Kubernetes labels provide a way to attach custom, identifying information to your objects. Selectors can then be used to filter objects using label data as criteria. Annotations, on the other hand, offer a more freeform way to attach useful but non-identifying metadata. In this lesson, we will discuss labels, selectors, and annotations. We will also demonstrate how to use them in a cluster.

Relevant Documentation

- https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/
- https://kubernetes.io/docs/concepts/overview/working-with-objects/annotations/

Lesson Reference

Here is a pod with some labels.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-production-label-pod
  labels:
    app: my-app
    environment: production
spec:
  containers:
  - name: nginx
    image: nginx
```

You can view existing labels with `kubectl describe`.

```
kubectl describe pod my-production-label-pod
```

Here is another pod with different labels.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-development-label-pod
  labels:
    app: my-app
    environment: development
spec:
  containers:
  - name: nginx
    image: nginx
```

You can use various selectors to select different subsets of objects.

```
kubectl get pods -l app=my-app

kubectl get pods -l environment=production

kubectl get pods -l environment=development

kubectl get pods -l environment!=production

kubectl get pods -l 'environment in (development,production)'

kubectl get pods -l app=my-app,environment=production
```

Here is a simple pod with some annotations.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-annotation-pod
  annotations:
    owner: terry@linuxacademy.com
    git-commit: bdab0c6
spec:
  containers:
  - name: nginx
    image: nginx
```

Like labels, existing annotations can also be viewed using `kubectl describe`.

```
kubectl describe pod my-annotation-pod
```