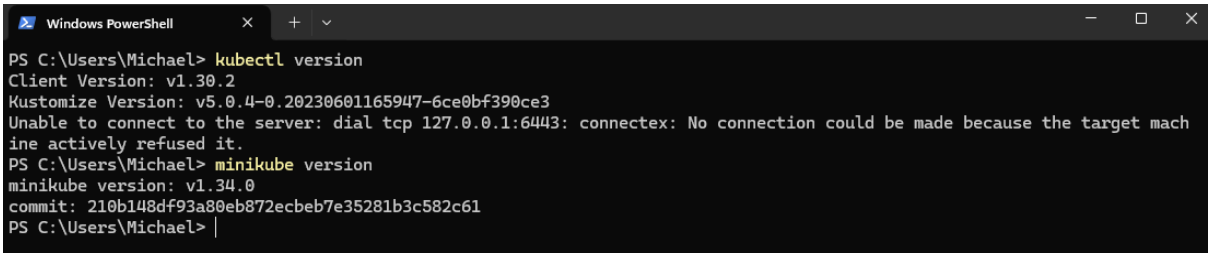
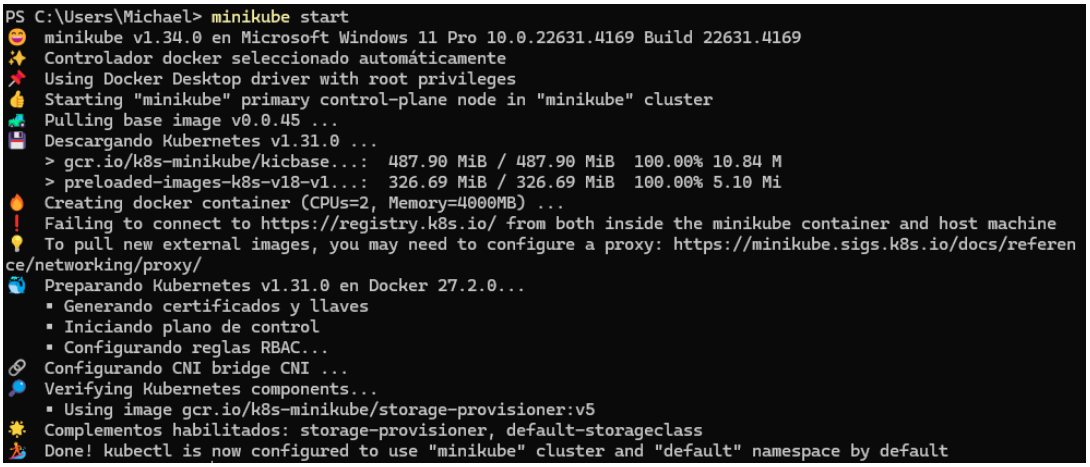
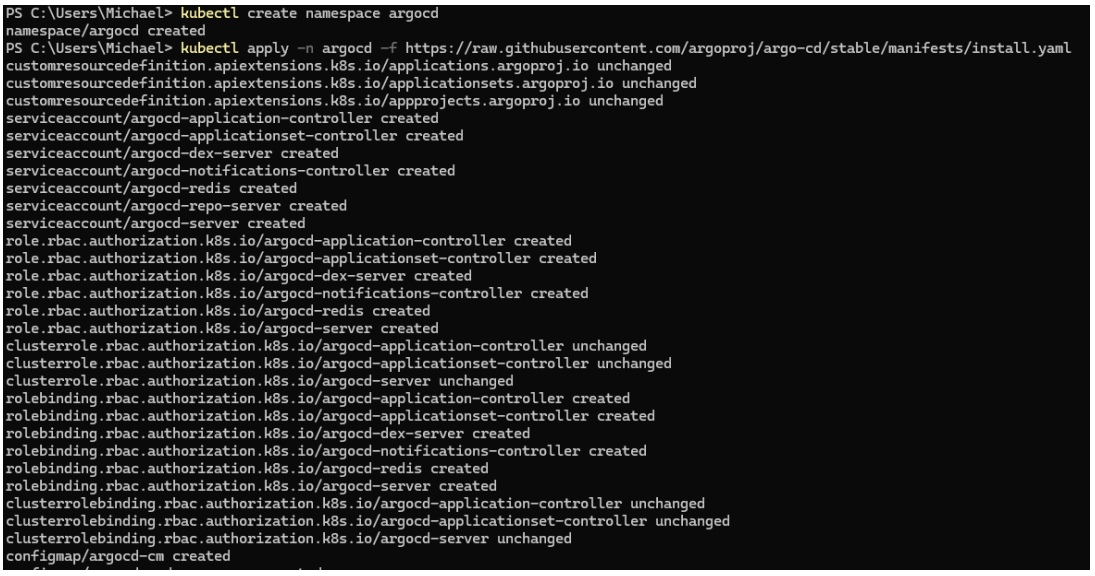
Desafío 13

Este nuevo desafío lo iniciares instalando **Minikube** y **kubectl** para administrar el cluster seguidamente abrimos la aplicación de **Docker** ya que **Minikube** lo usa para gestionar los cluster.



Seguido de esto ejecutamos en el powershell **Minikube** con el comando **Minikube start** y creamos el namespace de **ArgoCD** con el comando **kubectl create namespace argocd** luego de eso desplegamos los recursos de **ArgoCD** en **Minikube** con el comando **kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml**

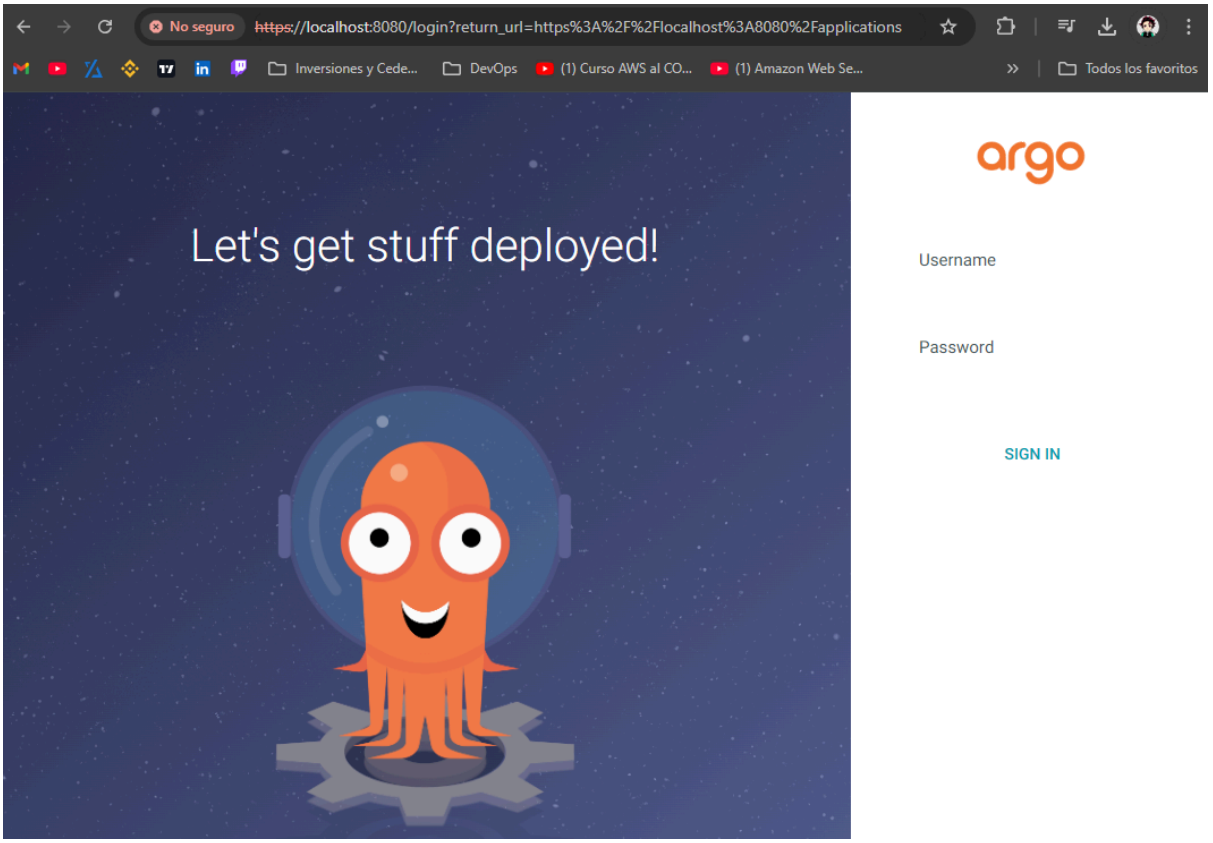




Exponemos el servicio de **ArgoCD** para acceder al dashboard con el comando **kubectl port-forward svc/argocd-server -n argocd 8080:443**

```
PS C:\Users\Michael> kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
```

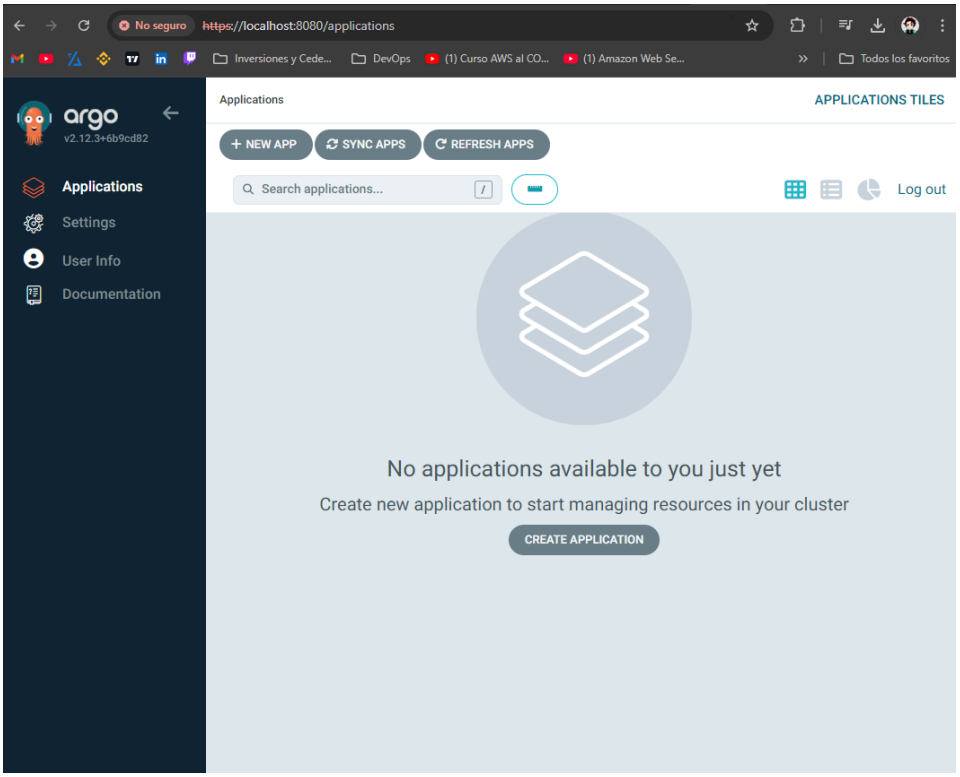
Ingresamos a nuestra dirección IP seguido del puerto 8080 https://localhost:8080



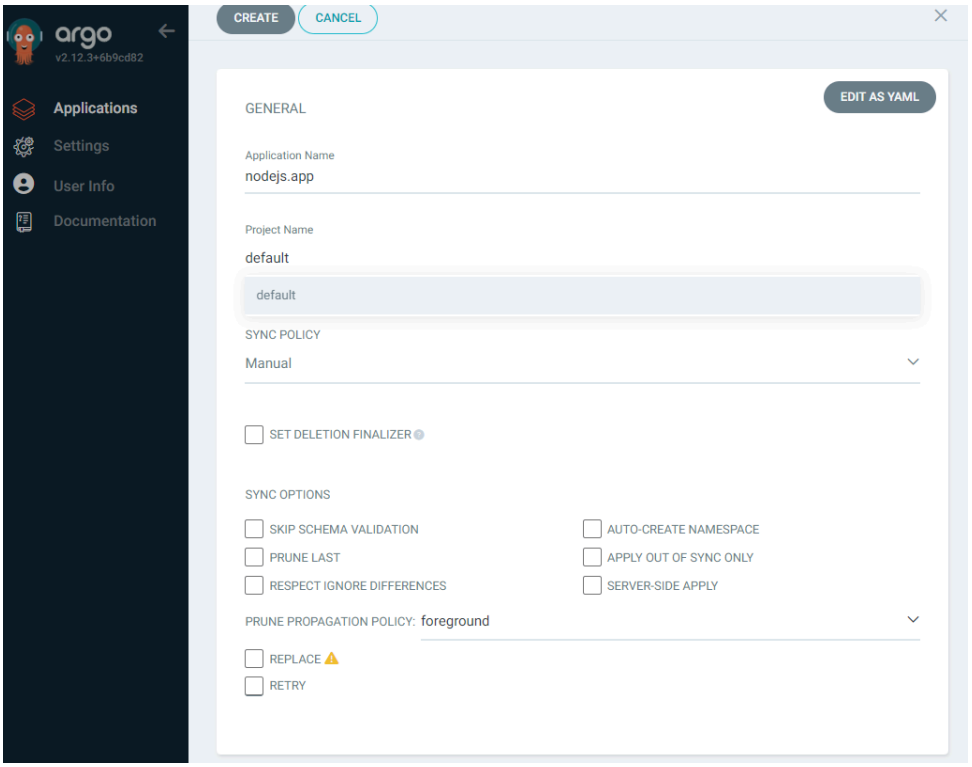
Seguido de esto obtenemos la contraseña del usuario admin para ingresar a **ArgoCD**, primero ingresamos el comando **kubectl get secret argocd-initial-admin-secret -n argocd -o jsonpath="{.data.password}"** para obtener el valor codificado en **base64** de la contraseña.
Luego con el comando **[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("<your-base64-password>"))** la decodificamos. Reemplazamos **"<your-base64-password>"** con la cadena que obtuvimos en el primer paso.

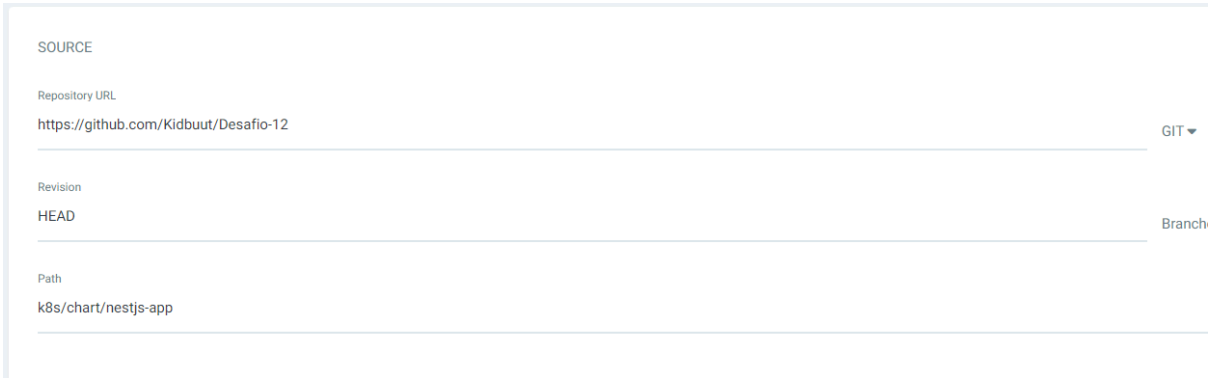
```
PS C:\Users\Michael> kubectl get secret argocd-initial-admin-secret -n argocd -o jsonpath="{.data.password}"
cDktSG5BeXR3Yk5QWThIdA==
PS C:\Users\Michael> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("cDktSG5BeXR3Yk5QWThIdA=="))
p9-HnAytwbNPY8Ht
PS C:\Users\Michael>
```

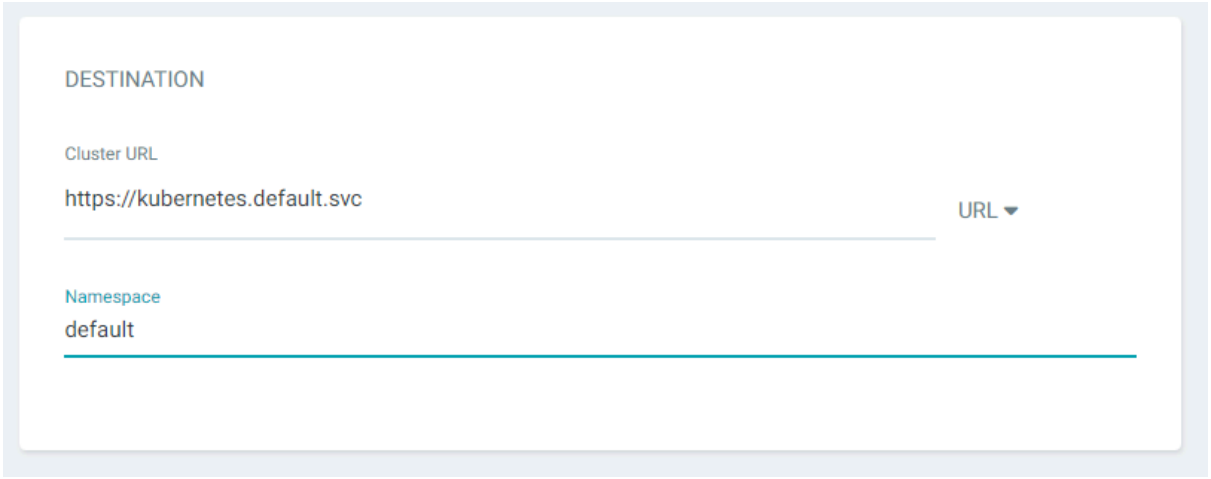
Luego accedemos nuevamente a nuestro https://localhost:8080 corriendo el comando en en powershell **kubectl port-forward svc/argocd-server -n argocd 8080:443** e ingresamos al dashboard con el usuario **admin** y el **password** descifrado en el paso anterior



Para crear nuestra aplicación nos dirigiremos a **Applications -> New App ->** y completamos los campos con los datos solicitados




SOURCE
Repository URL
https://github.com/Kidbuut/Desafio-12
GIT
Revision
HEAD
Branch
Path
k8s/chart/nestjs-app


DESTINATION
Cluster URL
https://kubernetes.default.svc
URL
Namespace
default

CREATE CANCEL

PARAMETERS

autoscaling.enabled	false
autoscaling.maxReplicas	100
autoscaling.minReplicas	1
autoscaling.targetCPUUtilizationPercentage	80
env.MONGO_DB_NAME	test
env.MONGO_DB_PASS	mongo123
env.MONGO_DB_URI	mongodb://mongo:mongo123@mongodb-service:27017/test
env.MONGO_DB_USER	mongo
ingress.enabled	false
ingress.hosts[0].host	chart-example.local
mongodb.dbName	test
mongodb.dbPassword	mongo123
mongodb.dbUser	mongo
mongodb.image	mongo
mongodb.replicaCount	1

Una vez tengamos la configuración completa, vemos cómo se despliegan