Kubernetes Labels and Annotation

Agenda:

1. Labels
2. Pod Create with Labels
3. Adding Labels to a Running Pod
4. Select Kubernetes objects using Label selector
5. Modify/Delete the Label
6. Annotation
7. Pod create with Annotation
8. Modify/Delete the annotation

Labels

- They are are the metadata which contain unique information of the Kubernetes objects
- key-value pair attached to the objects like pod
- We can attach the label at the time of creation or we can update it at runtime
- Kubernetes internally maintains the mappings of labels to corresponding objects using optimized data structures to make these queries faster.
- With labels, you can also run your specific pods on selected nodes.

Organize Pods by team/project/organization

- You can organize pods so that each team will see only their pods
- Specific to certain projects

```
metadata:
  labels:
     environment: development
     team: infra
     project: k8s-infra
```

- ## *Run a Pod on Specific Nodes*

- This is a very basic requirement when you want to run your pod on a specific node because of its hardware/specific configuration etc.

- We can do that using `node-selector`

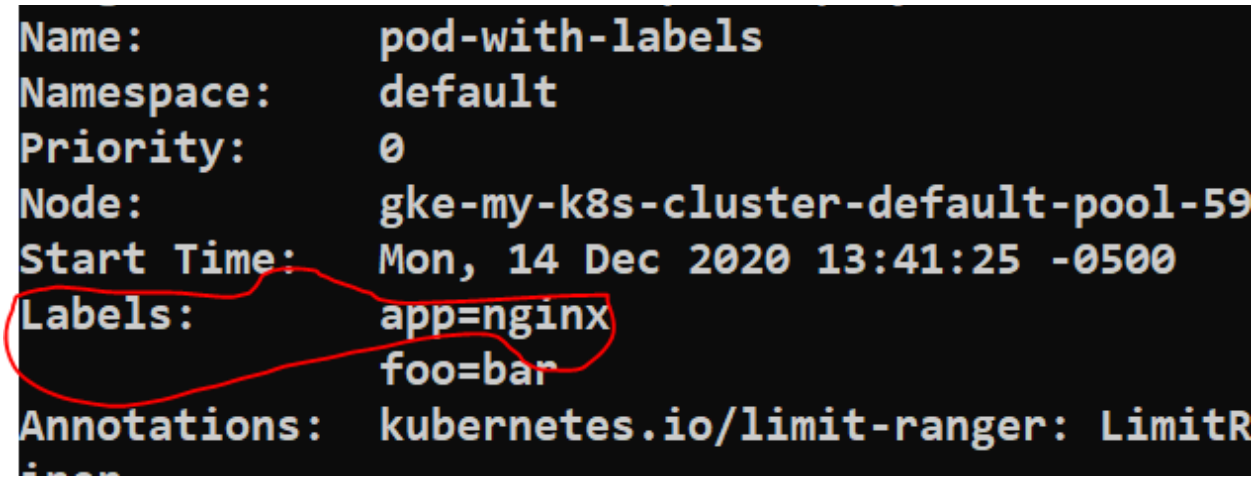
```
nodeSelector:
  region: east-us
  disktype: ssd
```

Create a Pod with Labels

```
kind: Pod
apiVersion: v1
metadata:
  name: pod-with-labels
  labels:
    app: nginx
    foo: bar
spec:
  containers:
  - name: first-container
    image: nginx
```

- Create a pod

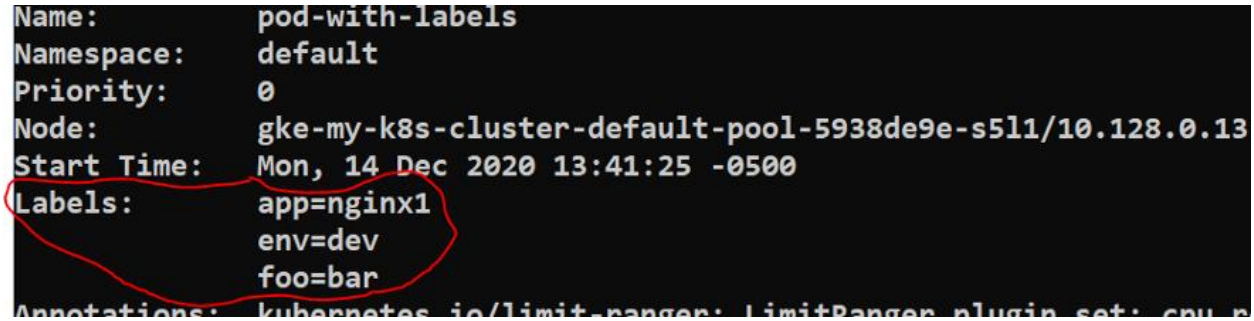
```
kubectl create -f pod-with-labels.yaml
kubectl describe pod pod-with-labels
```



Adding Labels to a Running Pod

You can update the existing label or add the new one using below command

```
kubectl label pod pod-with-labels app=nginx1 --overwrite
kubectl label pod pod-with-labels env=dev
kubectl label pod pod-with-labels team=dev org=abc
```



How to remove a label

```
kubectl label pod pod-with-labels env-
```

Selecting Kubernetes Objects Using Label Selectors

```
kubectl get pods -l app=nginx1
kubectl get pods -l app=nginx1,foo=bar
```

Annotation

Annotations are also key-value pairs that can be used to store the unstructured information pertaining to the Kubernetes objects.

It can be used to add timestamps, commit SHA, issue tracker links, or names/information about users who are responsible for specific objects in an organization

It can also be used to add information about client libraries or tools.

```
kind: Pod
apiVersion: v1
metadata:
  name: pod-with-annotation
  annotation:
    commit-message: testign for nginx
    owner: Nidhi
spec:
  containers:
  - name: first-container
    image: nginx
```

Create a Pod

```
kubectl create -f pod-with-annotation.yaml

kubectl describe pod pod-with-annotation
```

How to add annotation using command line

```
kubectl annotate pod pod-with-annotation jiralink=DED-232
kubectl annotate --overwrite pod <pod_name> <annotation_key>=<annotation_label>
```

Delete the annotation

```
kubectl annotate pod pod-with-annotation jiralink-
```