

## 4.4 데몬셋을 사용해 각 노드에서 정확히 한 개의 파드 실행하기

🕒 생성일	@2021년 3월 27일 오후 2:05
🏷️ 태그	

클러스터의 모든 노드에, 노드당 하나의 파드만 실행되길 원하는 경우

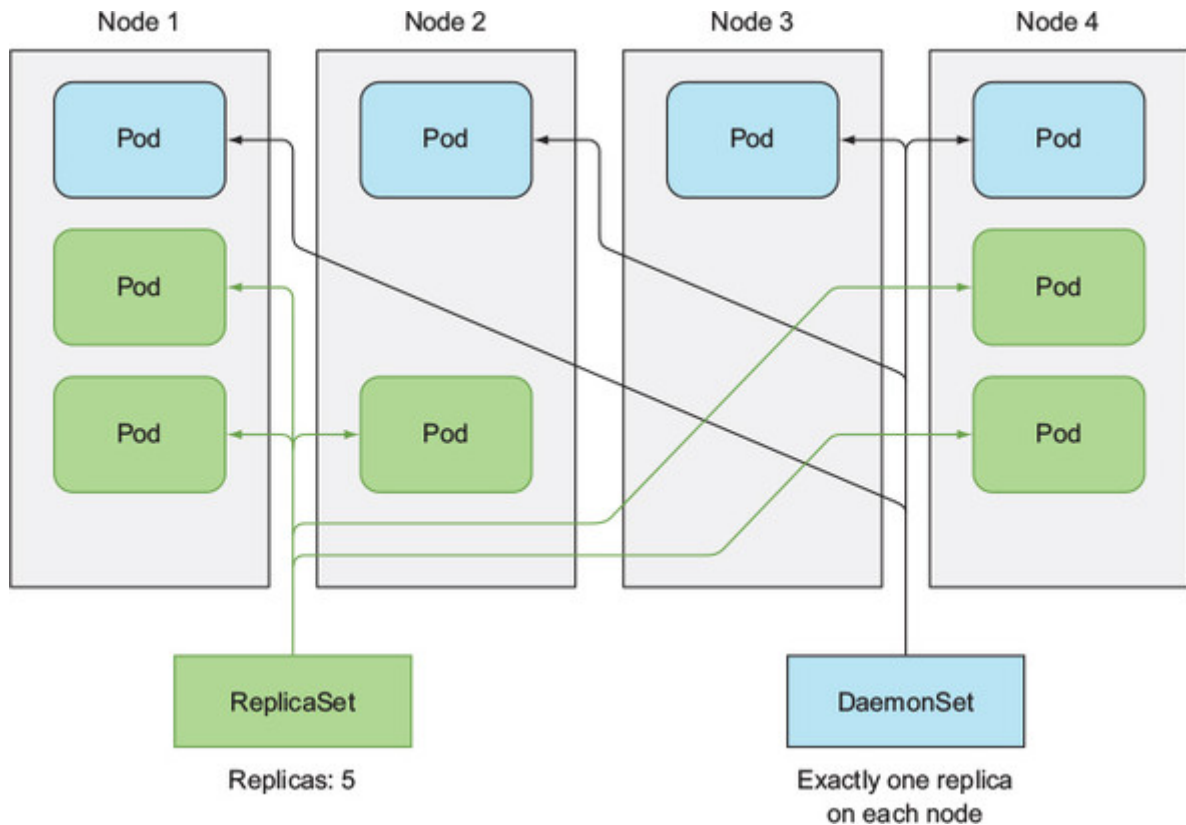
- 시스템 수준의 작업을 수행하는 인프라 관련 파드
- 노드에서 로그 수집기와 리소스 모니터를 실행하려는 경우
- 쿠버네티스의 kube-proxy 프로세스 : 서비스를 작동시키기 위해 모든 노드에서 실행돼야 한다.

쿠버네티스 노드에서 `systemd` 으로 시스템 프로세스 실행할 수도 있지만, 쿠버네티스가 제공하는 기능을 최대한 활용할 수 없다.

### 1. 데몬셋으로 모든 노드에 파드 실행하기

모든 클러스터 노드마다 파드를 하나만 실행하려면 `DaemonSet` 오브젝트를 생성해야함

- 데몬셋에 의해 생성되는 파드는 타깃 노드가 이미 지정돼 있음
- 데몬셋에 의해 생성되는 파드는 쿠버네티스 스케줄러를 건너뛴다.

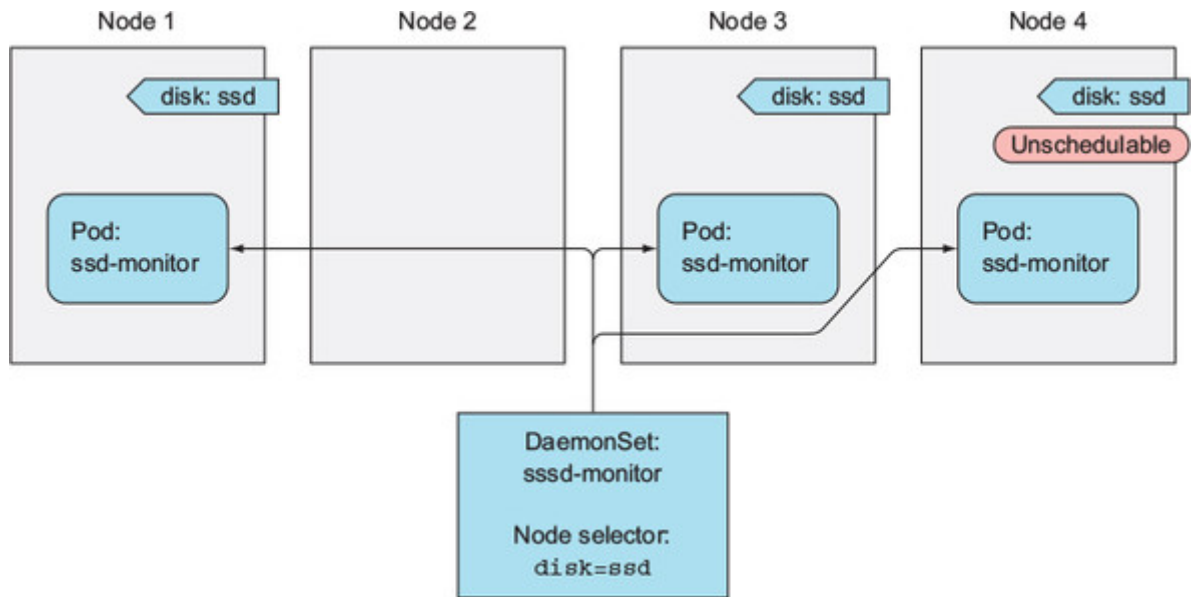


- 데몬셋은 노드 수 만큼 파드를 만들고, 각 노드 1-4에 배포
- 데몬셋은 원하는 복제본 수라는 개념이 없음
  - RS : 클러스터에 원하는 수의 파드 복제본이 존재하는지 확인
- 각 노드에서 파드셀렉터와 일치하는 파드 1개가 잘 실행중인지만 확인
- 노드 다운시, 데몬셋은 다른 곳에서 파드를 생성하지 않음
  - 클러스터에 새 노드 추가될 시, 데몬셋은 즉시 새 파드 인스턴스를 새 노드에 배포

## 2. 데몬셋을 사용해 특정 노드에서만 파드를 실행하기

데몬셋 정의는 파드 템플릿에서 node-selector 속성을 지정하면된다.

### 예제로 데몬셋 설명



`ssd-monitor` 라는 데몬 : ssd를 갖는 모든 노드에서 실행되어야 한다고 가정  
 → ssd 가진 노드(`disk=ssd` 레이블 추가)에서 이 데몬을 실행하는 데몬셋을 만들

## 데몬셋 YAML 정의 생성

5초마다 "SSD OK" 출력하는 ssd-monitor 프로세스를 실행하는 데몬셋 생성

vi로 `ssd-monitor-daemonset.yaml` 생성

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: ssd-monitor
spec:
  selector:
    matchLabels:
      app: ssd-monitor
  template:
    metadata:
      labels:
        app: ssd-monitor
    spec:
      nodeSelector: #파드템플릿은 disk=ssd 레이블이 있는 노드를 선택하는 노드 선택터를 가짐
        disk: ssd
      containers:
        - name: main
          image: luksa/ssd-monitor
```

- 이제 `disk=ssd` 레이블이 있는 각 노드에 이 파드의 인스턴스가 생성될것임

## 데몬셋 생성

```
kubectl create -f ssd-monitor-daemonset.yaml
kubectl get ds
kubectl get po
```

근데 0으로 되어있당....

머선일이고..

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice
❌ kubectl create -f ssd-monitor-daemonset.yaml
Error from server (AlreadyExists): error when creating "ssd-monitor-daemonset.yaml": daemonsets.apps "ssd-monitor" already exists
❌ kubectl get ds
No resources found in default namespace.
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice kubectl get ds
NAME          DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
ssd-monitor    0         0         0       0             0          disk=ssd       5m51s
```

- disk=ssd 레이블을 노드에 추가 안해서 그렇다 ㅎㄷ..
- 레이블 추가 하면 데몬셋이 노드의 레이블이 변경됐음을 감지하고 해당 레이블이 있는 모든 노드에 배포

## 필요한 레이블을 노드에 추가하기

```
kubectl get node
```

저는 GKE 노드 클러스터를 사용하기에.. 이걸로 진행해보겠어음

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice kubectl get node
NAME                                STATUS    ROLES    AGE    VERSION
gke-kubia-default-pool-378e3d66-f4rr Ready     <none>    15d    v1.18.12-gke.1210
gke-kubia-default-pool-378e3d66-fgr7 Ready     <none>    15d    v1.18.12-gke.1210
gke-kubia-default-pool-378e3d66-q5gx Ready     <none>    15d    v1.18.12-gke.1210
```

```
kubectl label node gke-kubia-default-pool-378e3d66-f4rr disk=ssd
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice kubectl label node gke-kubia-default-pool-378e3d66-f4rr disk=ssd
node/gke-kubia-default-pool-378e3d66-f4rr labeled
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice
```

이제 데몬셋을 체크해보자

```
kubectl get po
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
ssd-monitor-fkc64  1/1     Running   0          51s
```

아주 잘되고 있다. 다른 노드에도 레이블 추가시 데몬셋이 각 노드마다 파드 기동한당

## 노드에서 레이블 제거하기

노드에 ssd가 아닌 disk=hdd로 레이블을 잘못기재 했다면?

```
kubectl label node gke-kubia-default-pool-378e3d66-f4rr disk=hdd --overwrite
```

```
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl label node gke-kubia-default-pool-378e3d66-f4rr disk=hdd --overwrite
node/gke-kubia-default-pool-378e3d66-f4rr labeled
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
ssd-monitor-fkc64  1/1     Terminating   0          3m5s
jiseonsim@simjiseon-ui-MacBook-Air ~/Desktop/git/KubeStudy-practice$
```

Terminate 되고 있다.....!!!! ππ

ssd-monitor 데몬셋을 삭제해보자

```
kubectl delete ds ssd-monitor
```