

Desafío # 10

Fecha de entrega: 27/01/2025

Objetivo:

El objetivo de este desafío es guiar la instalación y configuración de ArgoCD en un entorno de Kubernetes. ArgoCD es una herramienta de GitOps utilizada para gestionar despliegues en Kubernetes de manera declarativa. A través de este ejercicio, se pondrán en práctica los pasos necesarios para implementar una infraestructura automatizada de despliegue continuo.

Escenario:

Partiendo de un entorno local de Kubernetes utilizando algunas de las soluciones vistas en clase como plataforma de desarrollo (Killercode, Kc0crl, Minikube, K3s, etc). El equipo de trabajo ha solicitado la implementación de ArgoCD para gestionar el flujo de despliegues de aplicaciones de manera automatizada. Se utilizarán manifiestos de Kubernetes y se integrará la plataforma con un repositorio Git para la gestión de aplicaciones.

Requisitos:

1. Instalar Kubectl
2. Instalar ArgoCD
3. Conectar un repositorio de Github

Entregables:

Los entregables establecidos para este proyecto con:

1. Código fuente de los manifiestos utilizados en el despliegue de ArgoCD y una aplicación gestionada.

2. Documentación detallada de los pasos seguidos para la instalación de ArgoCD en Minikube.
3. Evidencia del funcionamiento de ArgoCD con capturas del dashboard mostrando el despliegue exitoso de la aplicación.

Evaluación:

- Entrega en fecha.
- Redactar documentación legible y que sea comprendida por terceros.
- El entregable debe ser funcional, permitiendo la instalación de ArgoCD y el despliegue de una aplicación de ejemplo desde un repositorio Git.
- Añade material de soporte adicional.
 - Ejemplo: Diagrama de alto nivel.
- Cumple con las consignas solicitadas.
- El entregable es funcional.
 - Ejemplo: el script bash al ejecutarse funciona sin errores y realiza lo solicitado.

Documentos de referencia:

- <https://kubernetes.io/docs/tasks/tools/>
- https://argo-cd.readthedocs.io/en/stable/getting_started/
- <https://github.com/argoproj/argocd-example-apps>

Solución:

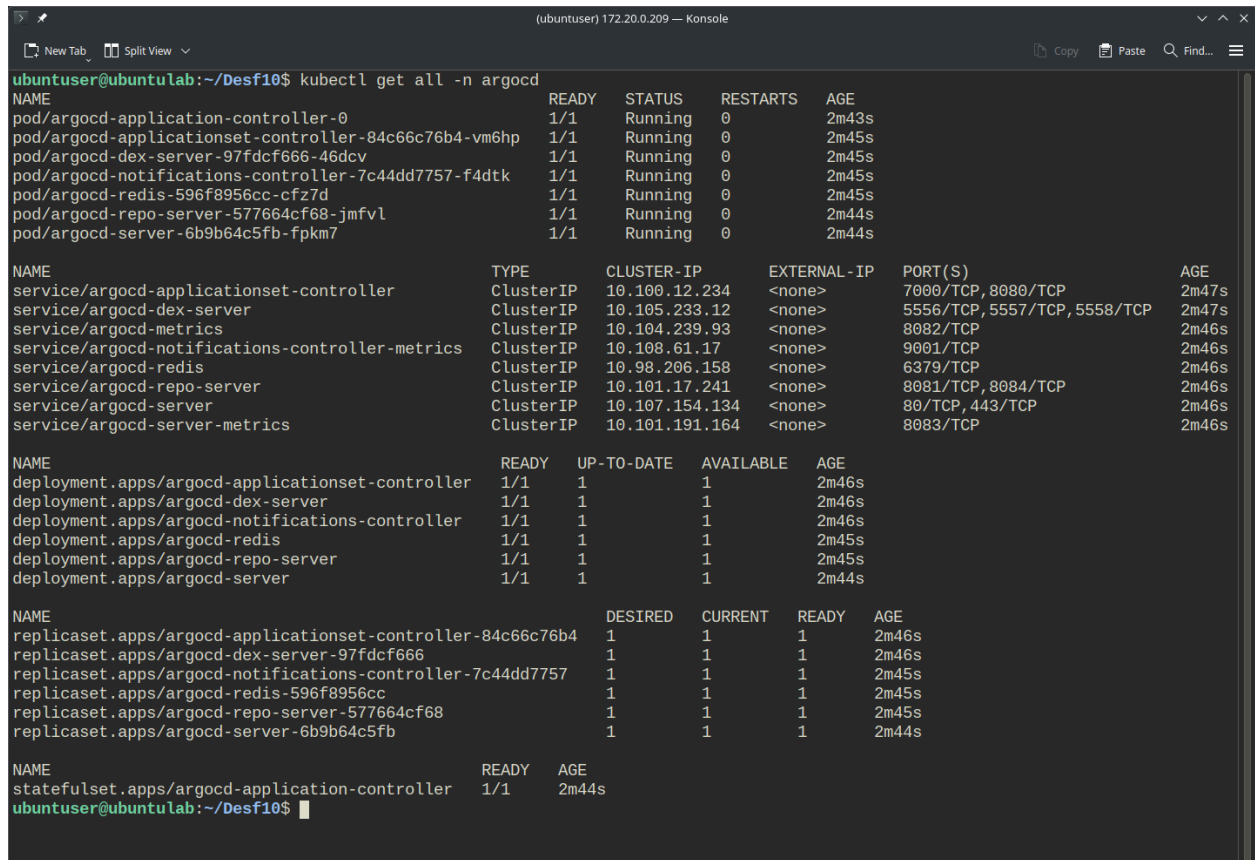
Setup:

Voy a instalar helm en mi server Ubuntu con minikube siguiendo las instrucciones de aqui:

https://argo-cd.readthedocs.io/en/stable/getting_started/

```
kubectl create namespace argocd
kubectl apply -n argocd -f
https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
1
```

luego de un momento se verifica con `kubectl get all -n argocd` (si no se especifica el namespace mostrará el default) y se verifica que todos los pods y servicios estan arriba.



```
(ubuntuuser) 172.20.0.209 — Konsole
New Tab Split View
Copy Paste Find...
ubuntuuser@ubuntulab:~/Desf10$ kubectl get all -n argocd
NAME                                     READY   STATUS    RESTARTS   AGE
pod/argocd-application-controller-0      1/1     Running   0          2m43s
pod/argocd-applicationset-controller-84c66c76b4-vm6hp  1/1     Running   0          2m45s
pod/argocd-dex-server-97fdcf666-46dcv    1/1     Running   0          2m45s
pod/argocd-notifications-controller-7c44dd7757-f4dtk  1/1     Running   0          2m45s
pod/argocd-redis-596f8956cc-cfz7d        1/1     Running   0          2m45s
pod/argocd-repo-server-577664cf68-jmfvl   1/1     Running   0          2m44s
pod/argocd-server-6b9b64c5fb-fpkm7       1/1     Running   0          2m44s

NAME                                     TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/argocd-applicationset-controller  ClusterIP     10.100.12.234    <none>           7000/TCP,8080/TCP 2m47s
service/argocd-dex-server                 ClusterIP     10.105.233.12    <none>           5556/TCP,5557/TCP,5558/TCP 2m47s
service/argocd-metrics                     ClusterIP     10.104.239.93    <none>           8082/TCP          2m46s
service/argocd-notifications-controller-metrics  ClusterIP     10.108.61.17     <none>           9001/TCP          2m46s
service/argocd-redis                       ClusterIP     10.98.206.158    <none>           6379/TCP          2m46s
service/argocd-repo-server                 ClusterIP     10.101.17.241    <none>           8081/TCP,8084/TCP 2m46s
service/argocd-server                      ClusterIP     10.107.154.134   <none>           80/TCP,443/TCP   2m46s
service/argocd-server-metrics              ClusterIP     10.101.191.164   <none>           8083/TCP          2m46s

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/argocd-applicationset-controller  1/1     1             1           2m46s
deployment.apps/argocd-dex-server                1/1     1             1           2m46s
deployment.apps/argocd-notifications-controller   1/1     1             1           2m46s
deployment.apps/argocd-redis                      1/1     1             1           2m45s
deployment.apps/argocd-repo-server                1/1     1             1           2m45s
deployment.apps/argocd-server                     1/1     1             1           2m44s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/argocd-applicationset-controller-84c66c76b4  1         1         1       2m46s
replicaset.apps/argocd-dex-server-97fdcf666                 1         1         1       2m46s
replicaset.apps/argocd-notifications-controller-7c44dd7757   1         1         1       2m45s
replicaset.apps/argocd-redis-596f8956cc                     1         1         1       2m45s
replicaset.apps/argocd-repo-server-577664cf68                1         1         1       2m45s
replicaset.apps/argocd-server-6b9b64c5fb                    1         1         1       2m44s

NAME                                     READY   AGE
statefulset.apps/argocd-application-controller  1/1     2m44s
ubuntuuser@ubuntulab:~/Desf10$
```

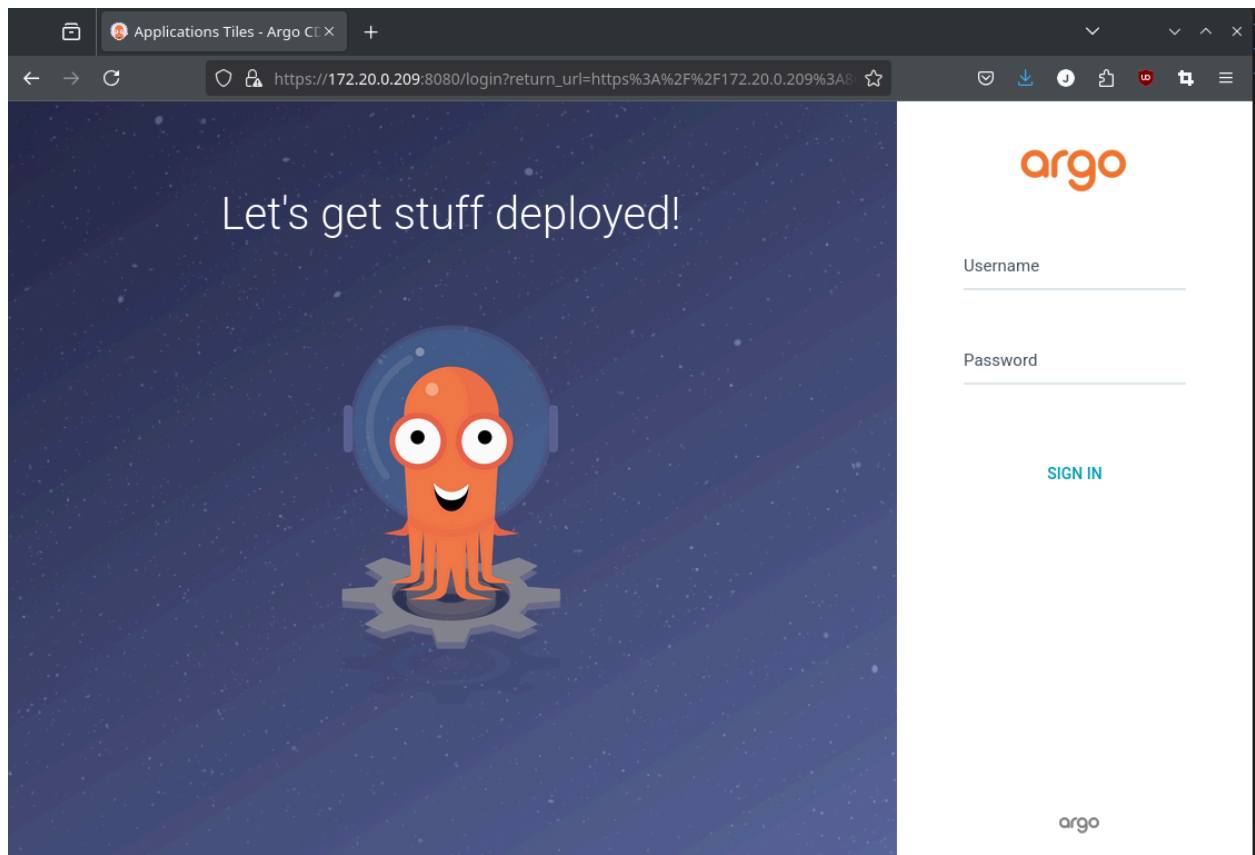
Nota: acá tuve un problemon. Resulta que para conectarme a la interfaz de administración de Argo se necesita crear un tunnel, y la documentación sugiere un patch para cambiar el type del repo-server a LoadBalancer y usar el comando;

```
kubectl port-forward svc/argocd-server -n argocd 8080:443
```

No hubo manera de hacerlo andar. Pero pude aplicar un workaround, sin hacer el patch y aplicando este comando:

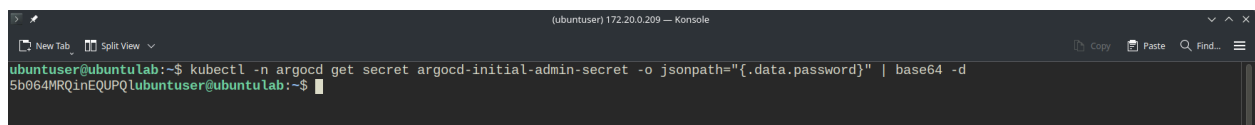
```
kubectl port-forward svc/argocd-server -n argocd 8080:443 --address 0.0.0.0
```

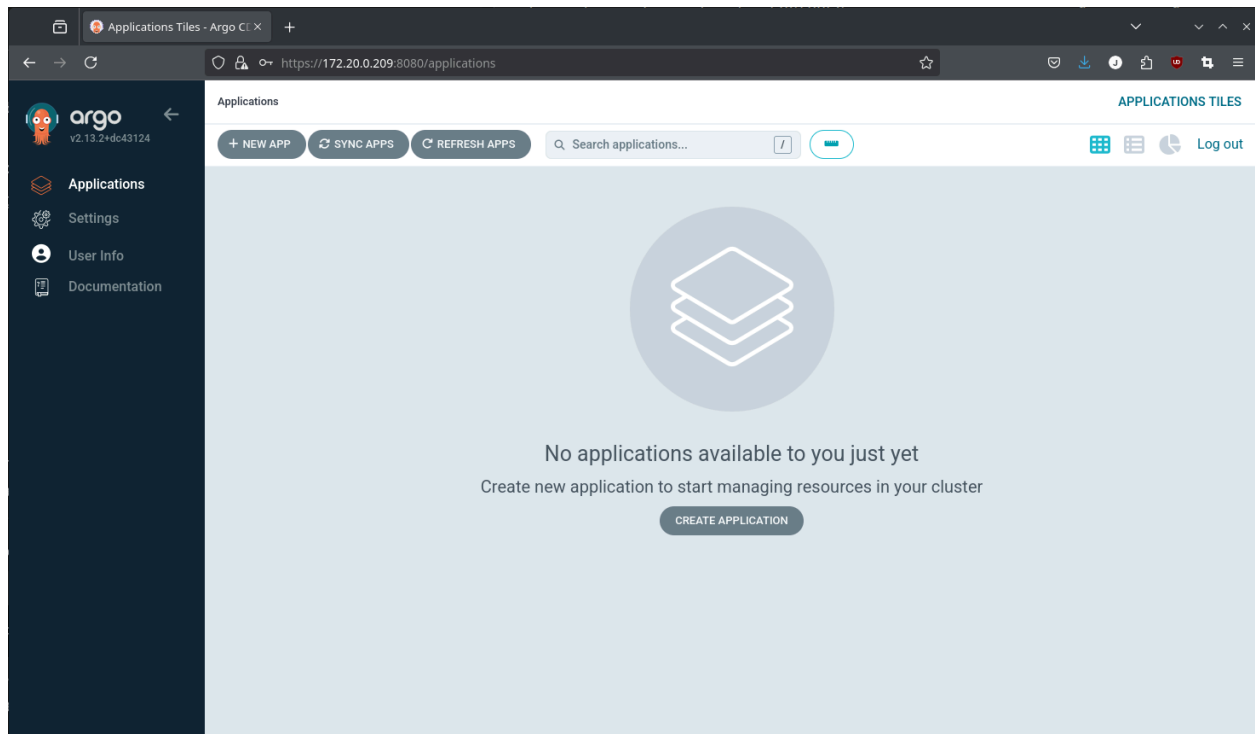
pude conectarme por medio del browser remoto:



Y con las instrucciones de <https://medium.com/@mehmetdabashi/installing-argocd-on-minikube-and-deploying-a-test-application-caa68ec55fbf> pude obtener la clave de administracion de la interfaz de Argocd sin el ArgoCLI

```
kubectl -n argocd get secret argocd-initial-admin-secret -o  
jsonpath="{.data.password}" | base64 -d
```



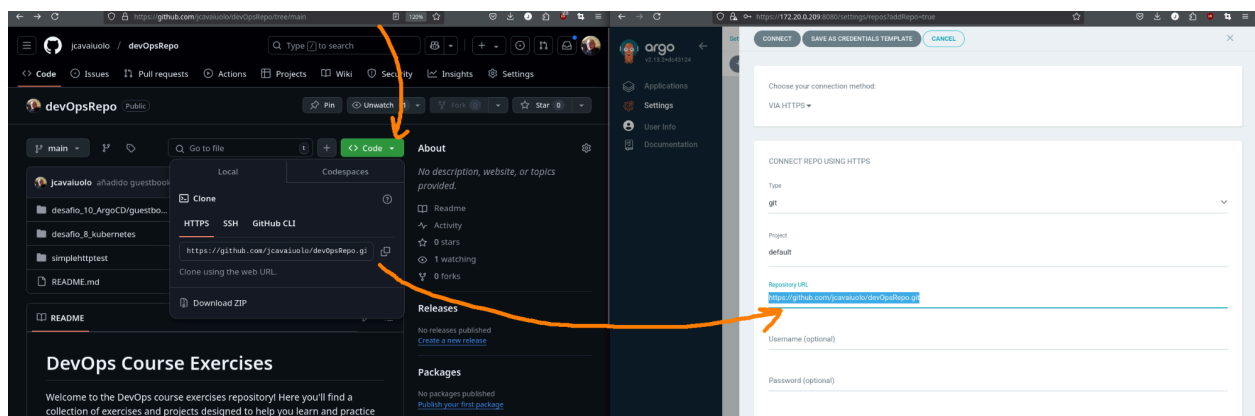


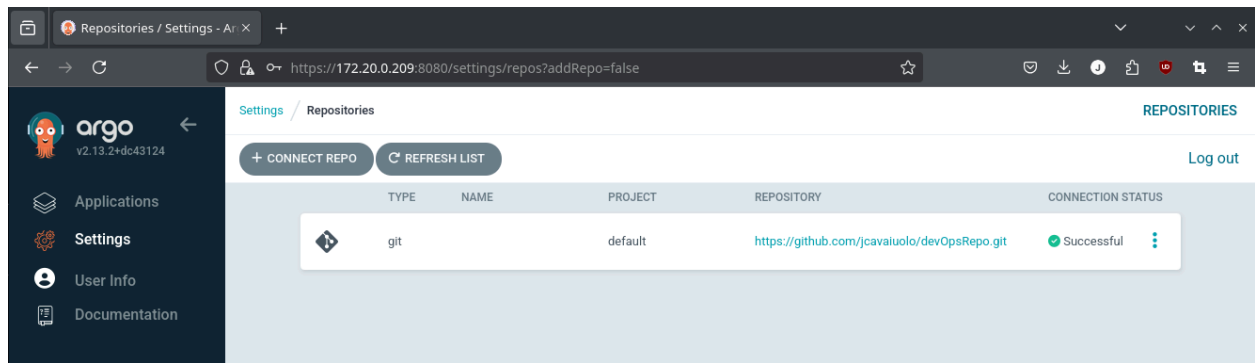
También voy a setear una copia de la app de guestbook de <https://github.com/argoproj/argocd-example-apps> esto debería darme la oportunidad de pushear cambios a github y que esto se refleje en el cluster.

El código residirá en el directorio https://github.com/jcavaiuolo/devOpsRepo/tree/main/desafio_10_ArgoCD/guestbook

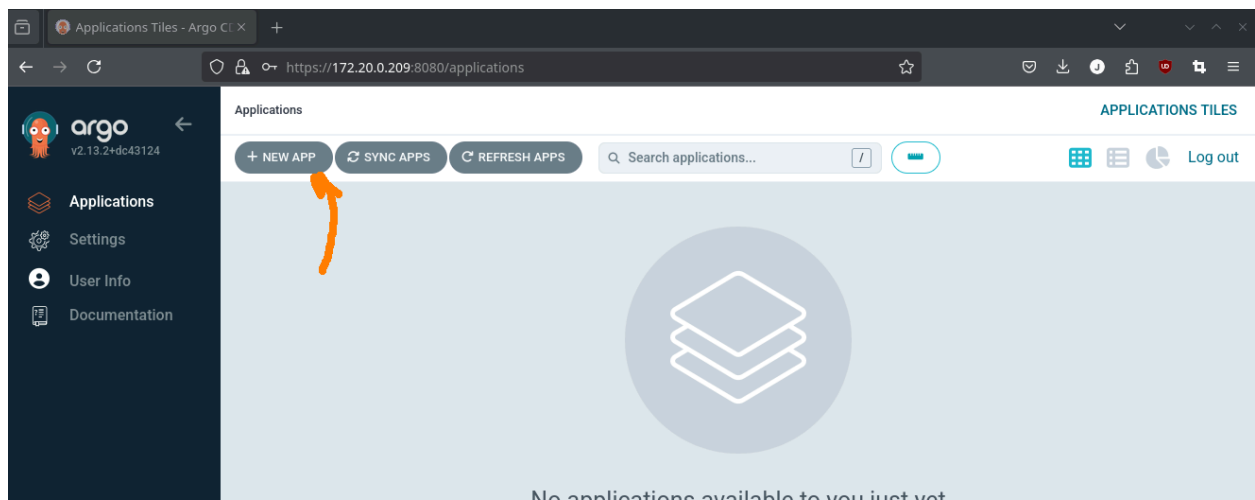
Crear la Aplicación en ArgoCD:

Primero voy a agregar en Settings >> Repositories >> Connect Repo

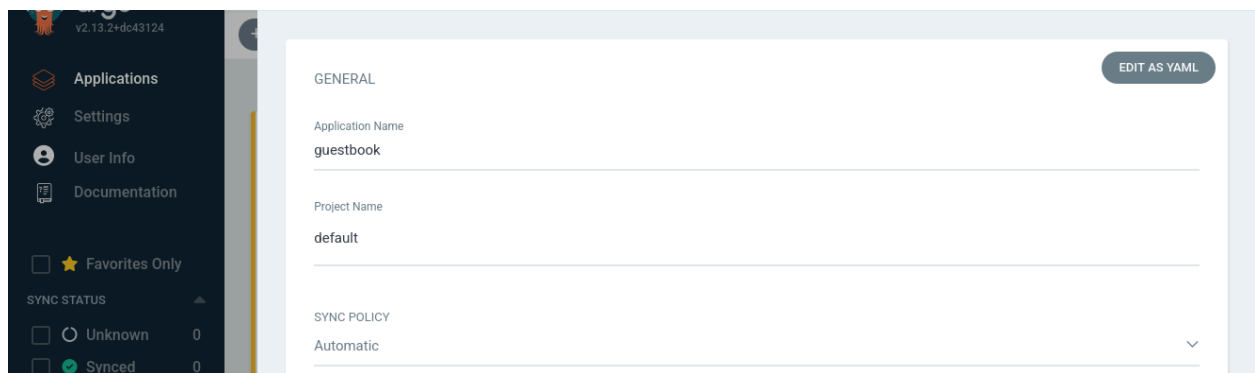




Luego en la sección Applications >> New App



Voy a completar las opciones de la siguiente manera:



HEALTH STATUS

☐ Unknown 0
 ☐ Progressing 0
 ☐ Suspended 0
 ☐ Healthy 0
 ☐ Degraded 0
 ☐ Missing 1

LABELS

LABELS

PROJECTS

PROJECTS

SOURCE

Repository URL

https://github.com/jcavaiuolo/devOpsRepo.git

GIT ✓

Revision

HEAD

Branches ▼

Path

desafio_10_ArgoCD/guestbook

HEALTH STATUS

☐ Unknown 0
 ☐ Progressing 0
 ☐ Suspended 0
 ☐ Healthy 0
 ☐ Degraded 0
 ☐ Missing 1

LABELS

LABELS

DESTINATION

Cluster URL

https://kubernetes.default.svc

URL ▼

Namespace

guestbook

La aplicación ha sido creada pero con algun error:

Applications Tiles - Argo CD

[←](#)
[→](#)
[↺](#)
[↻](#)

https://172.20.0.209:8080/applications?showFavorites=false&proj=&sync=&autoSync=&health=&...

argo

v2.13.2+dc43124

Applications

Settings

User Info

Documentation

☐ ★ Favorites Only

SYNC STATUS

☐ Unknown 0
 ☐ Synced 0
 ☐ OutOfSync 1

HEALTH STATUS

☐ Unknown 0
 ☐ Progressing 0
 ☐ Suspended 0
 ☐ Healthy 0
 ☐ Degraded 0
 ☐ Missing 1

Applications

+ NEW APP

↻ SYNC APPS

↻ REFRESH APPS

🔍 Search applications...

📄

🔴

📊

📅

👤 Log out

Sort: name ▼ Items per page: 10 ▼

guestbook

☆

Project: default

Labels:

Status: Missing OutOfSync Syncing

Repository: https://github.com/jcavaiuolo/devOpsRepo.git

Target Revisi... HEAD

Path: desafio_10_ArgoCD/guestbook

Destination: in-cluster

Namespace: guestbook

Created At: 01/30/2025 22:31:52 (a few seconds ago)

Last Sync: 01/30/2025 22:32:23 (a few seconds ago)

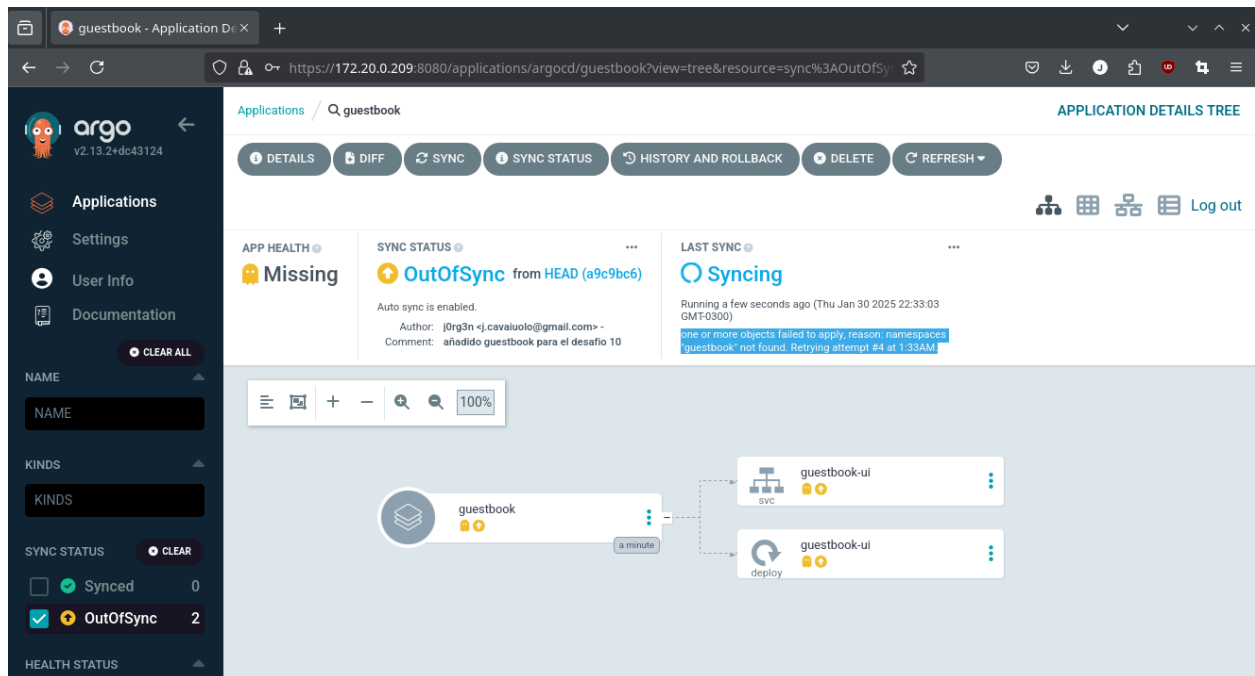
↻ SYNC

↻

🔴

Troubleshooting:

desde dentro de la aplicacion puedo ver el error: "one or more objects failed to apply, reason: namespaces "guestbook" not found. Retrying attempt #4 at 1:33AM."



no hay en el manifiesto de la aplicación la instrucción para crear el namespace. Así que voy a agregarlo y hacer el push al repositorio:

```
j0rg3n@teletraan:~/Desktop/devopsrepo/desafio_10_ArgoCD/guestbook$ nano namespace.yaml
j0rg3n@teletraan:~/Desktop/devopsrepo/desafio_10_ArgoCD/guestbook$ cat namespace.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: guestbook
j0rg3n@teletraan:~/Desktop/devopsrepo/desafio_10_ArgoCD/guestbook$ ll
total 12
-rw-r--r--. 1 j0rg3n j0rg3n 389 Jan 30 22:06 guestbook-ui-deployment.yaml
-rw-r--r--. 1 j0rg3n j0rg3n 141 Jan 30 22:06 guestbook-ui-svc.yaml
-rw-r--r--. 1 j0rg3n j0rg3n 59 Jan 30 22:37 namespace.yaml
j0rg3n@teletraan:~/Desktop/devopsrepo/desafio_10_ArgoCD/guestbook$
j0rg3n@teletraan:~/Desktop/devopsrepo$ git add .
j0rg3n@teletraan:~/Desktop/devopsrepo$ git commit -m "añadido namespace al guestbook del desafio10"
[main 49bcab3] añadido namespace al guestbook del desafio10
1 file changed, 4 insertions(+)
create mode 100644 desafio_10_ArgoCD/guestbook/namespace.yaml
j0rg3n@teletraan:~/Desktop/devopsrepo$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 464 bytes | 464.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/jcavaiuolo/devOpsRepo.git
a9c9bc6..49bcab3 main -> main
j0rg3n@teletraan:~/Desktop/devopsrepo$
```

Luego de añadido el manifiesto para el namespace la app se sincronizó automáticamente:


```

ubuntuuser@ubuntu:~$ kubectl get all -n guestbook
NAME                                READY    STATUS    RESTARTS   AGE
pod/guestbook-ui-764d76f89d-zh7rx  1/1      Running   0           4m37s

NAME                                TYPE           CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/guestbook-ui               ClusterIP      10.99.181.80  <none>       80/TCP     4m37s

NAME                                READY    UP-TO-DATE   AVAILABLE   AGE
deployment.apps/guestbook-ui       1/1      1             1           4m37s

NAME                                DESIRED    CURRENT    READY   AGE
replicaset.apps/guestbook-ui-764d76f89d  1          1          1       4m37s
ubuntuuser@ubuntu:~$

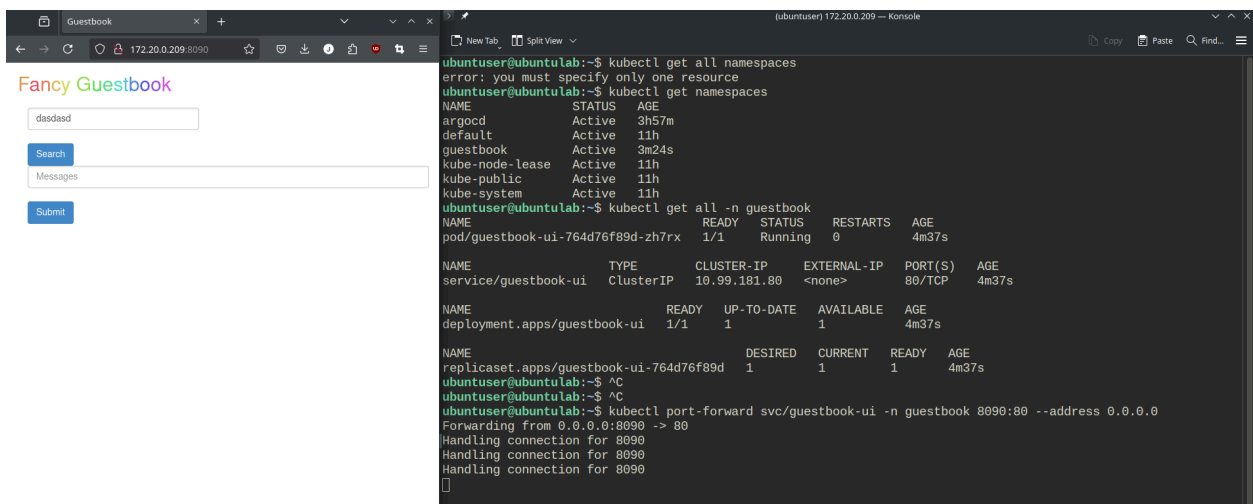
```

Como tipo de servicio es ClusterIP sólo es accesible desde el propio cluster, voy a usar el mismo comando que usé durante el proceso de setup para configurar un port forwarding para este servicio (teniendo cuidado de no pisarme con el puerto de administración de argocd) con el modificador `--address 0.0.0.0`

```

kubectl port-forward svc/guestbook-ui -n guestbook 8090:80 --address 0.0.0.0

```



The screenshot shows a web browser window on the left displaying the 'Fancy Guestbook' application. The application has a search bar with the text 'dasdasd' and a 'Search' button. Below the search bar is a 'Messages' section. On the right, a terminal window shows the following commands and output:

```

ubuntuuser@ubuntu:~$ kubectl get all namespaces
error: you must specify only one resource
ubuntuuser@ubuntu:~$ kubectl get namespaces
NAME              STATUS    AGE
argocd            Active    3h57m
default           Active    11h
guestbook         Active    3m24s
kube-node-lease   Active    11h
kube-public       Active    11h
kube-system       Active    11h
ubuntuuser@ubuntu:~$ kubectl get all -n guestbook
NAME                                READY    STATUS    RESTARTS   AGE
pod/guestbook-ui-764d76f89d-zh7rx  1/1      Running   0           4m37s

NAME                                TYPE           CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/guestbook-ui               ClusterIP      10.99.181.80  <none>       80/TCP     4m37s

NAME                                READY    UP-TO-DATE   AVAILABLE   AGE
deployment.apps/guestbook-ui       1/1      1             1           4m37s

NAME                                DESIRED    CURRENT    READY   AGE
replicaset.apps/guestbook-ui-764d76f89d  1          1          1       4m37s
ubuntuuser@ubuntu:~$ ^C
ubuntuuser@ubuntu:~$ ^C
ubuntuuser@ubuntu:~$ kubectl port-forward svc/guestbook-ui -n guestbook 8090:80 --address 0.0.0.0
Forwarding from 0.0.0.0:8090 -> 80
Handling connection for 8090
Handling connection for 8090
Handling connection for 8090

```