

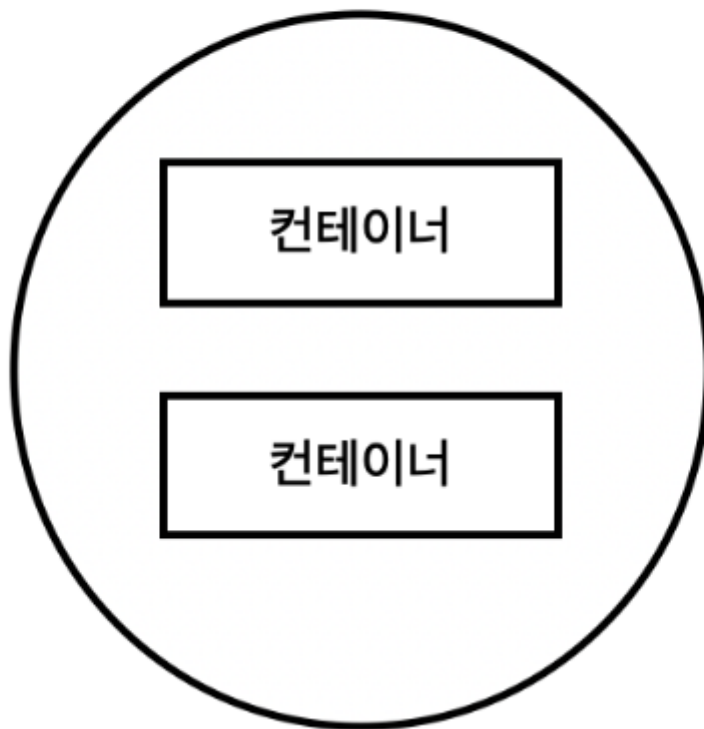
- Pod [📄](#) [📄](#) [📄](#) [📄](#)
- Deployment [📄](#) [📄](#)
- ReplicaSet [📄](#) [📄](#)
- StatefulSet [📄](#) [📄](#)
- DaemonSet [📄](#) [📄](#)
- HPA(Horizontal Pod Autoscaler) [📄](#) [📄](#)
- Job & CronJob [📄](#) [📄](#)

Pod

Pod

컨테이너 엔진을 사용하여 Pod를 실행할 수 있습니다. Pod는 컨테이너 엔진을 사용하여 실행되는 컨테이너의 집합입니다. Pod는 Kubernetes 클러스터에서 실행되는 기본 단위입니다.

Pod



Pod를 생성하기 위해 Yaml 파일을 사용합니다:

```
kubectl apply -f example-pod.yaml
pod/example-pod.yaml created
```

Pod가 생성된 후 Nginx 서비스를 실행합니다. Pod를 확인합니다:

```
kubectl get pods
NAME READY STATUS RESTARTS AGE
example-pod 1/1 Running 0 40s
```

Nginx 서비스는 80번 포트에서 실행되며, Pod는 80번 포트에서 트래픽을 수신합니다.

❏ ❏❏❏ ❏❏ ❏❏❏:

```
kubectrl delete -f example-pod.yaml
```

```
pod "example-pod" deleted
```

```
kubectrl get pods
```

```
No resources found in default namespace
```

Deployment 部署

部署过程涉及多个 Kubernetes 资源对象。主要对象包括：Pod (Pod), ReplicaSet (ReplicaSet), Service Account (Service Account)。

```
# Nginx 部署示例 YAML 文件内容
apiVersion: v1
kind: Pod
metadata:
  name: example-pod
spec:
  containers:
  - name: example-container
    image: nginx:latest
    ports:
    - containerPort: 80
      protocol: TCP
```

部署操作通常通过 YAML 文件定义，并包含 apiVersion, kind, metadata, spec 等关键字段。

- apiVersion** 指定 YAML 文件使用的 API 版本。
- kind** 指定资源类型，例如 Pod (kind: Pod)。
- metadata** 包含元数据，如 label, annotation, name 等。
- spec** 指定容器的配置，例如 - name: example-container, image: nginx:latest, 以及 containerPort: 80。

使用 `kubectl apply -f <yaml 文件>` 命令部署。

kubectl 是 Kubernetes 的命令行工具，用于管理集群资源。部署操作通常通过 `kubectl` 命令完成。

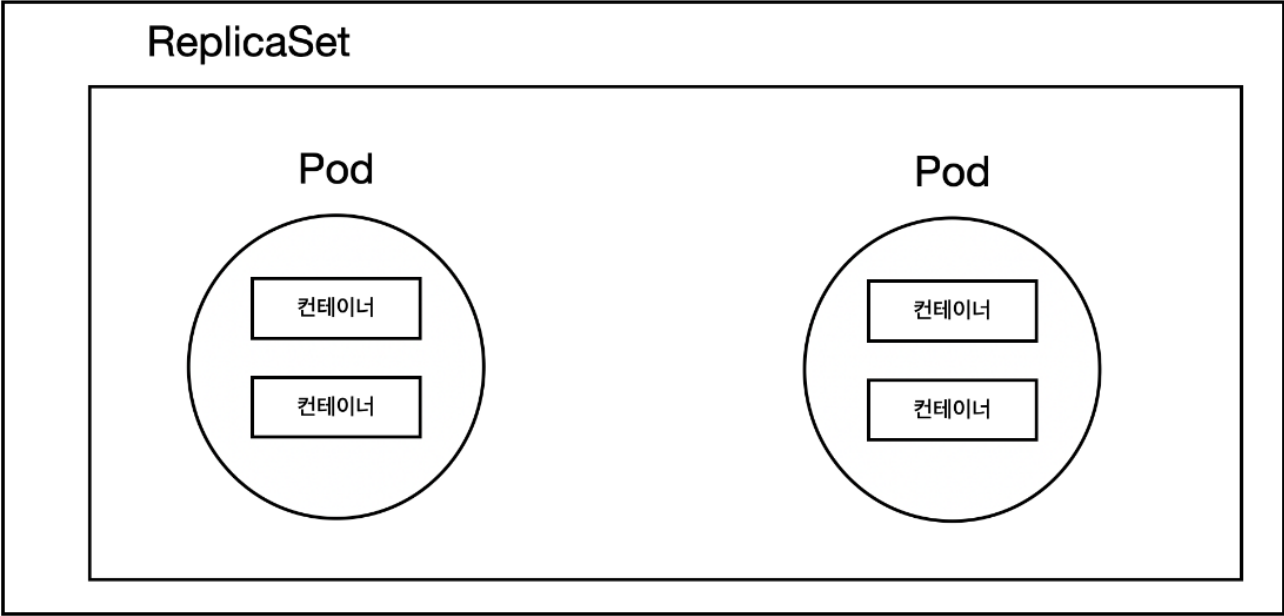
`kubectl` 命令用于部署资源。部署操作通常通过 `kubectl` 命令完成。

- `kubectl` 用于部署 Kubernetes 资源，如 Pod, ReplicaSet, Service 等。
- `kubectl` 用于查看集群状态，如 `kubectl get pods`。
- `kubectl` 用于创建资源，如 `kubectl create`。
- `kubectl` 用于删除资源，如 `kubectl delete`。

Deployment

部署操作通常通过 YAML 文件定义，并包含 apiVersion, kind, metadata, spec 等关键字段。

Deployment



Yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-nginx
  template:
    metadata:
      name: my-nginx-pod
    labels:
      app: my-nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.10
          ports:
            - containerPort: 80
```

Yaml kind 는 Deployment 이고, replicas 는 3 이고, selector 는 app: my-nginx 이고, template 는 pod 의 정보를 담고 있다.

3 - 0, 3 - 1, 2 - 2, 1 - 3, 0 - 3

kubectl describe deployment {deployment name} 显示部署的详细信息，包括副本数、更新策略等。

kubectl rollout history deployment {deployment name} 显示部署的历史记录。

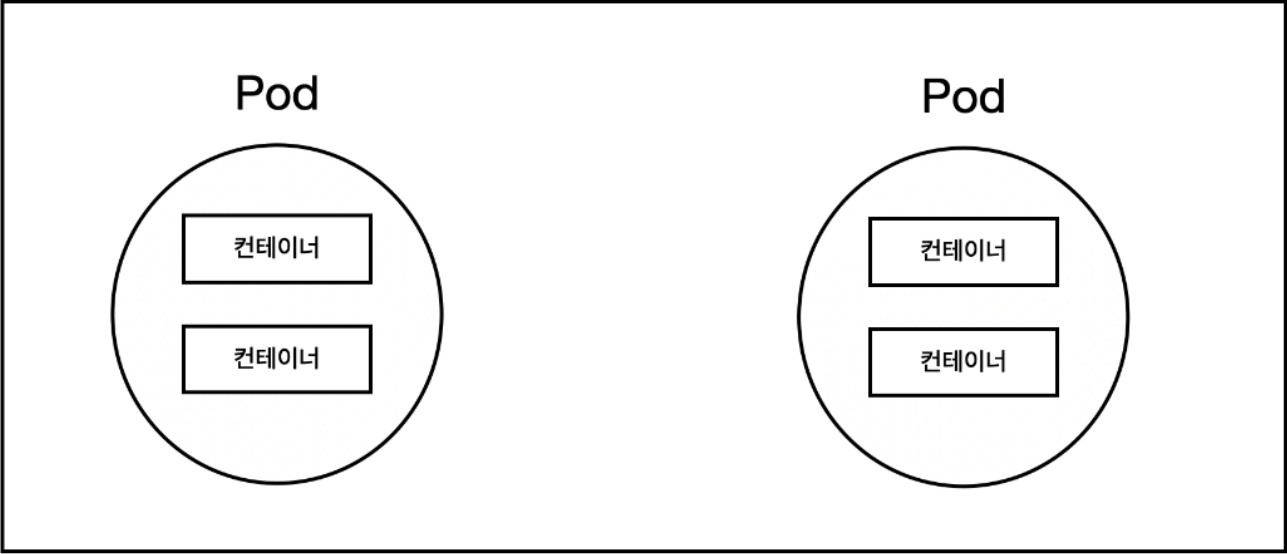
kubectl rollout undo deployment {deployment name} --to-revision={revision} 将部署回滚到指定的版本。

ReplicaSet

ReplicaSet

Pod는 Kubernetes의 기본 실행 단위입니다. ReplicaSet은 Pod의 수를 일정하게 유지하기 위해 Pod를 생성하고 관리합니다. Pod는 컨테이너를 실행하는 데 사용됩니다.

ReplicaSet



예:

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replicaset-nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-nginx-pods-label # Pod의 라벨을 지정합니다.
  template:
    # Pod의 템플릿을 지정합니다.
    metadata:
      name: my-nginx-pod
    labels:
      app: my-nginx-pods-label
    spec:
```

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

YAML 7 Replicas: 3 template ReplicaSet matchLabels 10 replicas: matchLabels

StatefulSet  

DaemonSet  

HPA(Horizontal Pod
Autoscaler) □□□ □□

Job & CronJob