

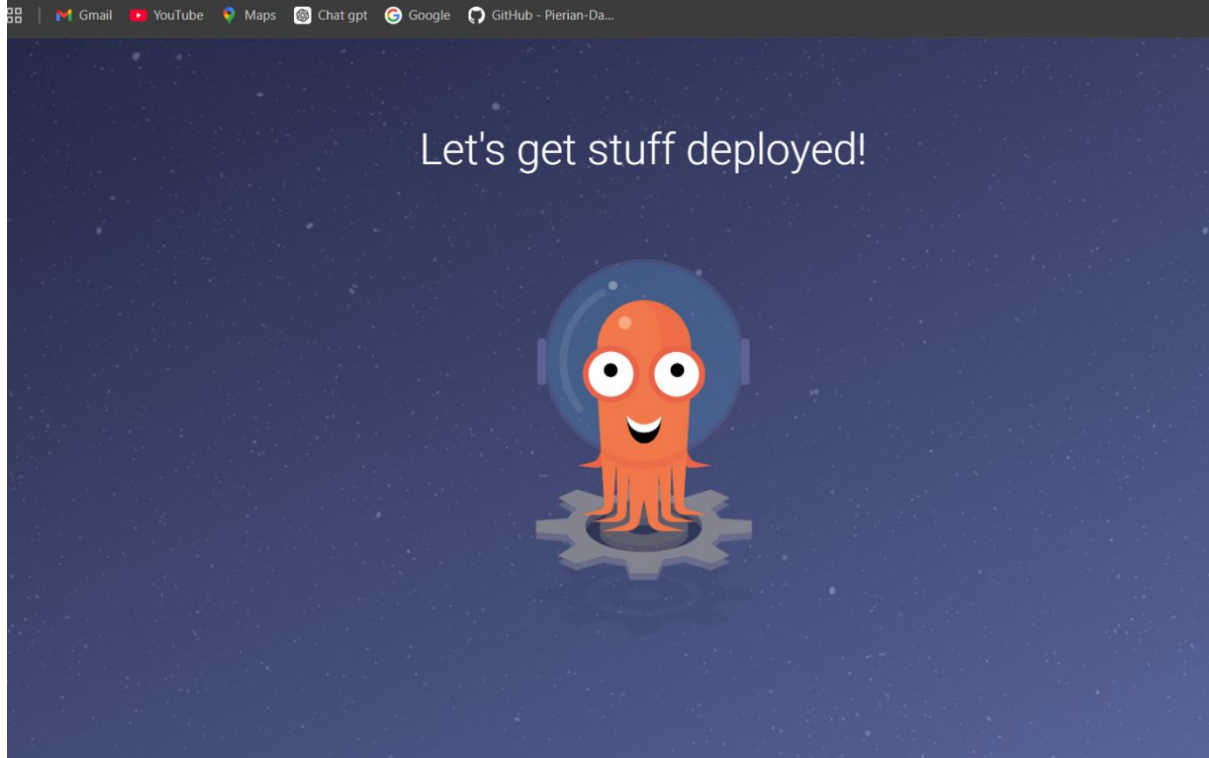
```

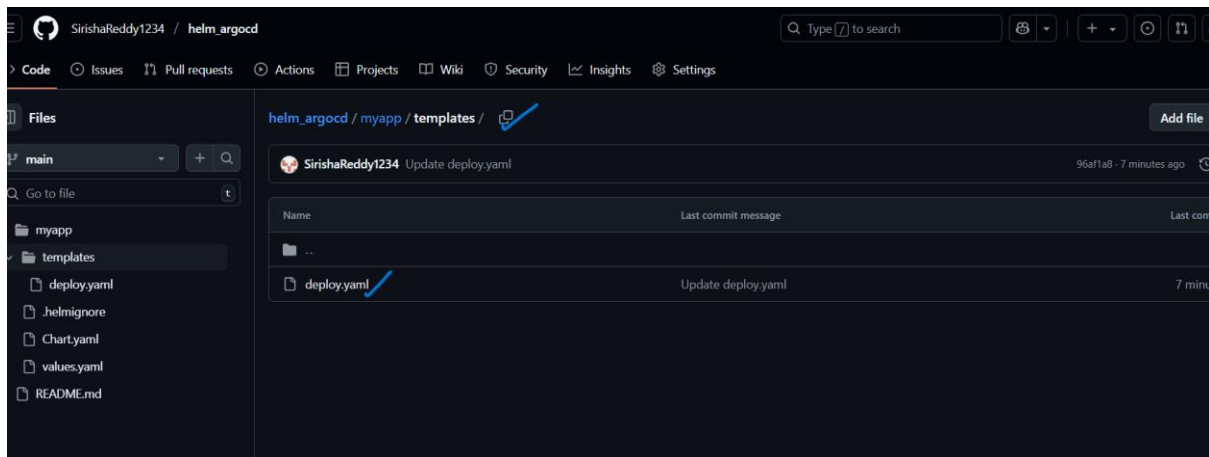
Last login: wed Apr 16 13:11:28 2025 from 106.216.201.193
ubuntu@k8s-master:~$ ls
deploy.yaml  deploy1.yaml  get_helm.sh  pv.yaml  pvc.yaml  pvcl.yaml  sc.yaml
ubuntu@k8s-master:~$ rm -rf *
ubuntu@k8s-master:~$ ls
ubuntu@k8s-master:~$ helm repo add argo https://argoproj.github.io/argo-helm
"argo" already exists with the same configuration, skipping
ubuntu@k8s-master:~$ ls
ubuntu@k8s-master:~$ kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
my-argo-cd-argocd-application-controller-0  1/1      Running   1 (33m ago)  21h
my-argo-cd-argocd-applicationset-controller-869cf6d798-7pkc4  1/1      Running   1 (33m ago)  21h
my-argo-cd-argocd-dex-server-5f8dd87dfd-gwzkq  1/1      Running   1 (33m ago)  21h
my-argo-cd-argocd-notifications-controller-688bf6796b-1k7xl  1/1      Running   1 (33m ago)  21h
my-argo-cd-argocd-redis-5566777d97-2bvd1  1/1      Running   1 (33m ago)  21h
my-argo-cd-argocd-repo-server-84f9b986cd-jmqmd  1/1      Running   1 (33m ago)  21h
my-argo-cd-argocd-server-598f9c45b4-n2tk5  1/1      Running   1 (33m ago)  21h
nginx-deployment1-75f9c5858-rr27s  1/1      Running   2 (33m ago)  42h
nginx-deployment1-75f9c5858-vvff7  1/1      Running   2 (33m ago)  41h
ubuntu@k8s-master:~$ kubectl get svc
NAME                                TYPE               CLUSTER-IP    EXTERNAL-IP    PORT(S)                                AGE
kubernetes                         ClusterIP          10.96.0.1      <none>          443/TCP                                7d20h
my-argo-cd-argocd-applicationset-controller  ClusterIP          10.103.67.58   <none>          7000/TCP                                21h
my-argo-cd-argocd-dex-server            ClusterIP          10.107.171.114 <none>          5556/TCP, 5557/TCP                     21h
my-argo-cd-argocd-redis                 ClusterIP          10.104.7.119   <none>          6379/TCP                                21h
my-argo-cd-argocd-repo-server           ClusterIP          10.96.95.145   <none>          8081/TCP                                21h
my-argo-cd-argocd-server                 NodePort           10.110.18.224   <none>          80:31375/TCP, 443:32095/TCP            21h
my-service                             NodePort           10.98.11.115    <none>          80:30007/TCP                            7d19h
ubuntu@k8s-master:~$ kubectl get secrets
NAME                                TYPE      DATA   AGE
argocd-initial-admin-secret         Opaque    1       21h
argocd-notifications-secret         Opaque    0       21h
argocd-redis                        Opaque    1       21h
argocd-secret                       Opaque    5       21h
sh.helm.release.v1.my-argo-cd.v1    helm.sh/release.v1  1       21h
ubuntu@k8s-master:~$ kubectl edit secrets argocd-initial-admin-secret
Edit cancelled, no changes made.
ubuntu@k8s-master:~$ echo -n "d1JhaEQ0YVNhMk9JcU1PdQ==" | base64
IGRsSmhhRVEwWVZOSElrOUpjVTFQZFE9PQ==
ubuntu@k8s-master:~$ echo -n "d1JhaEQ0YVNhMk9JcU1PdQ==" | base64
IGRsSmhhRVEwWVZOSElrOUpjVTFQZFE9PQ==
ubuntu@k8s-master:~$ kubectl edit secrets argocd-initial-admin-secret
Edit cancelled, no changes made.
ubuntu@k8s-master:~$

```

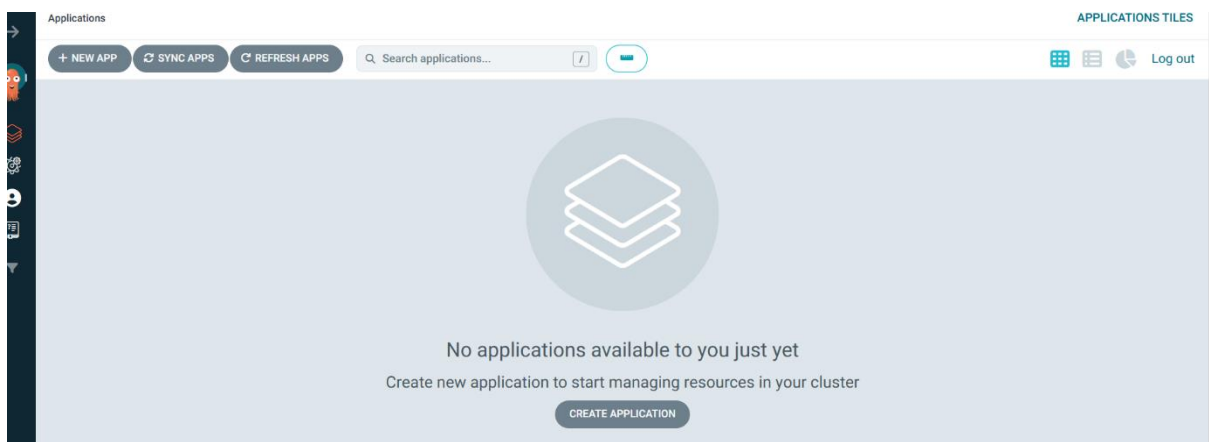
After installing Argo CD in my Kubernetes cluster, I modified the argocd-server service type from ClusterIP to NodePort to enable external access. To log in to the Argo CD application, I used the admin username and password, which was stored in a secret as a base64-encoded string; I decoded the password to obtain the actual value.

Not secure https://13.233.110.252:31375/login?return_url=https%3A%2F%2F13.233.110.252%3A31375%2Fapplications

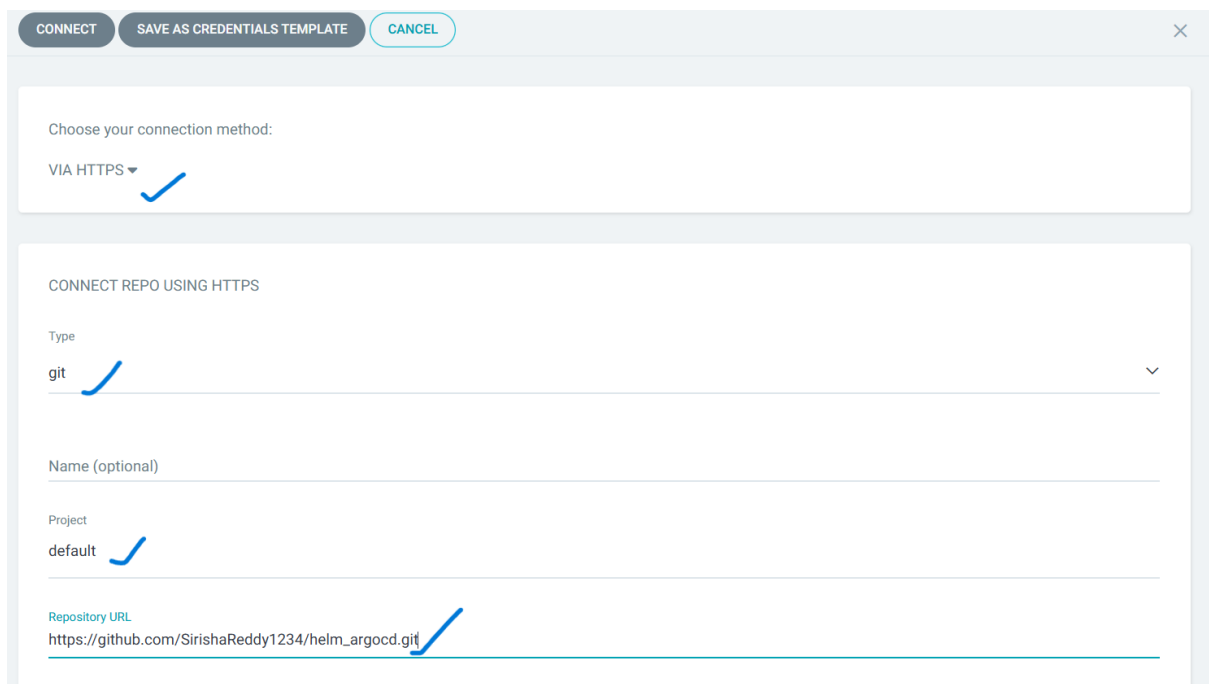




I created a Helm chart to package my Kubernetes configuration files and pushed it to my GitHub repository. Argo CD is connected to this repository and automatically deploys the YAML files defined in the Helm chart.



I login into the argo CD.



I'm registering the GitHub repository with Argo CD by providing the repository URL.

The screenshot shows the 'CREATE' form for a new Argo CD application. The form is divided into three main sections: GENERAL, SOURCE, and DESTINATION. In the GENERAL section, 'Application Name' is set to 'myapp', 'Project Name' is 'default', and 'SYNC POLICY' is 'Manual'. There are checkboxes for 'REPLACE' and 'RETRY'. In the SOURCE section, 'Repository URL' is 'https://github.com/SirishaReddy1234/helm_argocd.git', 'Revision' is 'HEAD', and 'Path' is 'myapp/templated'. A small preview of a YAML file is visible on the right. The DESTINATION section is currently empty.

CREATE CANCEL

EDIT AS YAML

GENERAL

Application Name
myapp ✓

Project Name
default ✓

SYNC POLICY
Manual ✓

☐ REPLACE ⚠️

☐ RETRY

SOURCE

Repository URL
https://github.com/SirishaReddy1234/helm_argocd.git ✓

Revision
HEAD

Path
myapp/templated ✓

DESTINATION

After creating the repository, I'm creating an application in Argo CD by providing general, source, and destination information.

The screenshot shows the 'APPLICATION DETAILS TREE' for the 'myapp' application. The top navigation bar includes 'DETAILS', 'DIFF', 'SYNC', 'SYNC STATUS', 'HISTORY AND ROLLBACK', 'DELETE', and 'REFRESH'. The 'APP HEALTH' is 'Healthy'. The 'SYNC STATUS' is 'Synced to HEAD (7115102)'. The 'LAST SYNC' is 'Sync OK to 7115102'. Below this, a diagram shows the application's structure: 'myapp' (application) is connected to 'nginx-deployment' (deployment), which is connected to 'nginx-deployment-d556bf558' (revision), which is then connected to three 'pod' instances. Each pod is labeled 'nginx-deployment-d556bf558...' and has a status of 'running'.

Applications / myapp

APPLICATION DETAILS TREE

DETAILS DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH
Healthy

SYNC STATUS
Synced to HEAD (7115102) ✓

LAST SYNC
Sync OK to 7115102

myapp

nginx-deployment

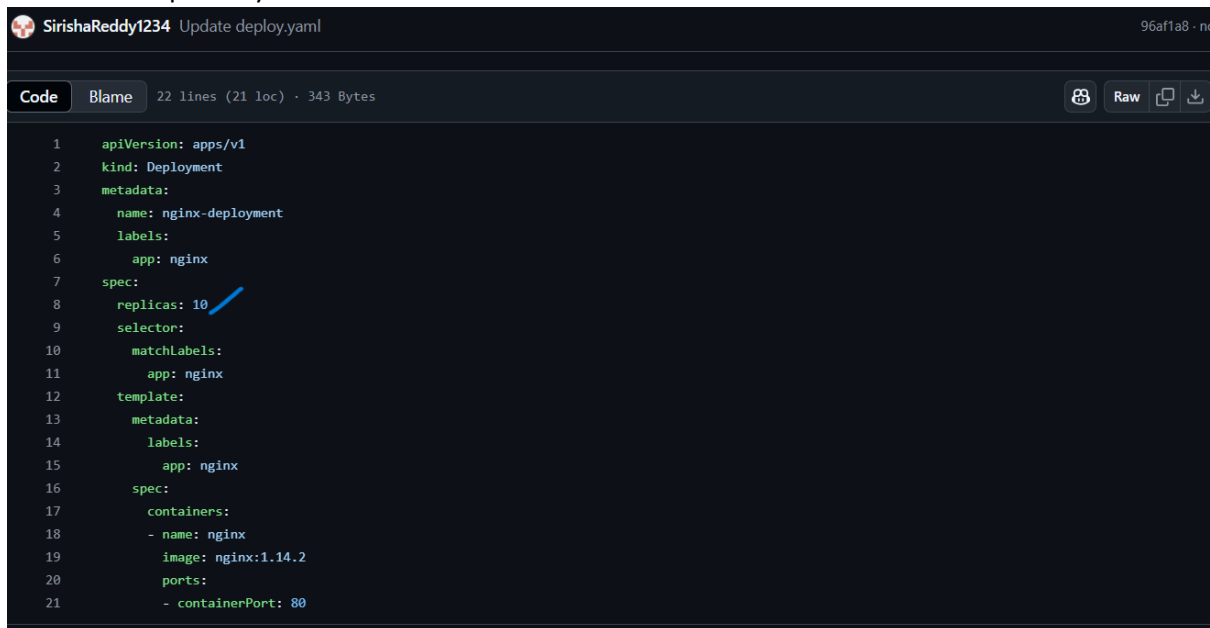
nginx-deployment-d556bf558

nginx-deployment-d556bf558...

nginx-deployment-d556bf558...

nginx-deployment-d556bf558...

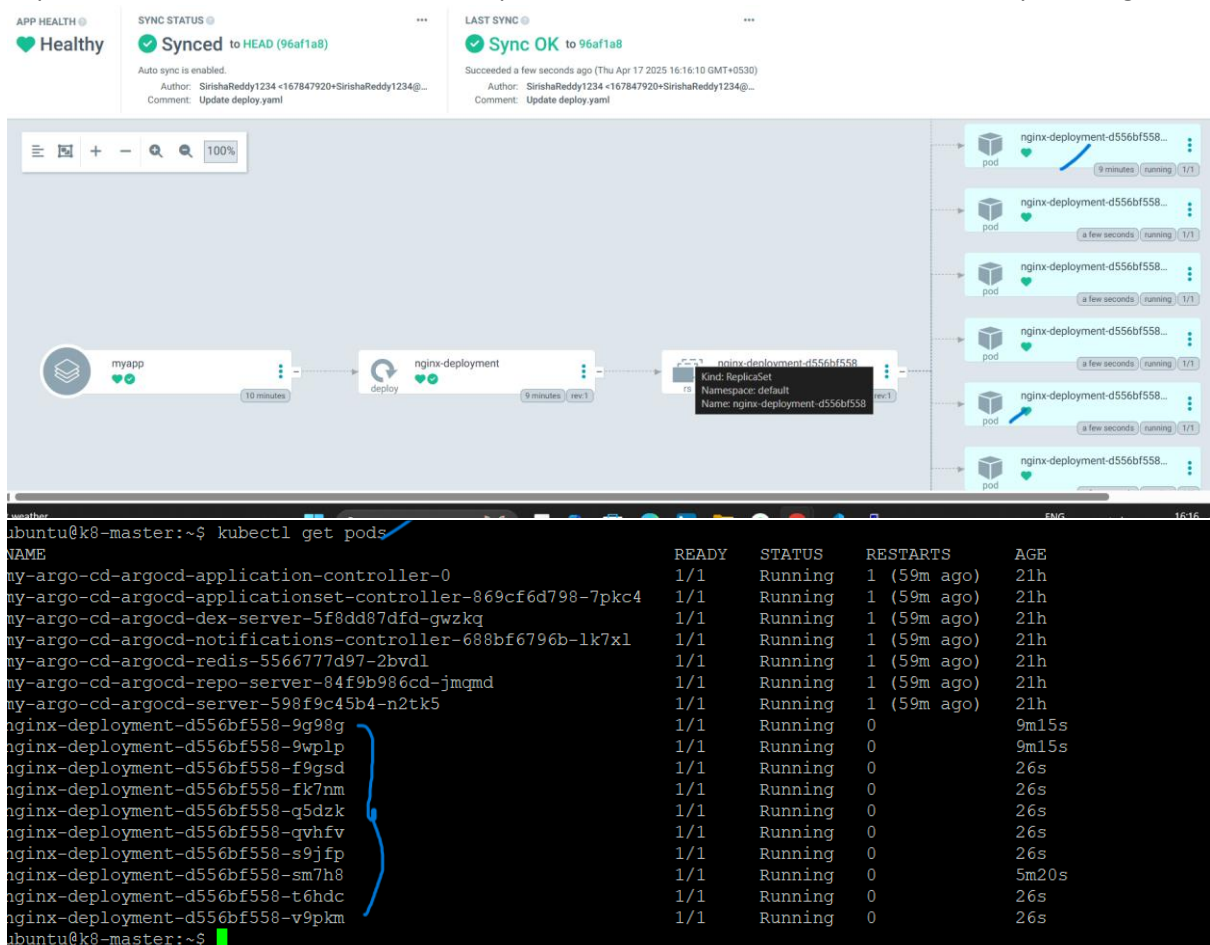
Now that the application is created, Argo CD will synchronize and apply all the YAML files defined in the GitHub repository.



The screenshot shows a GitHub repository interface for a file named 'deploy.yaml'. The file content is as follows:

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx-deployment
5    labels:
6      app: nginx
7  spec:
8    replicas: 10
9    selector:
10     matchLabels:
11       app: nginx
12  template:
13    metadata:
14      labels:
15        app: nginx
16    spec:
17      containers:
18      - name: nginx
19        image: nginx:1.14.2
20        ports:
21        - containerPort: 80
```

I updated the YAML file to increase the replica count from 3 to 10 and enabled auto-sync in Argo CD.



The screenshot displays the Argo CD web interface. At the top, the 'APP HEALTH' is 'Healthy'. The 'SYNC STATUS' shows 'Synced to HEAD (96af1a8)'. The 'LAST SYNC' status is 'Sync OK to 96af1a8', with a message 'Succeeded a few seconds ago (Thu Apr 17 2025 16:16:10 GMT+0530)'. Below this, a diagram shows the application 'myapp' syncing to a 'deploy' resource, which then syncs to a 'Kind: ReplicaSet' resource. To the right, a list of pods is shown, all in a 'Running' state. At the bottom, a terminal window shows the output of the command 'kubectl get pods', listing various Argo CD components and the nginx deployment pods.

NAME	READY	STATUS	RESTARTS	AGE
my-argo-cd-argocd-application-controller-0	1/1	Running	1 (59m ago)	21h
my-argo-cd-argocd-applicationset-controller-869cf6d798-7pkc4	1/1	Running	1 (59m ago)	21h
my-argo-cd-argocd-dex-server-5f8dd87dfd-gwzkg	1/1	Running	1 (59m ago)	21h
my-argo-cd-argocd-notifications-controller-688bf6796b-1k7x1	1/1	Running	1 (59m ago)	21h
my-argo-cd-argocd-redis-5566777d97-2bvd1	1/1	Running	1 (59m ago)	21h
my-argo-cd-argocd-repo-server-84f9b986cd-jmqmd	1/1	Running	1 (59m ago)	21h
my-argo-cd-argocd-server-598f9c45b4-n2tk5	1/1	Running	1 (59m ago)	21h
nginx-deployment-d556bf558-9g98g	1/1	Running	0	9m15s
nginx-deployment-d556bf558-9wplp	1/1	Running	0	9m15s
nginx-deployment-d556bf558-f9gsd	1/1	Running	0	26s
nginx-deployment-d556bf558-fk7nm	1/1	Running	0	26s
nginx-deployment-d556bf558-q5dzk	1/1	Running	0	26s
nginx-deployment-d556bf558-qvhfv	1/1	Running	0	26s
nginx-deployment-d556bf558-s9jfp	1/1	Running	0	26s
nginx-deployment-d556bf558-sm7h8	1/1	Running	0	5m20s
nginx-deployment-d556bf558-t6hdc	1/1	Running	0	26s
nginx-deployment-d556bf558-v9pkm	1/1	Running	0	26s

With auto-sync enabled, Argo CD will automatically detect the change and apply the updated configuration to the cluster