FoxuTech Follow
Jun 7 · 3 min read

How to create and change the Kubernetes yaml using dry-run and edit

Sometime we may find hard to create a Kubernetes yaml or there will be some indentation issue or we are lazy to create a whole file or we may experience it is time consuming work. What if we say, you no need to type complete file, instead you can run *kubectl* command, it can create the files for you...!!! Yes, you can use *kubectl run* and *kubectl create* with the *–dry-run=client* flag for creating pods and deployments it should give the skeleton file. Let's check with some examples, so it will be easy to understand.

Create a basic NGINX Pod

Here is the command we use to run the sample Nginx pod with *kubectl*.

```
# kubectl run nginx --image=nginx
```

This will create a nginx pod on the cluster.

```
# kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          34s
```

As this is quite easy to run the command, but what if need to document the changes in IAC or any place for future purpose? We should keep as *yaml*, this should be tricky for the newbies or beginners.

```
# kubectl run nginx --image=nginx --dry-run=client -o yaml
```

This will output a yaml file you can then apply/create or update as needed.

```
metadata:
  creationTimestamp: null
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

Create a deployment

```
# kubectl create deployment --image=nginx nginx
```

This creates a deployment without needing to go fetch a deployment yaml.

```
# kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   1/1     1            1           34s
```

As like before for pods, we can try below command for deployment.

```
# kubectl create deployment --image=nginx nginx --dry-run=client -o
yaml
```

This will give you a template for creating a deployment.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: nginx
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
```

As we see now, the files are just shown on the command line, we can redirect that to file, as below.

```
# kubectl create deployment --image=nginx nginx --dry-run=client -o
yaml > nginx-deployment.yaml
```

So far, we have seen, how to create new one, but another challagen we may have been, we may need to make the change on the fly or existing running pods or deployments. For that *kubectl* contains edit, which you can use to make quick updates to your yaml. For example, after applying the previously created file:

```
# kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx created
```

You can directly edit the deployment yaml as follows:

```
# kubectl edit deployment/nginx
```

Please be informed, any changes you make using *kubectl edit* won't be applied directly to the yaml file, it's just making changes on the running resource. If you wish record the changes, you should make a entry on yaml directly and use *kubectl apply*. Check the difference between kubectl apply and kubectl create on https://foxutech.com/kubectl-apply-vs-kubectl-create/

if you like this post **BuyMeACoffee**:
https://www.buymeacoffee.com/foxutech