Pods

» Labels, Selectors, and More

Overview

Get a list of all pods in the default namespace.

```
kubectl get pods
```

Let's create some pods with labels.

```
kubectl run www-dev --image nginx --restart Never -l app=my-app,tier=frontend,environment=dev --port 80
kubectl run www-prod --image nginx --restart Never -l app=my-app,tier=frontend,environment=production --port 80
kubectl run db-dev --image mariadb --restart Never -l app=my-app,tier=backend,environment=dev --env MYSQL_ROOT_PASSWORD=secret --port
 3306
```

Verify that the pods were successfully created.

```
kubectl get pods
kubectl get pods -o wide
kubectl get pods --show-labels
```

You can also add and remove labels with the `kubectl label` command.

```
kubectl label pod www-dev version=1
```

Verify the `version=1` label was set.

```
kubectl get pods -l version=1 -L version
```

Remove the label.

```
kubectl label pods www-dev version-
```

Verify the `version=1` label was removed.

```
kubectl get pods -l version=1 -L version
```

Selectors

List the pods, along with the labels that we just set.

```
kubectl get pods --show-labels
kubectl get pods -L app,environment,tier
```

Regular Selectors

Use the `-l` or `--selector` flags to specify a selector to filter the results with `kubectl get .`

```
kubectl get pods -L app,environment,tier -l tier=frontend
```

List out the pods in the `frontend` tier of the `dev` environment.

```
kubectl get pods -L app,environment,tier -l tier=frontend,environment=dev
```

List out the pods that are not part of the `dev` environment.

```
kubectl get pods -L app,environment,tier -l environment!=dev
```

Set-based Selectors

Use set-based selectors for complex filtering. The following resources support set-based selectors: Jobs, Deployments, ReplicaSets, and DaemonSets.

```
kubectl get pods -l "tier notin (backend,cache),environment in (dev)"
```

Viewing Log Files

View the logs for the MariaDB pod.

```
kubectl logs db-dev
```

```
kubectl logs db-dev -f
```

Describe

View the description of the first pod.

```
kubectl describe pod www-dev
```

View the manifest file associated with the first pod.

```
kubectl get pod www-dev -o json
kubectl get pod www-dev -o yaml
```

Fetch a specific value using JSONPath. You can alternatively use jq (https://github.com/stedolan/jq).

Check out https://github.com/stedolan/jq/wiki/For-JSONPath-users (https://github.com/stedolan/jq/wiki/For-JSONPath-users).

List the pods with the label `app=my-app` .

```
kubectl get pods -l app=my-app -o wide -L app,environment,tier
```

Use JSONPath to print the name of the first pod in the list.

```
kubectl get pods -l app=my-app --output jsonpath={.items[0]..metadata.name}; echo
```

Use JSONPath to print the cluster IP address for the third pod in the list.

```
kubectl get pods -l app=my-app --output jsonpath={.items[2]..status.podIP}; echo
```

Clean Up

Delete the pods using a selector.

```
kubectl delete pod -l app=my-app
```

View any remaining resources.

```
kubectl get all
```