Desafío No. 9

Bootcamp DevOps 63703

Presentado por:
Marco Vanegas
2023

Entregables

1. Manifiesto de Kubernetes que se aplicará en el cluster, puede ser el deploy de un pod con una imagen.

2. Documento con instrucciones de cómo se realizó el desafío (de ser posible adjuntar capturas de pantalla).

Consejos

Se recomienda el uso de Argo CD pero se pueden utilizar otras herramientas, así también se recomienda no automatizar el sync de Argo para tener un poco más de control sobre cuando aplicamos los cambios en nuestras aplicaciones.

Illustrations by Pixeltrue on icons8

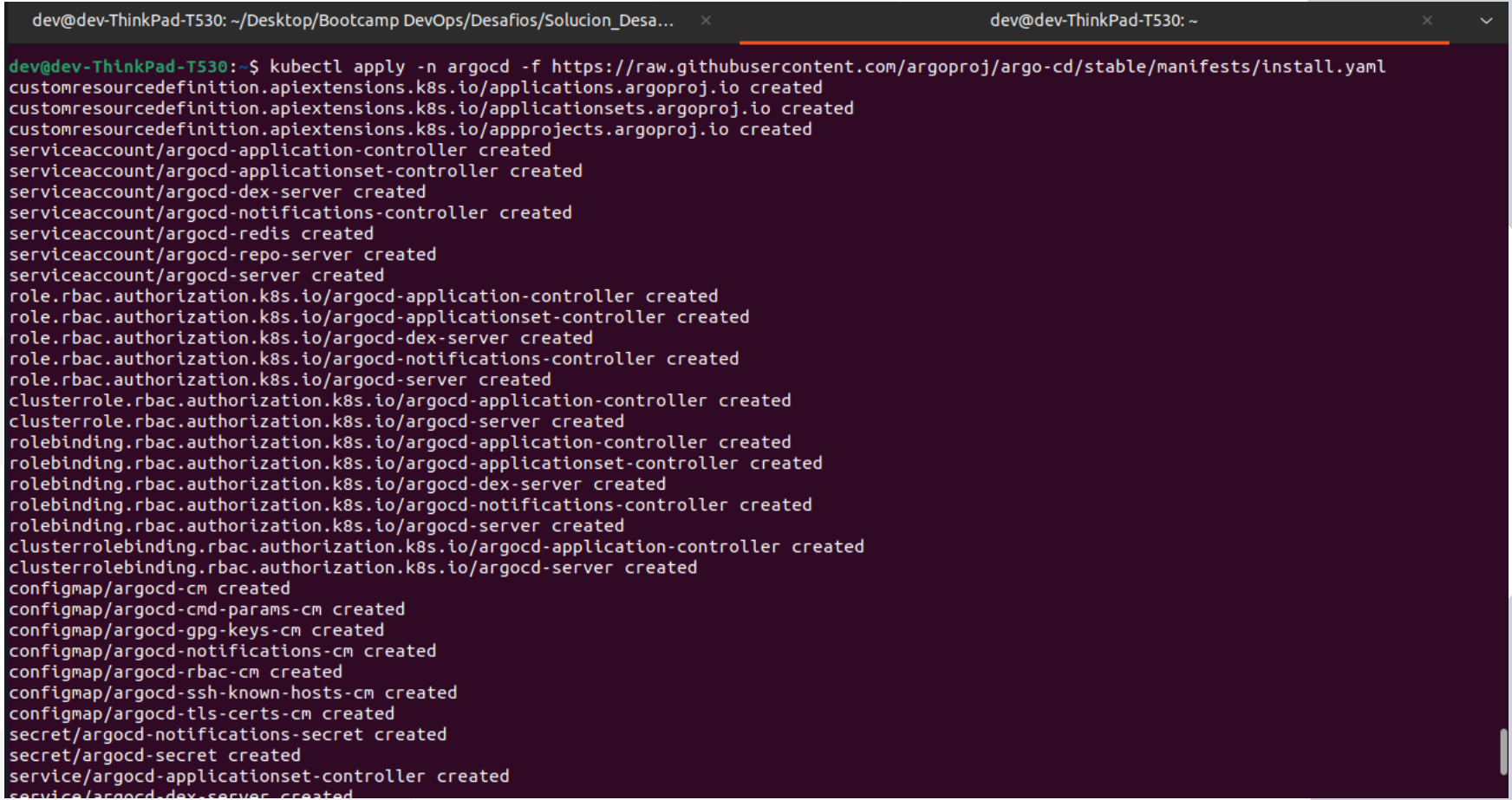
1. Se inicializa el servicio docker-desktop con el comando `systemctl --user start docker-desktop`, e igualmente se inicia sesión con `docker login`. Posteriormente, se inicia minikube `minikube start`.

```
dev@dev-ThinkPad-T530:~/Desktop/Bootcamp DevOps/Desafios/Solucion_Desafios/Solucion_Desafio_9$ systemctl --user status docker-desktop
● docker-desktop.service - Docker Desktop
     Loaded: loaded (/usr/lib/systemd/user/docker-desktop.service; disabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-11-03 08:17:22 -05; 2s ago
   Main PID: 397237 (com.docker.back)
      Tasks: 9 (limit: 9068)
     Memory: 49.7M
        CPU: 252ms
     CGroup: /user.slice/user-1000.slice/user@1000.service/app.slice/docker-desktop.service
             └─397237 /opt/docker-desktop/bin/com.docker.backend

nov 03 08:17:21 dev-ThinkPad-T530 systemd[3381]: Starting Docker Desktop...
nov 03 08:17:22 dev-ThinkPad-T530 systemd[3381]: Started Docker Desktop.
dev@dev-ThinkPad-T530:~/Desktop/Bootcamp DevOps/Desafios/Solucion_Desafios/Solucion_Desafio_9$
```

```
dev@dev-ThinkPad-T530:~$ minikube start
😄  minikube v1.31.2 on Ubuntu 22.04
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.27.4 on Docker 24.0.4 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🔎  Verifying Kubernetes components...
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
dev@dev-ThinkPad-T530:~$
```

2. Se crea el namespace de Argo CD `kubectl create ns argocd` y luego se ejecuta `kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml`.



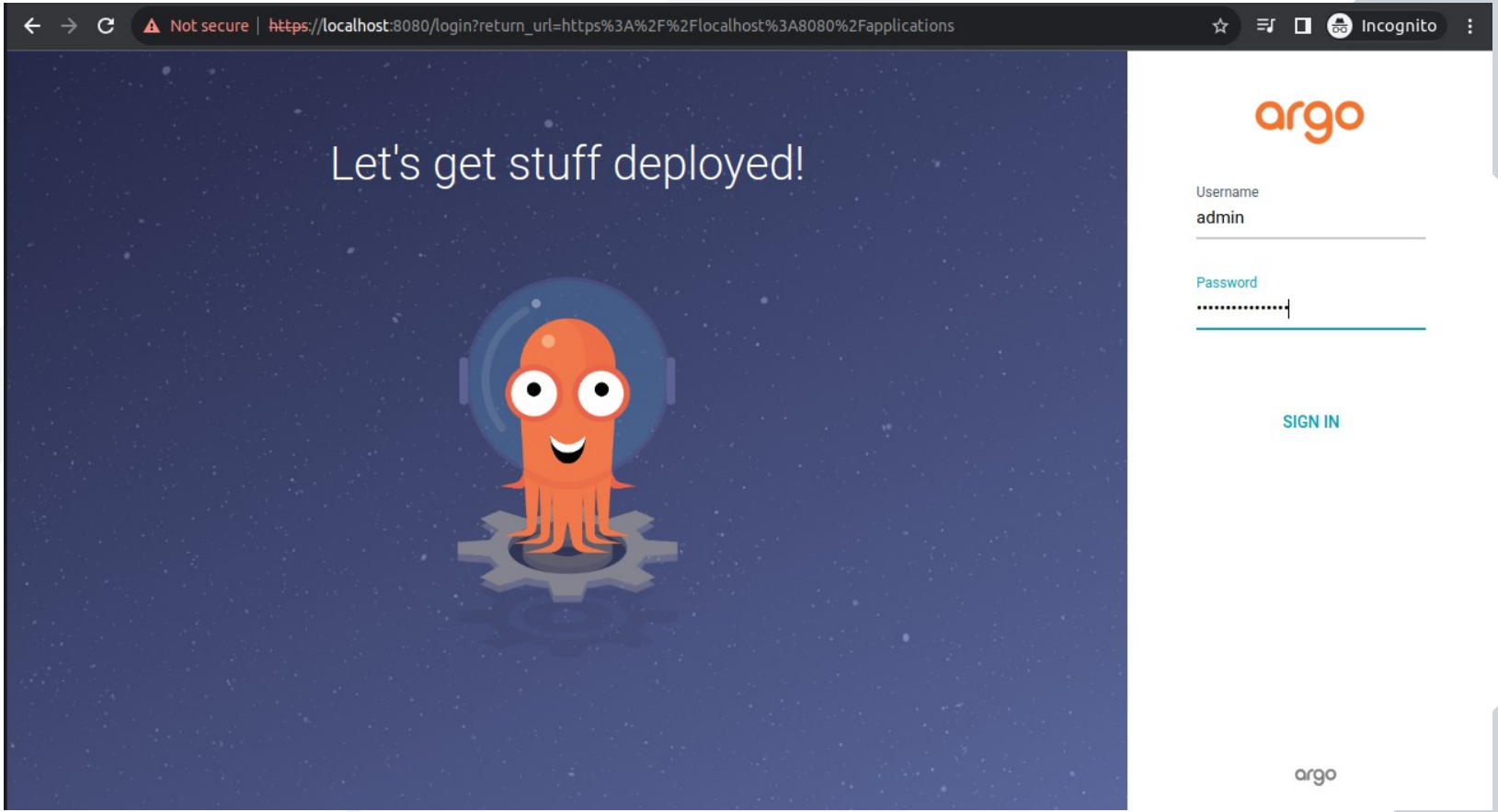
3. Se genera la clave para ingresar a la consola de Argo CD mediante el comando `kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d`.

```
dev@dev-ThinkPad-T530:~$ kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d
reO2IfR5x4iOOjgTdev@dev-ThinkPad-T530:~$
```

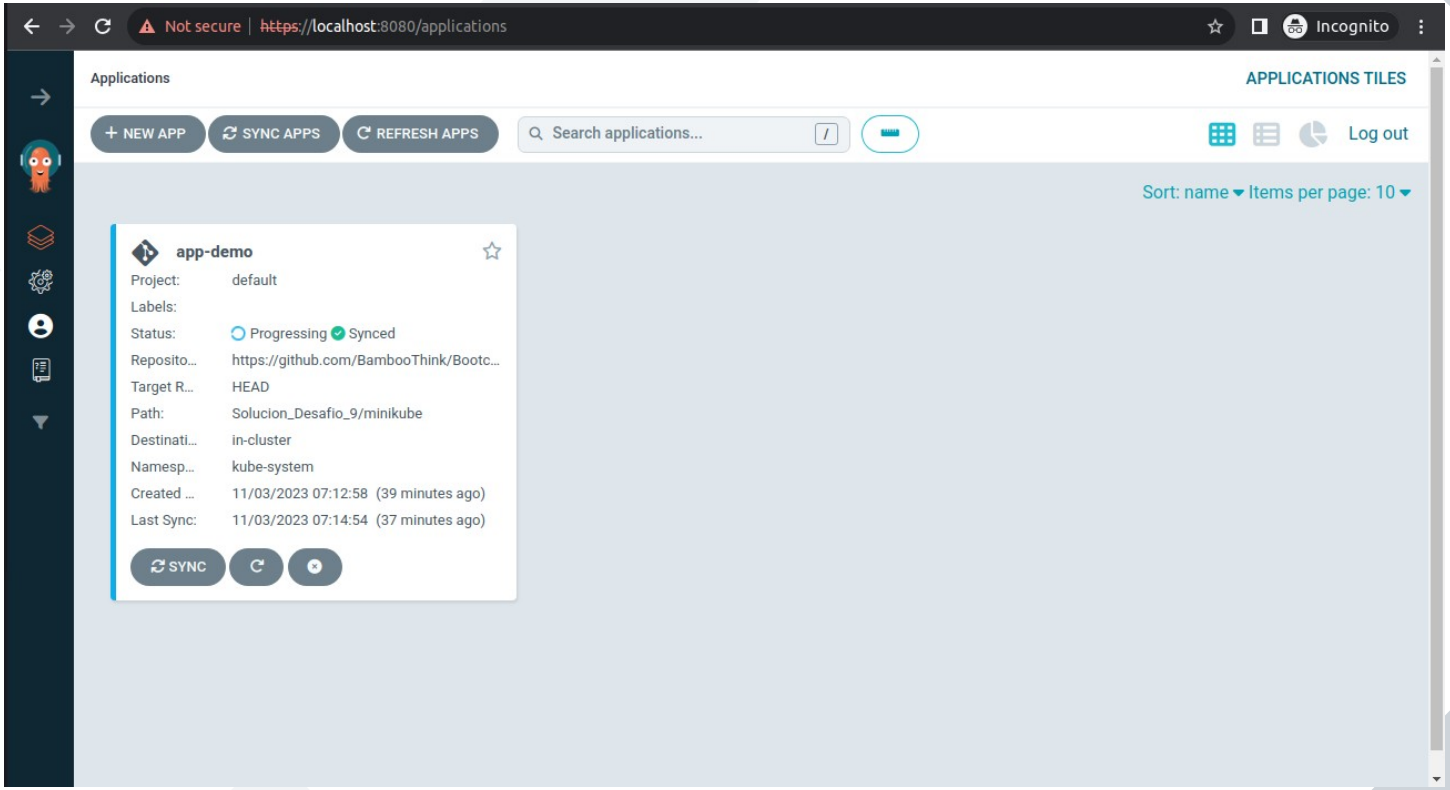
Luego, se ejecuta el comando `kubectl port-forward svc/argocd-server -n argocd 8080:443` que permite acceder a la consola de Argo CD mediante el navegador.

```
dev@dev-ThinkPad-T530:~$ kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
```

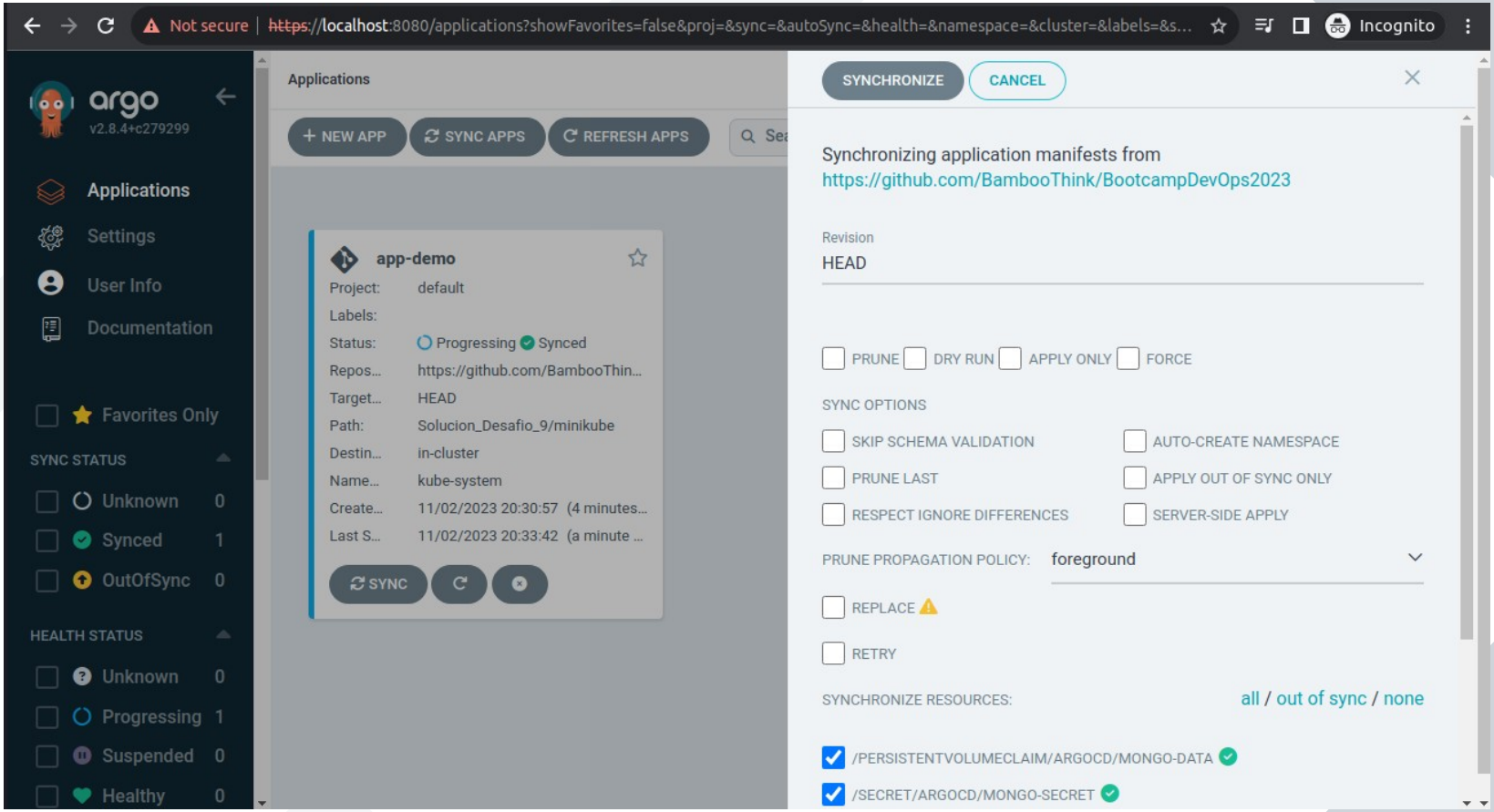
4. Se ingresa a la consola de Argo CD con el usuario `admin` y la contraseña generada en el paso 4.



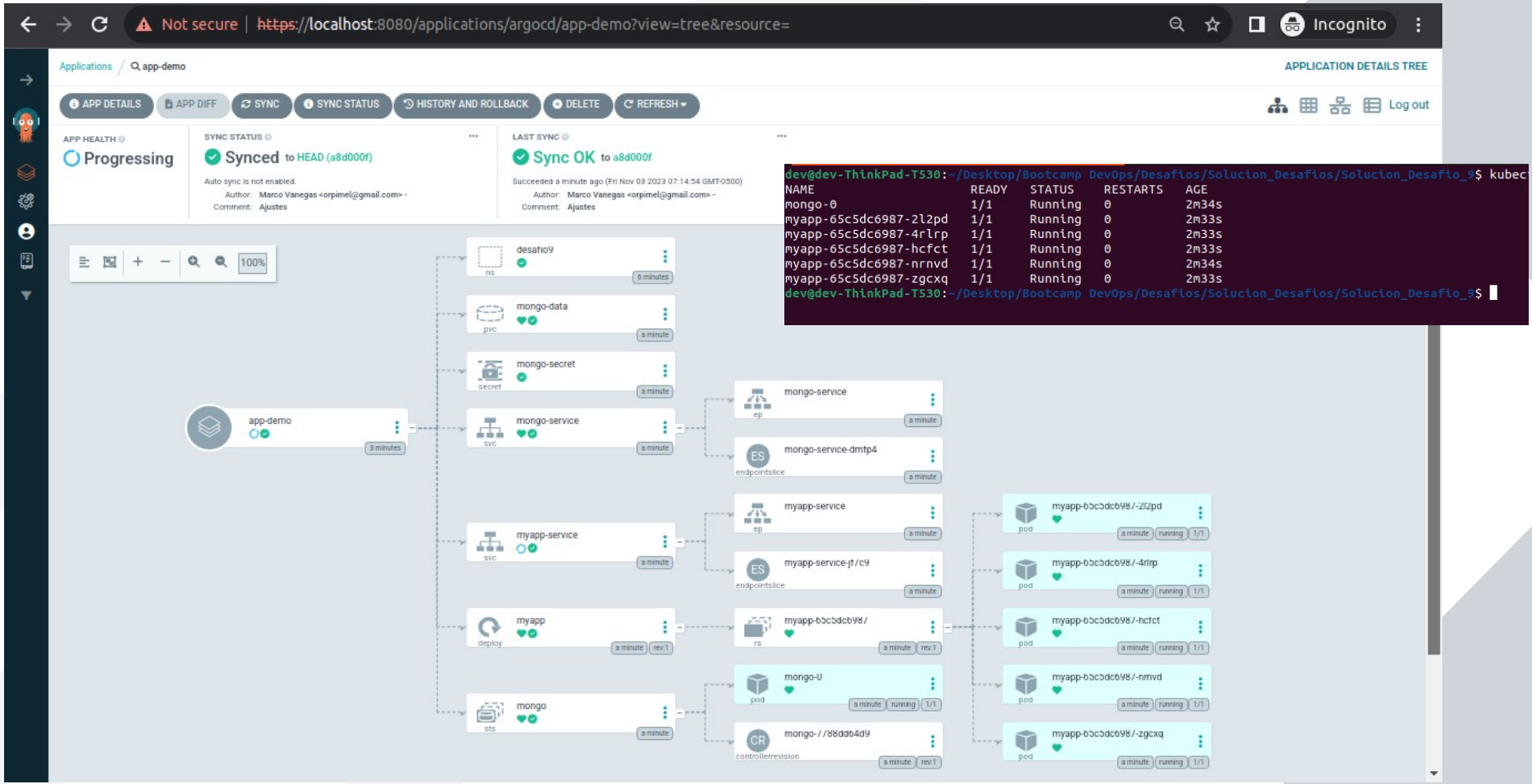
5. Se ejecuta `kubectl apply -f my-app.yaml` para crear la app en Argo CD. Este manifiesto, contiene el namespace argocd para que el objeto pueda ser visualizado en la consola de Argo CD. A continuación, se valida su creación mediante el comando `kubectl get apps -n argocd`, y la visualización del proceso en la consola de Argo CD.



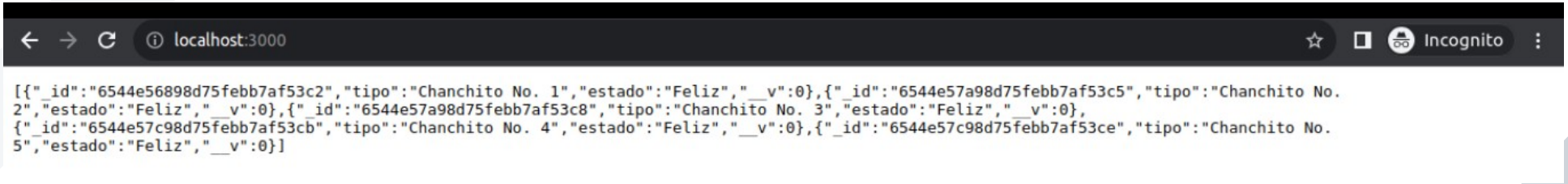
6. Se dá clic en SYNC y luego en SYNCHRONIZE en el modal para ejecutar los manifiestos del repositorio de GitHub relacionado en my-app.yaml del paso 7.



7. Se valida la creación de los pods mediante el comando `kubectl get pods -n desafio9`, e igualmente se valida en la consola de Argo CD.



8. Se ejecuta el comando `kubectl port-forward svc/myapp-service -n desafio9 3000:3000` para acceder a la app desde el navegador, y posteriormente se valida el funcionamiento de la aplicación, para ello se ingresa a http://localhost:3000/crear, se recarga la página 5 veces (F5) y luego se valida la creación de los registros en la url http://localhost:3000.



Repositorio de GitHub donde se encuentra el código, junto con los pasos en detalle:

https://github.com/BambooThink/BootcampDevOps2023/tree/main/Solucion_Desafio_9