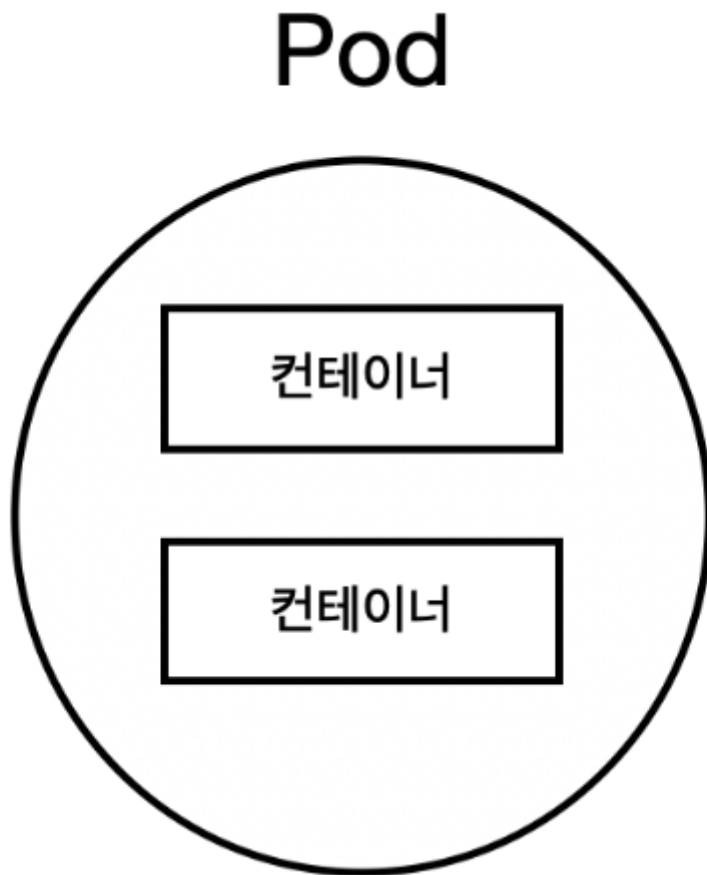


- Pod [📄](#) [📄](#) [📄](#) [📄](#) [📄](#)
- Deployment [📄](#) [📄](#)
- ReplicaSet [📄](#) [📄](#)
- StatefulSet [📄](#) [📄](#)
- DaemonSet [📄](#) [📄](#)
- HPA(Horizontal Pod Autoscaler) [📄](#) [📄](#)
- Job & CronJob [📄](#) [📄](#)

Pod

Pod

컨테이너를 실행하는 단위인 Pod(포드)는 Kubernetes의 기본 단위입니다. Pod는 하나 이상의 컨테이너를 포함하며, 이들은 공유 네트워크 스택과 storage를 사용합니다.



Pod를 생성하기 위해 Yaml 파일을 사용합니다:

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

Pod에 Nginx를 배포합니다. 확인:

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

Nginx Pod는 80번 포트에서 트래픽을 수신하며, 이는 서비스와 연결되어 있습니다.

❏ ❏❏❏ ❏❏ ❏❏❏:

```
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。核心对象包括：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`、`ReplicaSet`、`Deployment`。
- metadata** 包含元数据，如 `name`、`labels`、`annotations`。
- spec** 定义资源的具体配置，例如 `containers` 列表，包含 `name`、`image`、`ports` 等。

使用 `kubectl apply -f <yaml文件>` 命令部署资源。

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` 完成。

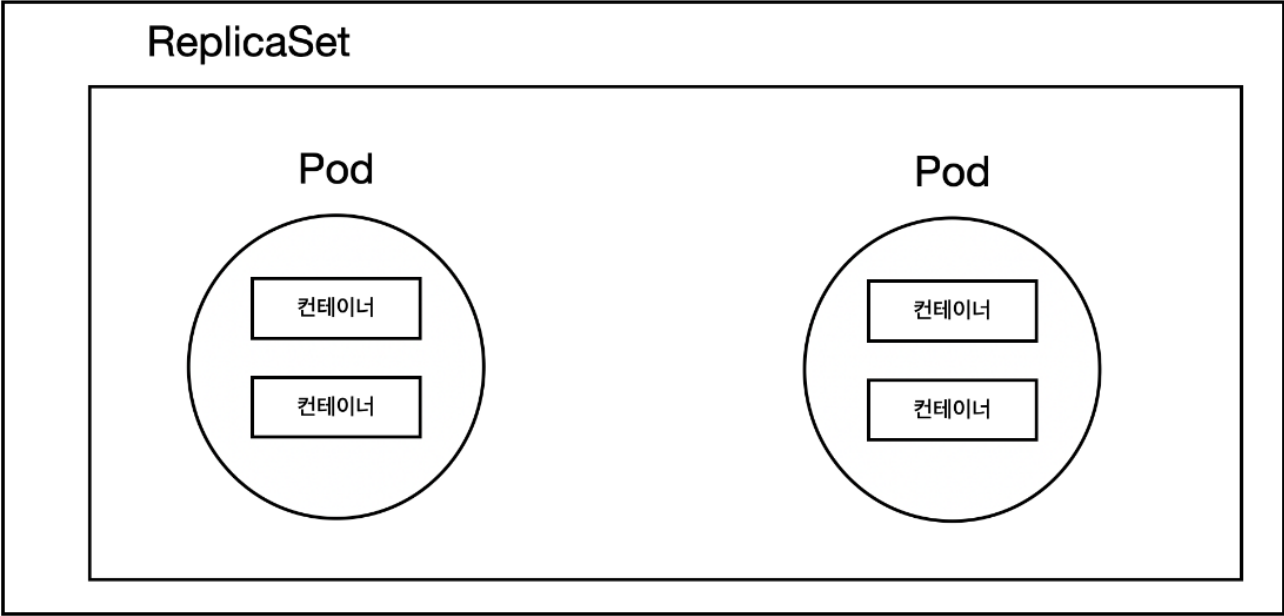
`kubectl` 命令用于部署资源。部署操作通常通过 `kubectl` 完成。

- `kubectl` 部署 Kubernetes 资源，如 `Pod`、`ReplicaSet`、`Deployment`。
- `kubectl` 部署资源时，需要指定资源名称、命名空间、镜像、端口等。
- `kubectl` 部署资源时，需要指定资源名称、命名空间、镜像、端口等。
- `kubectl` 部署资源时，需要指定资源名称、命名空间、镜像、端口等。

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。

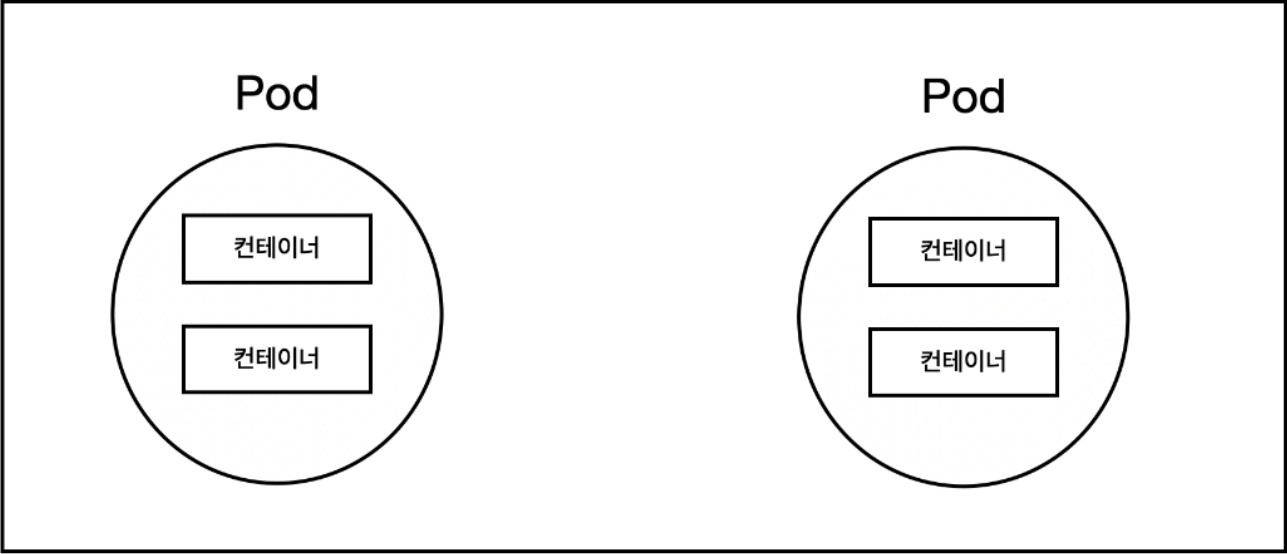
kubectl rollout undo deployment {deployment name} --to-revision={revision} 将部署回滚到指定的版本。

ReplicaSet

ReplicaSet

Pod는 클러스터에서 실행되는 애플리케이션의 단위입니다. ReplicaSet은 Pod의 수를 지정하고, Pod가 실행 중일 때 Pod가 실행 중이 아닌 Pod를 생성하여 Pod의 수를 지정합니다. Pod는 클러스터에서 실행되는 애플리케이션의 단위입니다.

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 Replicas matchLabels replicas: matchLabels

StatefulSet  

DaemonSet  

HPA(Horizontal Pod
Autoscaler) □□□ □□

Job & CronJob