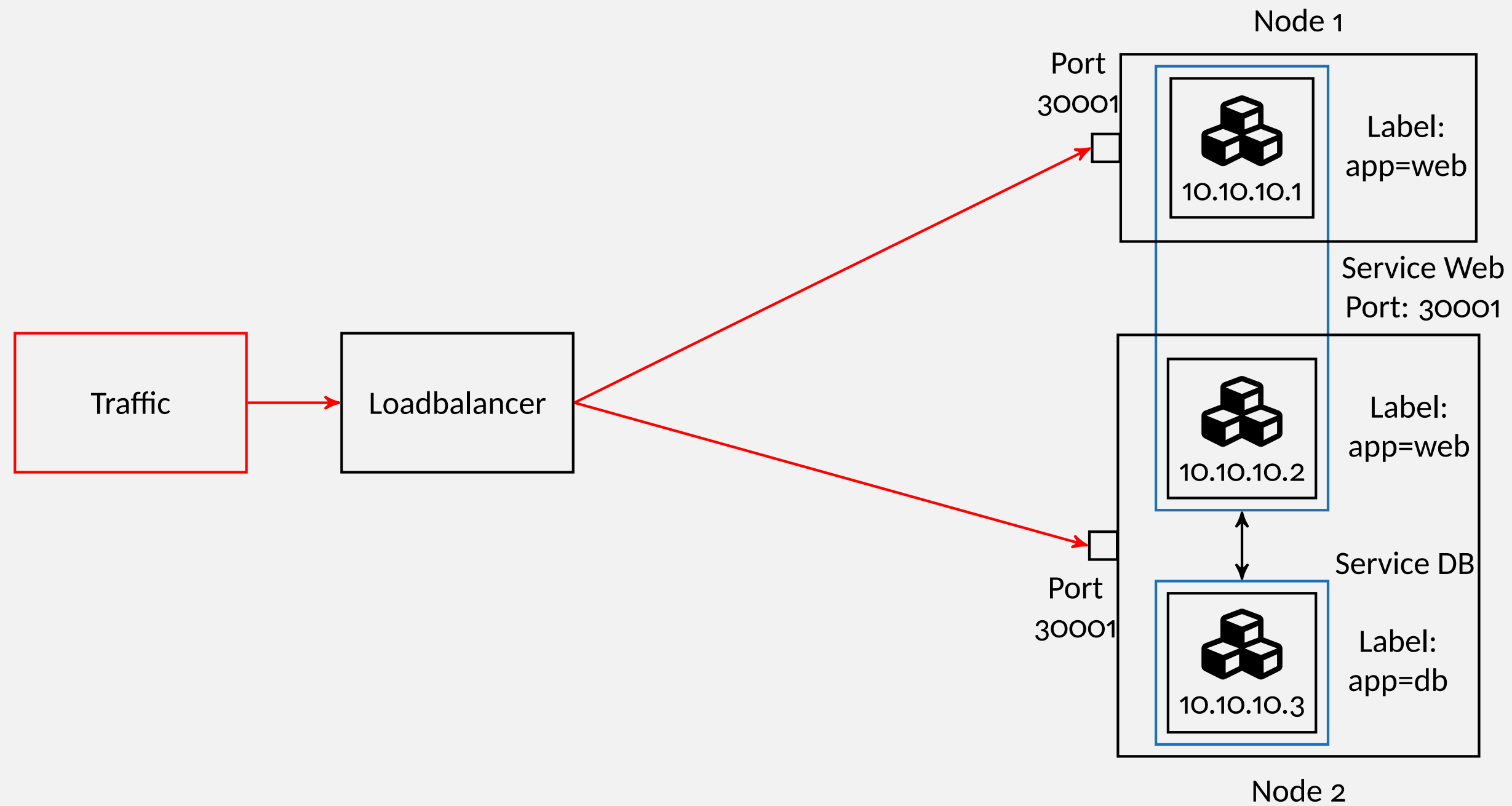
NodePort



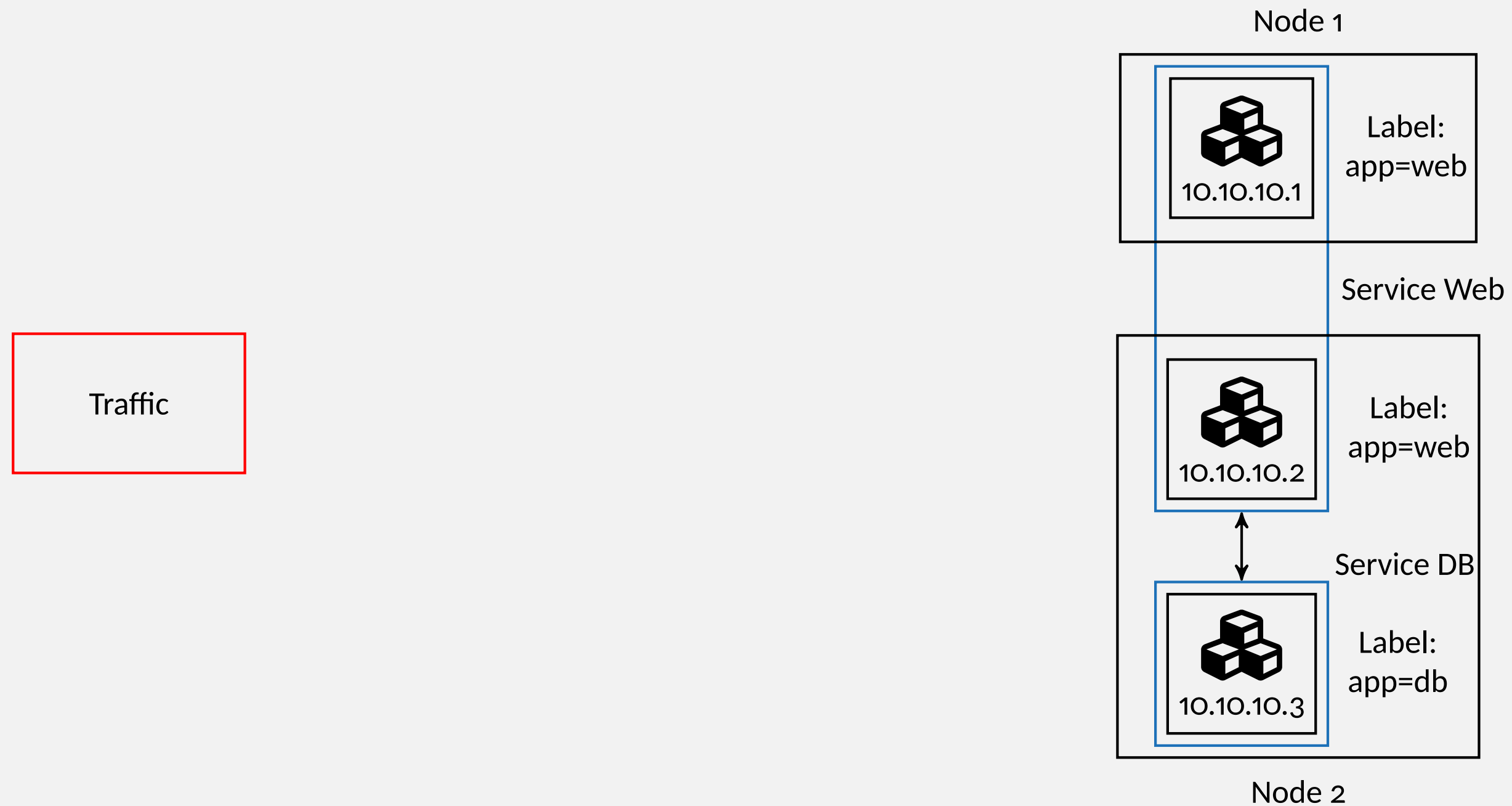
Loadbalancer



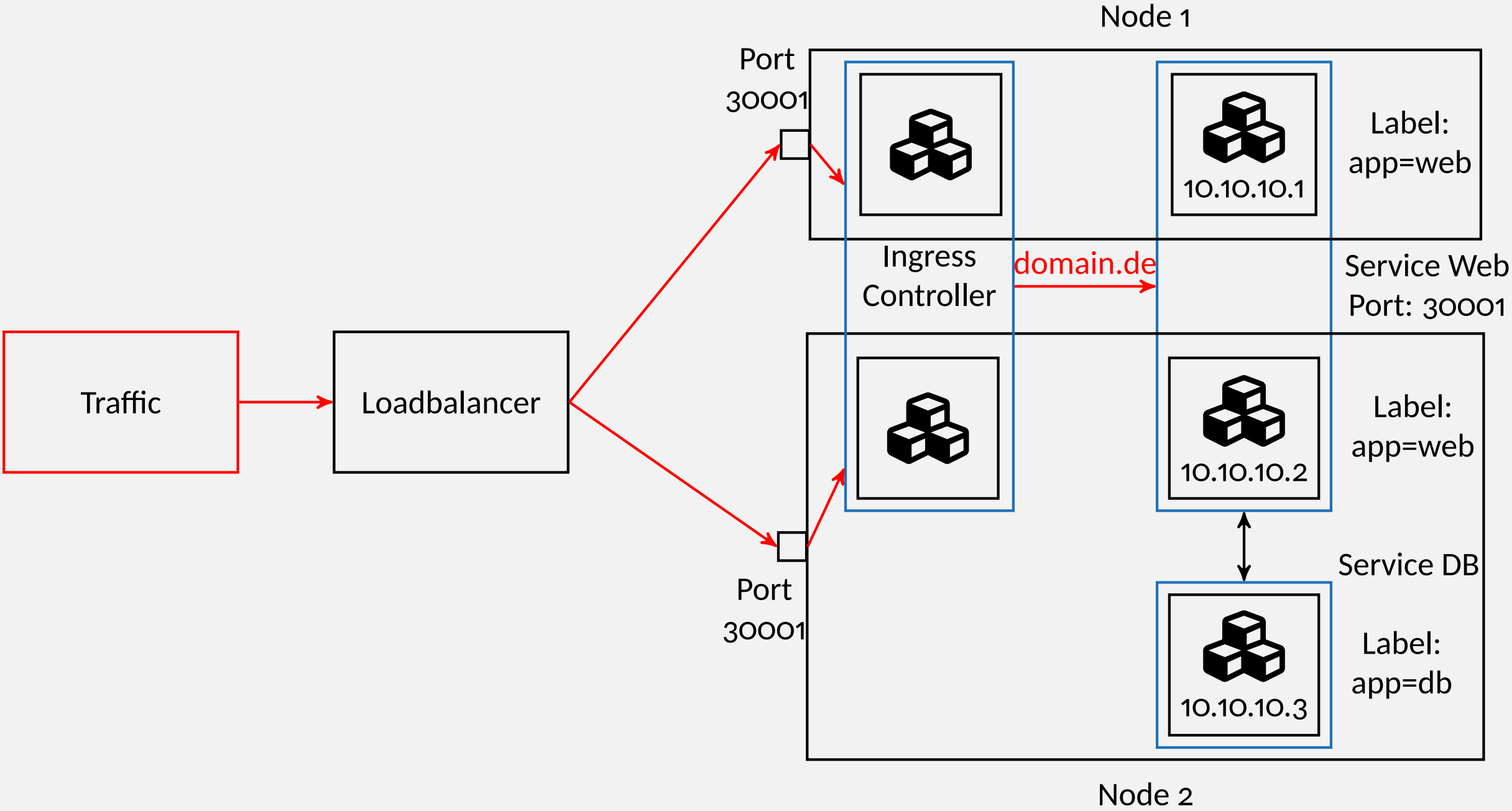
Loadbalancer



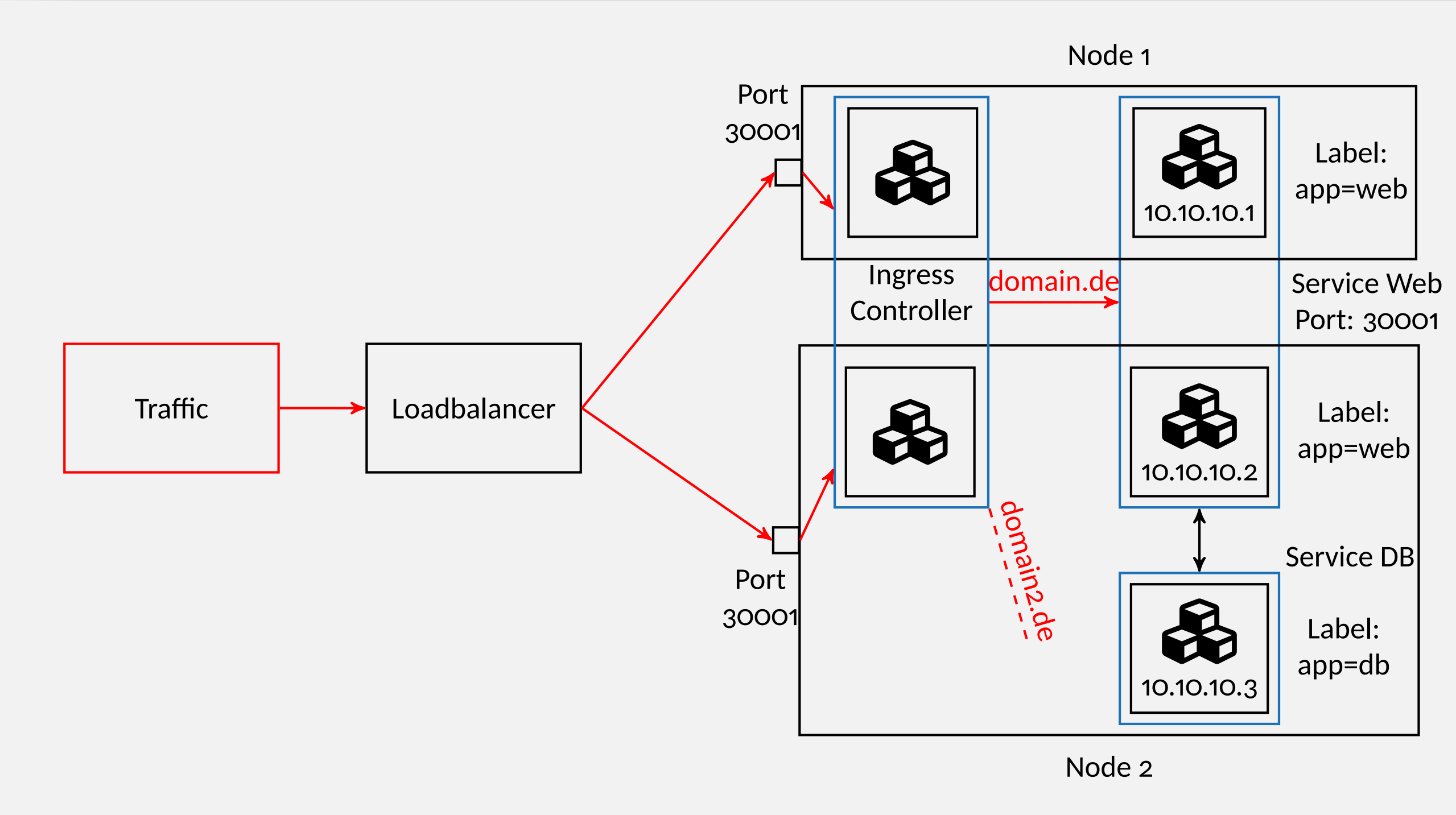
Ingress Controller



Ingress Controller



Ingress Controller



Pod Status

Pod Lifecycle

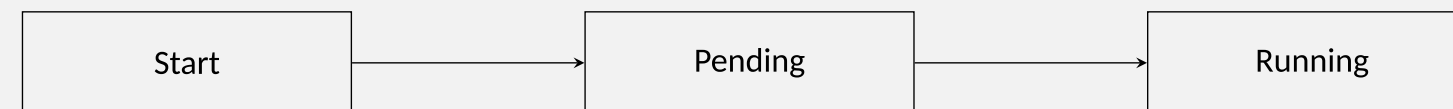
Start

Pod Lifecycle - Pending



Pod wird gescheduled und Images heruntergeladen

Pod Lifecycle - Running



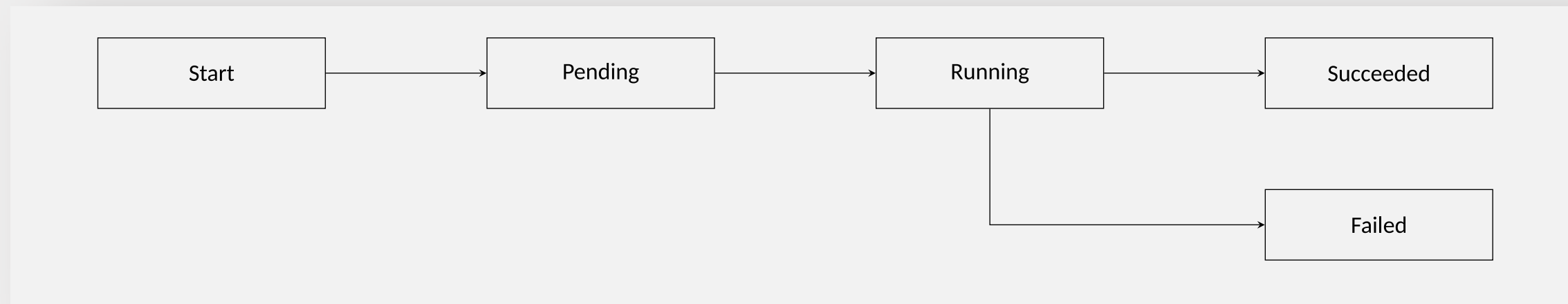
Min. ein Container läuft

Pod Lifecycle - Succeeded



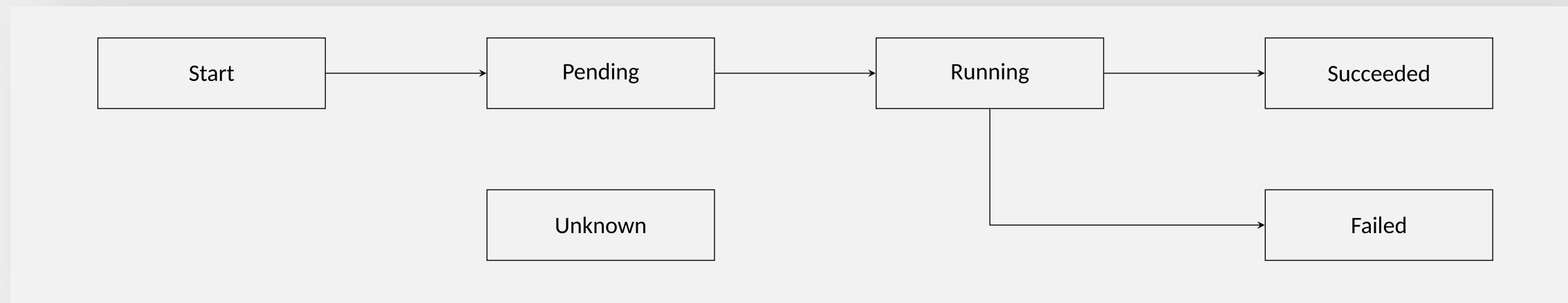
Alle Container im Pod wurden erfolgreich beendet

Pod Lifecycle - Failed



Alle Container wurden beendet. Min. einer mit Exitcode $\neq 0$

Pod Lifecycle - Unknown



Status kann nicht ermittelt werden

Pod Errors:

- `FailedScheduling`
- `ImagePullBackOff`
- `CrashLoopBackOff`
- ...

Pod Errors: `FailedScheduling`

- Pod kann nicht gescheduled werden
- Pod hängt in `Pending`
- Häufig verursacht durch:
 - Nicht genügend Ressourcen
 - Fehlkonfiguration bei Taints, (Anti-)Affinity, Nodes
 - Probleme mit PersistentVolumeClaims

Pod Errors: `ImagePullBackOff`

- Image kann nicht geladen werden
- Wiederholte Versuche mit exponentiellem Backoff
- Häufig verursacht durch:
 - Falscher Image-Name/Tag
 - Netzwerkproblemen
 - Fehlende Authentifizierung

Images von privaten Registries

- Kein Zugriff auf die Registry => `ImagePullBackOff`
- Rate-Limit => `ImagePullBackOff`
 - z.B. DockerHub: 100 anonyme Pulls in 6h
- Default: Pull ohne Authentifizierung
- Registry mit Authentifizierung => Image Pull Secrets

Image Pull Secrets

Secret erstellen:

```
kubectl create secret docker-registry <secret-name> \
  --docker-server=<registry-url> \
  --docker-username=<username> \
  --docker-password=<password> \
  --docker-email=<email>
```

Secret in Deployment einfügen:

```
spec:
  containers:
  - name: my-container
    image: myregistry.com/myproject/myimage:tag
  imagePullSecrets:
  - name: <secret-name>
```

Pod Errors: `CrashLoopBackOff`

- Pod stürzt ab, wird neu gestartet, stürzt wieder ab
- Neustarts mit exponentiellem Backoff
- Häufig verursacht durch:
 - Fehler im Startup-Command
 - Unbehandelte Exception
 - Fehler in der Liveness-Probe

Fehler finden und beheben

```
# 1. Überblick verschaffen
kubectl get pods

# 2. Status und Events nachschauen
kubectl get pods -o yaml <pod-name>
kubectl describe pod <pod-name>

# 3. Logs ansehen
kubectl logs <pod-name>
```

Probes

Probes - Arten

- `livenessProbe`
- `readinessProbe`
- `startupProbe`

Probes - `livenessProbe`

- Container wird neu gestartet, wenn Probe fehlschlägt
- Wird regelmäßig ausgeführt

Probes - livenessProbe

```
(...)  
containers:  
- name: liveness  
  image: registry.k8s.io/busybox  
  args:  
  - /bin/sh  
  - -c  
  - touch /tmp/healthy; sleep 30; rm -f /tmp/healthy; sleep 600  
  livenessProbe:  
    exec:  
      command:  
      - cat  
      - /tmp/healthy  
    initialDelaySeconds: 5  
    periodSeconds: 5
```

Probes - `readinessProbe`

- Container wird aus dem Service genommen, wenn Probe fehlschlägt
- Wird regelmäßig ausgeführt

Probes - readinessProbe

```
(...)  
  readinessProbe:  
    exec:  
      command:  
      - cat  
      - /tmp/healthy  
    initialDelaySeconds: 5  
    periodSeconds: 5
```

Probes - `startupProbe`

- Wird nur einmal ausgeführt
- Wird vor `livenessProbe` und `readinessProbe` ausgeführt
- Für Pods mit langen Startzeiten

Probes - startupProbe

```
(...)  
  startupProbe:  
    exec:  
      command:  
      - cat  
      - /tmp/healthy  
    initialDelaySeconds: 10
```

Probes - Checks

- `HTTP`: Prüft, ob ein bestimmter Endpunkt erreichbar ist
- `TCP`: Prüft, ob ein bestimmter Port erreichbar ist
- `Exec`: Führt ein bestimmtes Kommando aus

Probes - Optionen

- `initialDelaySeconds`: Wartezeit vor dem ersten Test
- `periodSeconds`: Intervall zwischen den Tests
- `timeoutSeconds`: Zeit, die ein Test maximal dauern darf
- `successThreshold`: Anzahl der erfolgreichen Tests
- `failureThreshold`: Anzahl der fehlgeschlagenen Tests

Zur Kapitelübersicht

- Vorheriges Kapitel: [Docker Troubleshooting](#)
- Nächstes Kapitel: [Kubernetes Admin Troubleshooting](#)