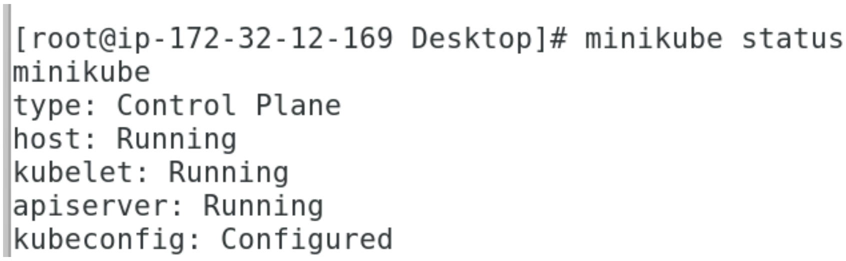
Once the minikube is running successfully then,



```
[root@ip-172-32-12-169 Desktop]# minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

1)First create a name space by using the below command

→kubectl create namespace argocd

2)execute this command to download :

kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

3)To watch the pods :

Kubectl get pods -n argocd -w

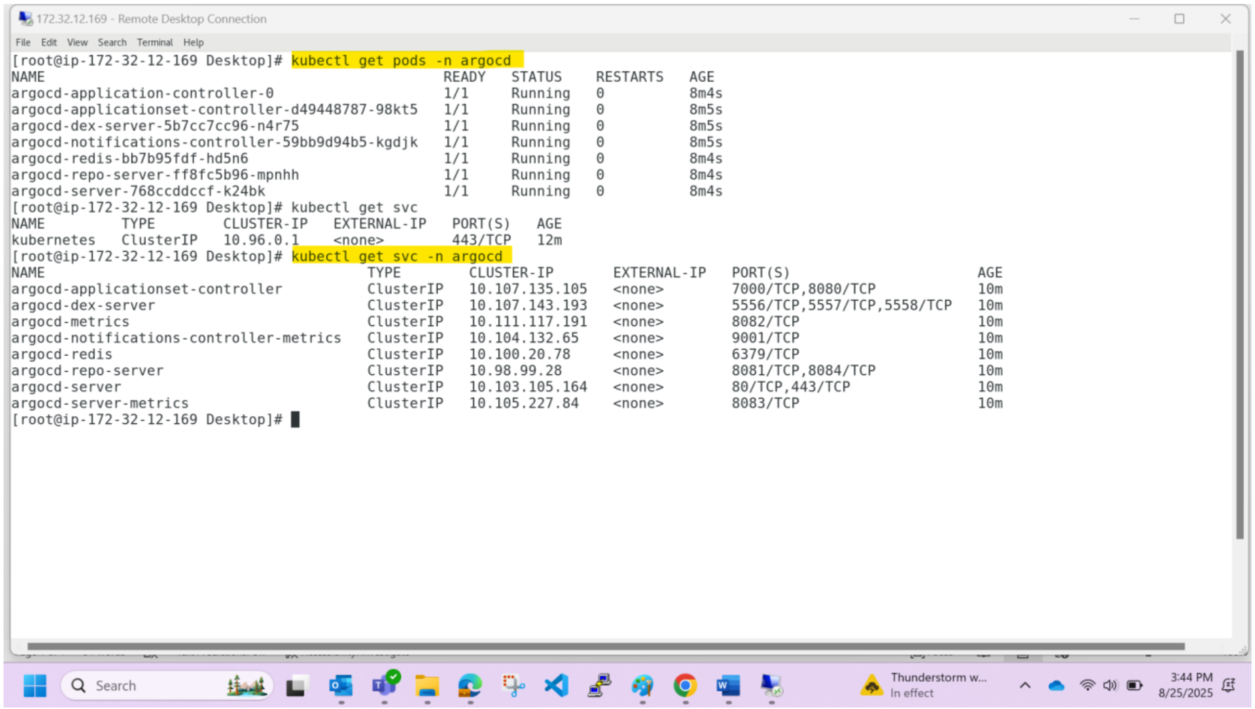
It will take time and if it taking more time then use ctrl+c

→pods got installed ok

To check the pods → Kubectl get pods -n argocd

4)Now see the services that are created, why exactly we need services is bcoz we need to interact with argo cd UI

→kubectl get svc -n argocd

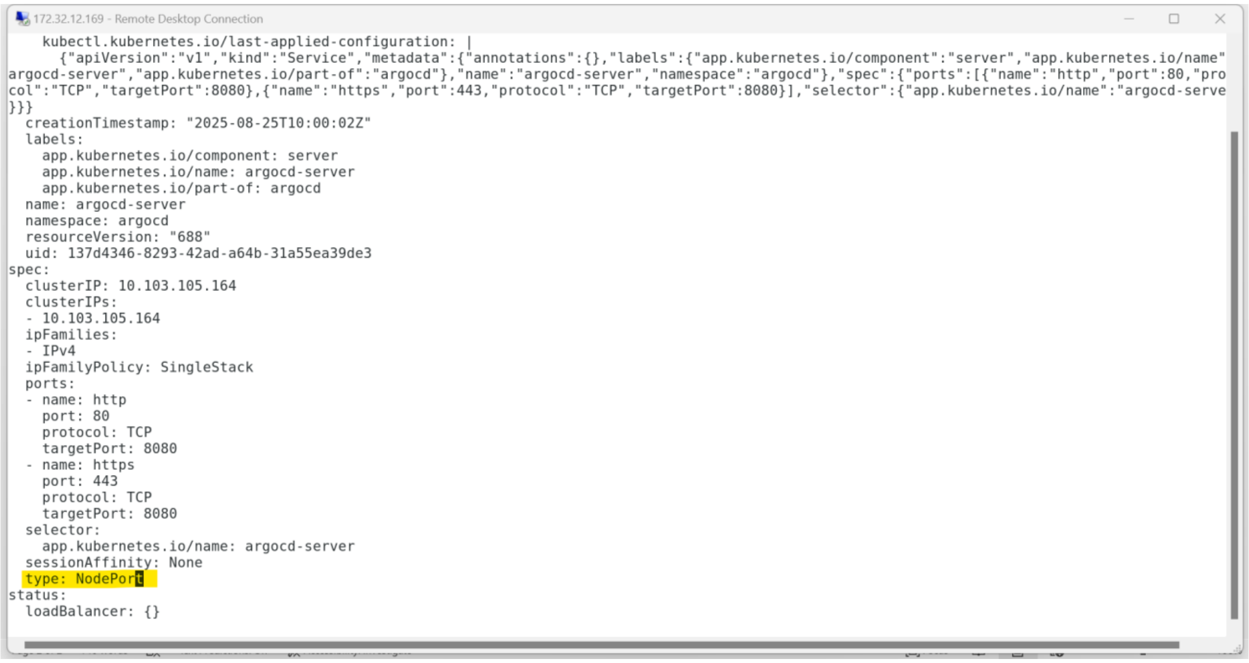

172.32.12.169 - Remote Desktop Connection
File Edit View Search Terminal Help
Search
Thunderstorm w...
In effect
3:44 PM
8/25/2025

```
[root@ip-172-32-12-169 Desktop]# kubectl get pods -n argocd
NAME                                               READY   STATUS    RESTARTS   AGE
argocd-application-controller-0                    1/1     Running   0          8m4s
argocd-applicationset-controller-d49448787-98kt5   1/1     Running   0          8m5s
argocd-dex-server-5b7cc7cc96-n4r75                 1/1     Running   0          8m5s
argocd-notifications-controller-59bb9d94b5-kgdjk   1/1     Running   0          8m5s
argocd-redis-bb7b95fdf-hd5n6                       1/1     Running   0          8m4s
argocd-repo-server-ff8fc5b96-mpnhh                 1/1     Running   0          8m4s
argocd-server-768ccddccf-k24bk                     1/1     Running   0          8m4s
[root@ip-172-32-12-169 Desktop]# kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   12m
[root@ip-172-32-12-169 Desktop]# kubectl get svc -n argocd
NAME                                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
argocd-applicationset-controller          ClusterIP   10.107.135.105   <none>        7000/TCP,8080/TCP            10m
argocd-dex-server                         ClusterIP   10.107.143.193   <none>        5556/TCP,5557/TCP,5558/TCP   10m
argocd-metrics                            ClusterIP   10.111.117.191   <none>        8082/TCP                     10m
argocd-notifications-controller-metrics   ClusterIP   10.104.132.65    <none>        9001/TCP                     10m
argocd-redis                              ClusterIP   10.100.20.78     <none>        6379/TCP                     10m
argocd-repo-server                        ClusterIP   10.98.99.28      <none>        8081/TCP,8084/TCP            10m
argocd-server                             ClusterIP   10.103.105.164   <none>        80/TCP,443/TCP               10m
argocd-server-metrics                     ClusterIP   10.105.227.84    <none>        8083/TCP                     10m
[root@ip-172-32-12-169 Desktop]# █
```

5)We have a service called an arocd-server(this service is responsible for interacting with UI or CLI)

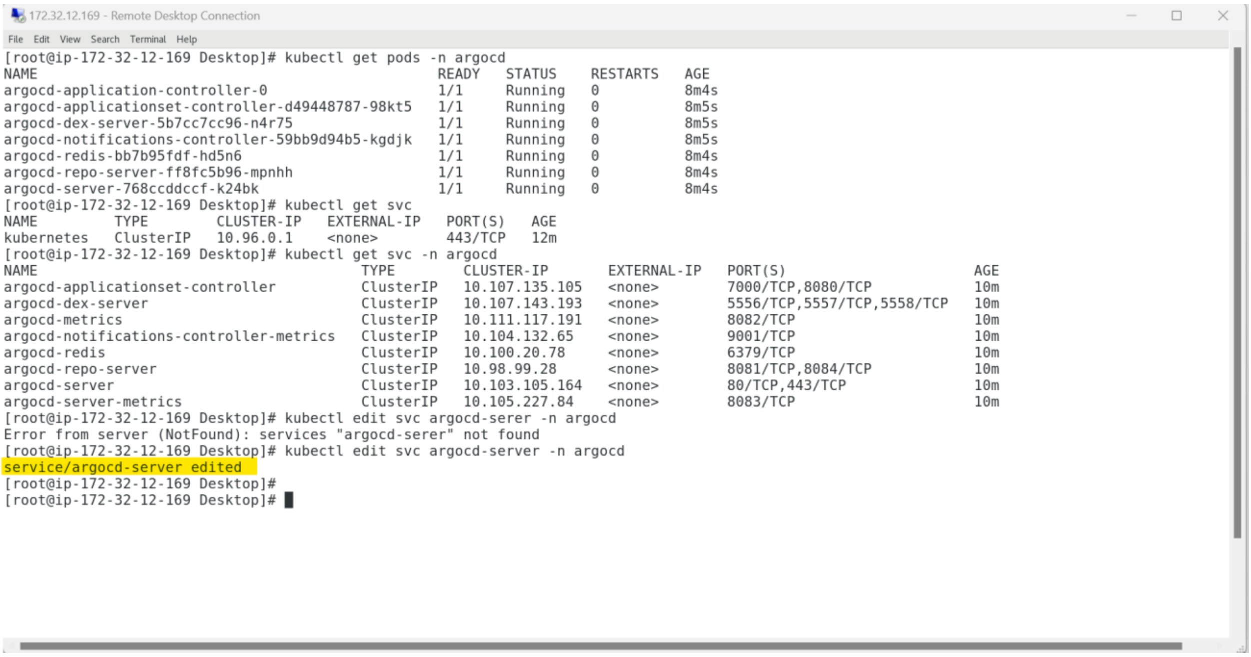
→ kubectl edit svc argocd-server -n argocd

Execute the above command and why we are doing this is because by default this service comes up with ‘ClusteIP’



```
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app.kubernetes.io/component":"server","app.kubernetes.io/name"
argocd-server","app.kubernetes.io/part-of":"argocd"},"name":"argocd-server","namespace":"argocd"},"spec":{"ports":[{"name":"http","port":80,"pro
col":"TCP","targetPort":8080},{"name":"https","port":443,"protocol":"TCP","targetPort":8080}],"selector":{"app.kubernetes.io/name":"argocd-serve
}}}
  creationTimestamp: "2025-08-25T10:00:02Z"
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: argocd-server
    app.kubernetes.io/part-of: argocd
  name: argocd-server
  namespace: argocd
  resourceVersion: "688"
  uid: 137d4346-8293-42ad-a64b-31a55ea39de3
spec:
  clusterIP: 10.103.105.164
  clusterIPs:
  - 10.103.105.164
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 8080
  - name: https
    port: 443
    protocol: TCP
    targetPort: 8080
  selector:
    app.kubernetes.io/name: argocd-server
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}
```

Now service got changed to ClusteIp to Nodeport

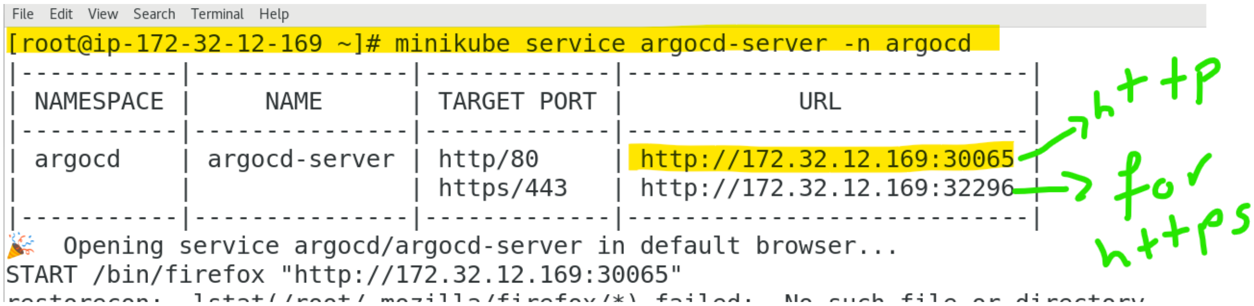


172.32.12.169 - Remote Desktop Connection
File Edit View Search Terminal Help

```
[root@ip-172-32-12-169 Desktop]# kubectl get pods -n argocd
NAME                                                READY   STATUS    RESTARTS   AGE
argocd-application-controller-0                     1/1     Running   0          8m4s
argocd-applicationset-controller-d49448787-98kt5    1/1     Running   0          8m5s
argocd-dex-server-5b7cc7cc96-n4r75                  1/1     Running   0          8m5s
argocd-notifications-controller-59bb9d94b5-kgdjk    1/1     Running   0          8m5s
argocd-redis-bb7b95fdf-hd5n6                        1/1     Running   0          8m4s
argocd-repo-server-ff8fc5b96-mpnhh                  1/1     Running   0          8m4s
argocd-server-768ccddccf-k24bk                      1/1     Running   0          8m4s
[root@ip-172-32-12-169 Desktop]# kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   12m
[root@ip-172-32-12-169 Desktop]# kubectl get svc -n argocd
NAME                                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
argocd-applicationset-controller          ClusterIP   10.107.135.105   <none>        7000/TCP,8080/TCP            10m
argocd-dex-server                         ClusterIP   10.107.143.193   <none>        5556/TCP,5557/TCP,5558/TCP   10m
argocd-metrics                            ClusterIP   10.111.117.191   <none>        8082/TCP                     10m
argocd-notifications-controller-metrics   ClusterIP   10.104.132.65    <none>        9001/TCP                     10m
argocd-redis                              ClusterIP   10.100.20.78     <none>        6379/TCP                     10m
argocd-repo-server                        ClusterIP   10.98.99.28      <none>        8081/TCP,8084/TCP            10m
argocd-server                             ClusterIP   10.103.105.164   <none>        80/TCP,443/TCP               10m
argocd-server-metrics                     ClusterIP   10.105.227.84    <none>        8083/TCP                     10m
[root@ip-172-32-12-169 Desktop]# kubectl edit svc argocd-serer -n argocd
Error from server (NotFound): services "argocd-serer" not found
[root@ip-172-32-12-169 Desktop]# kubectl edit svc argocd-server -n argocd
service/argocd-server edited
[root@ip-172-32-12-169 Desktop]#
[root@ip-172-32-12-169 Desktop]# █
```

Now I can access this from terminal itself once the type is changed from clusteIP to Nodeport

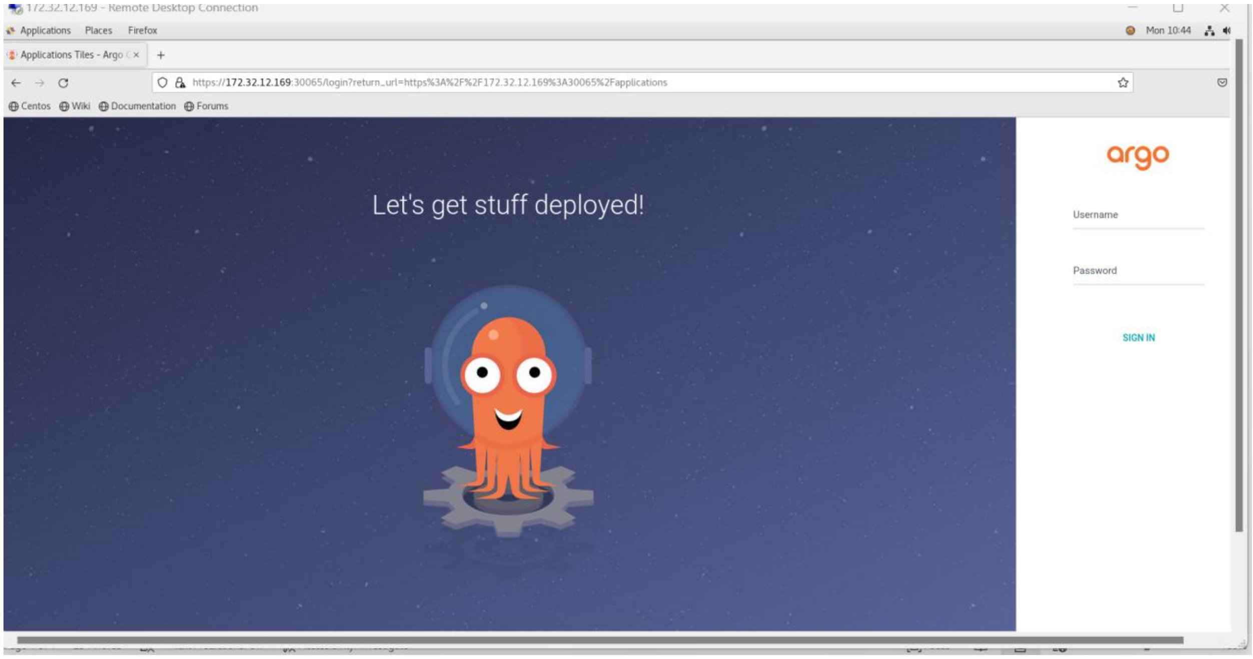
6)Now execute the command minikube service argocd-server -n argocd , here minikube will create you a tunnel and will provide you an IP address to access the argocd UI.



```
[root@ip-172-32-12-169 ~]# minikube service argocd-server -n argocd
|-----------|---------------|-------------|----------------------------|
| NAMESPACE |     NAME      | TARGET PORT |            URL             |
|-----------|---------------|-------------|----------------------------|
| argocd    | argocd-server | http/80     | http://172.32.12.169:30065 |
|           |               | https/443   | http://172.32.12.169:32296 |
|-----------|---------------|-------------|----------------------------|
🎉  Opening service argocd/argocd-server in default browser...
START /bin/firefox "http://172.32.12.169:30065"
restorecon:  lstat(/root/.mozilla/firefox/*) failed:  No such file or directory
```

Now take the http link and paste it on the browser and and you get a pop and go with “Advanced” then go with “Accept the Risk and continue”

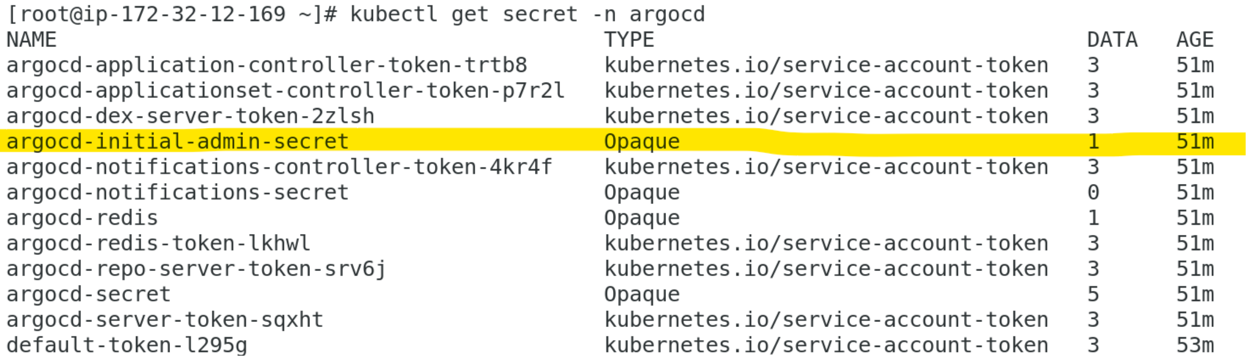
→Now you will get the UI page of argocd as shown below:



172.32.12.169 - Remote Desktop Connection
Applications
Places
Firefox
Mon 10:44
Applications Tiles - Argo C
https://172.32.12.169:30065/login?return_url=https%3A%2F%2F172.32.12.169%3A30065%2Fapplications
Centos
Wiki
Documentation
Forums
Let's get stuff deployed!
argo
Username
Password
SIGN IN

7)Now for getting the secret execute the below command

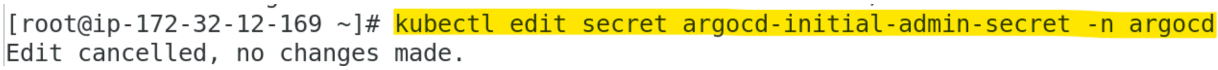
→ kubectl get secret -n argocd



```
[root@ip-172-32-12-169 ~]# kubectl get secret -n argocd
NAME                                           TYPE                                  DATA   AGE
argocd-application-controller-token-trtb8      kubernetes.io/service-account-token   3      51m
argocd-applicationset-controller-token-p7r2l   kubernetes.io/service-account-token   3      51m
argocd-dex-server-token-2zlsh                  kubernetes.io/service-account-token   3      51m
argocd-initial-admin-secret                    Opaque                                1      51m
argocd-notifications-controller-token-4kr4f    kubernetes.io/service-account-token   3      51m
argocd-notifications-secret                    Opaque                                0      51m
argocd-redis                                   Opaque                                1      51m
argocd-redis-token-lkhwl                       kubernetes.io/service-account-token   3      51m
argocd-repo-server-token-srv6j                 kubernetes.io/service-account-token   3      51m
argocd-secret                                  Opaque                                5      51m
argocd-server-token-sqxht                      kubernetes.io/service-account-token   3      51m
default-token-l295g                            kubernetes.io/service-account-token   3      53m
```

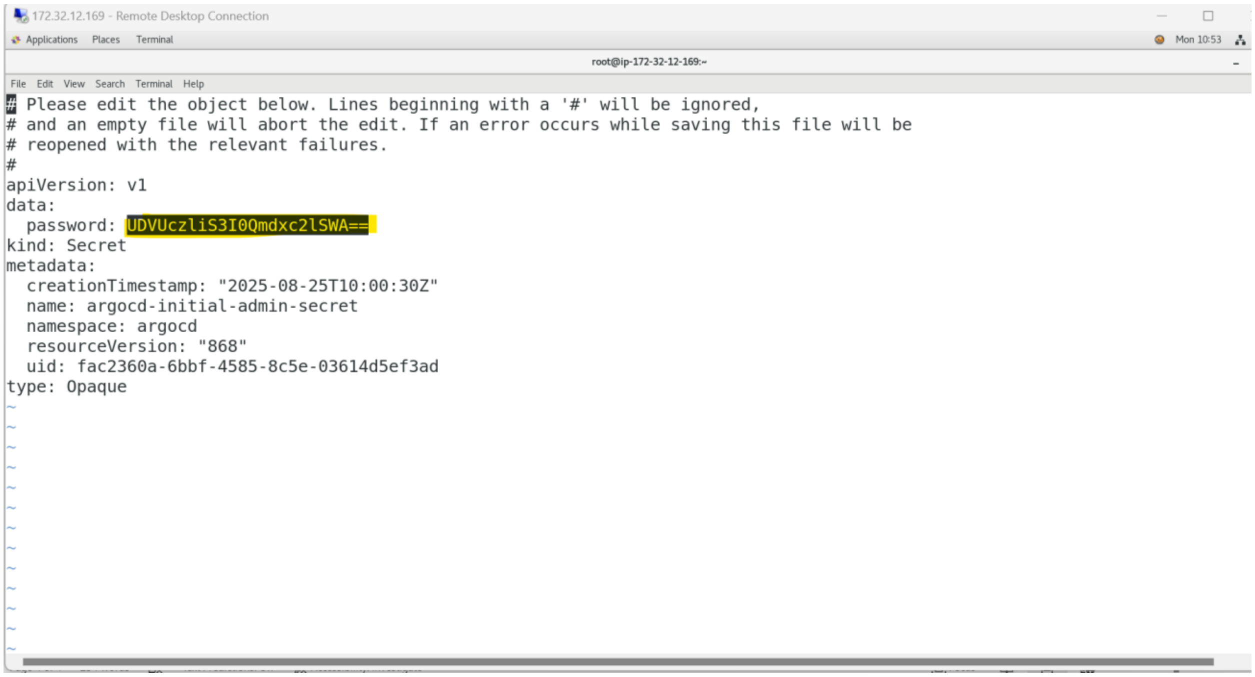
Now use the command to take the encrypted password by using the command

→kubectl edit secret argocd-initial-admin-secret -n argocd



```
[root@ip-172-32-12-169 ~]# kubectl edit secret argocd-initial-admin-secret -n argocd
Edit cancelled, no changes made.
```

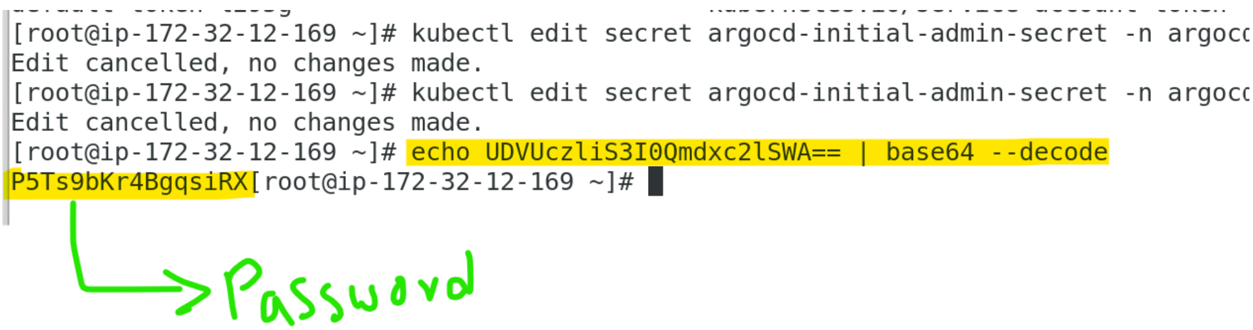
Now copy the password and don't make any changes just quit it (esc :q!)



```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  password: UDVUczliS3I0Qmdxc2lSWA==
kind: Secret
metadata:
  creationTimestamp: "2025-08-25T10:00:30Z"
  name: argocd-initial-admin-secret
  namespace: argocd
  resourceVersion: "868"
  uid: fac2360a-6bbf-4585-8c5e-03614d5ef3ad
type: Opaque
~
~
~
~
~
~
~
~
~
~
~
~
~
```

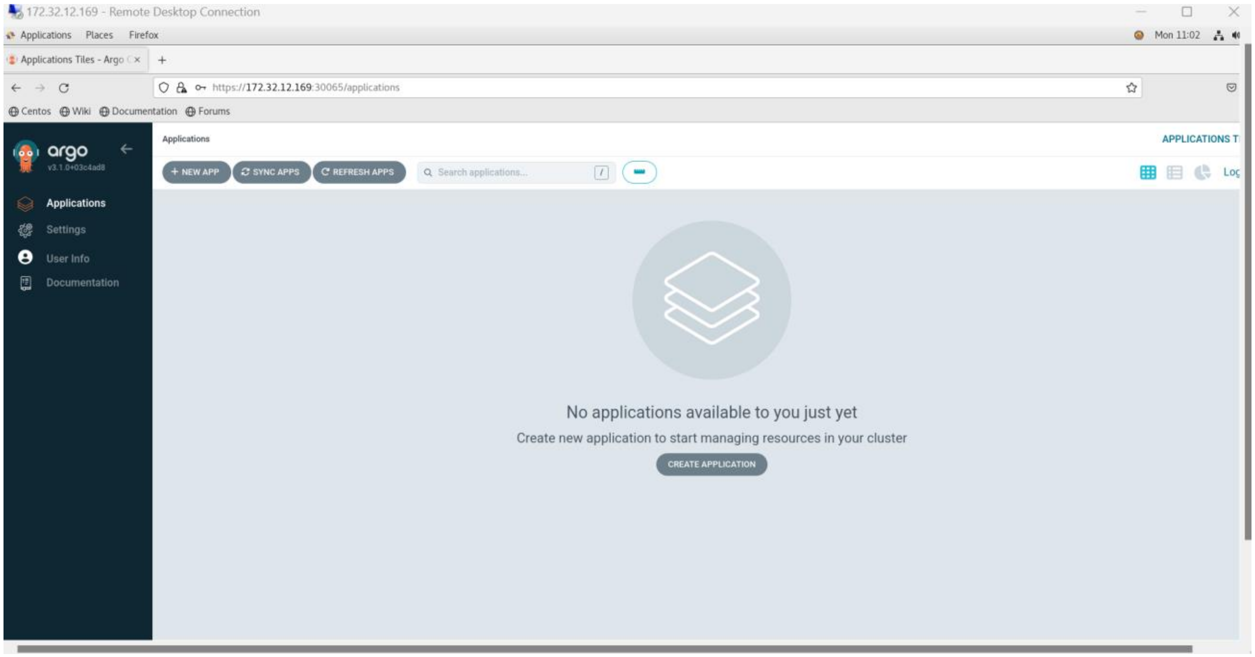
Now secret is encrypted in a base 64 way and we need to decrypt that ok

Execute "echo UDVUczlis#I0Qmdxc2lsWA== | base64 –decode"



```
[root@ip-172-32-12-169 ~]# kubectl edit secret argocd-initial-admin-secret -n argoc
Edit cancelled, no changes made.
[root@ip-172-32-12-169 ~]# kubectl edit secret argocd-initial-admin-secret -n argoc
Edit cancelled, no changes made.
[root@ip-172-32-12-169 ~]# echo UDVUczliS3I0Qmdxc2lSWA== | base64 --decode
P5Ts9bKr4BgqsiRX[root@ip-172-32-12-169 ~]#
```

Now after giving user name as admin and password argocd UI page will come up as shown in below



No applications available to you just yet

Create new application to start managing resources in your cluster