



Desafío No. 9

Bootcamp DevOps 63703

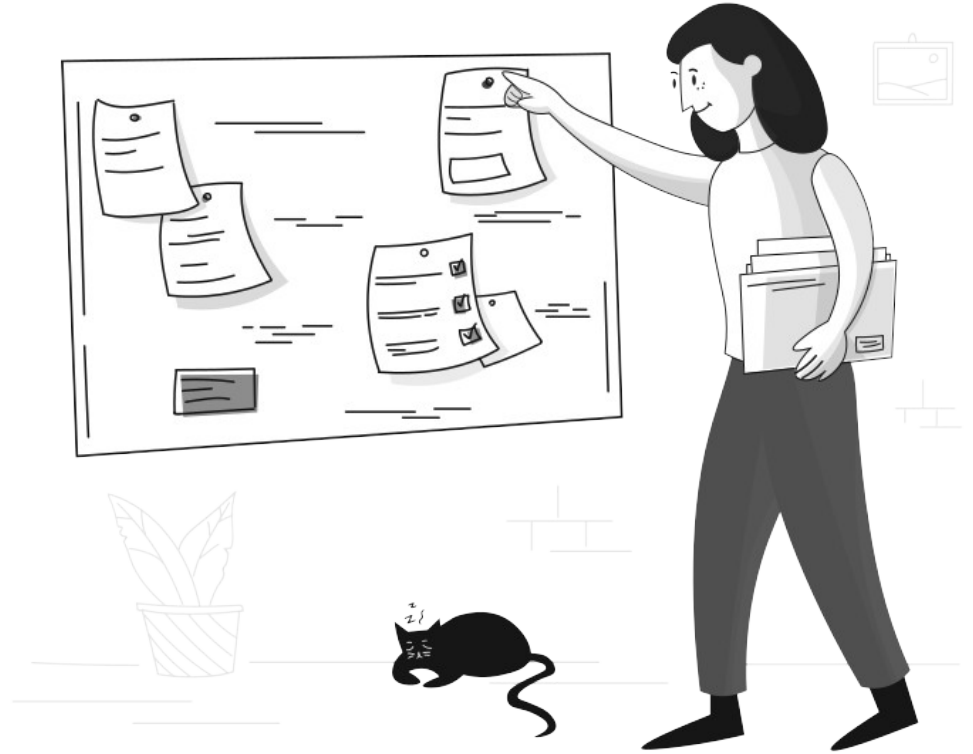
Presentado por:
Marco Vanegas
2023

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>`.

```
dev@dev-ThinkPad-T530: ~/Desktop/Bootcamp DevOps/Desafios/Solucion_Desa... x dev@dev-ThinkPad-T530: ~
dev@dev-ThinkPad-T530:~$ kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/applicationsets.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/appprojects.argoproj.io created
serviceaccount/argocd-application-controller created
serviceaccount/argocd-applicationset-controller created
serviceaccount/argocd-dex-server created
serviceaccount/argocd-notifications-controller created
serviceaccount/argocd-redis created
serviceaccount/argocd-repo-server created
serviceaccount/argocd-server created
role.rbac.authorization.k8s.io/argocd-application-controller created
role.rbac.authorization.k8s.io/argocd-applicationset-controller created
role.rbac.authorization.k8s.io/argocd-dex-server created
role.rbac.authorization.k8s.io/argocd-notifications-controller created
role.rbac.authorization.k8s.io/argocd-server created
clusterrole.rbac.authorization.k8s.io/argocd-application-controller created
clusterrole.rbac.authorization.k8s.io/argocd-server created
rolebinding.rbac.authorization.k8s.io/argocd-application-controller created
rolebinding.rbac.authorization.k8s.io/argocd-applicationset-controller created
rolebinding.rbac.authorization.k8s.io/argocd-dex-server created
rolebinding.rbac.authorization.k8s.io/argocd-notifications-controller created
rolebinding.rbac.authorization.k8s.io/argocd-server created
clusterrolebinding.rbac.authorization.k8s.io/argocd-application-controller created
clusterrolebinding.rbac.authorization.k8s.io/argocd-server created
configmap/argocd-cm created
configmap/argocd-cmd-params-cm created
configmap/argocd-gpg-keys-cm created
configmap/argocd-notifications-cm created
configmap/argocd-rbac-cm created
configmap/argocd-ssh-known-hosts-cm created
configmap/argocd-tls-certs-cm created
secret/argocd-notifications-secret created
secret/argocd-secret created
service/argocd-applicationset-controller created
service/argocd-dex-server created
```

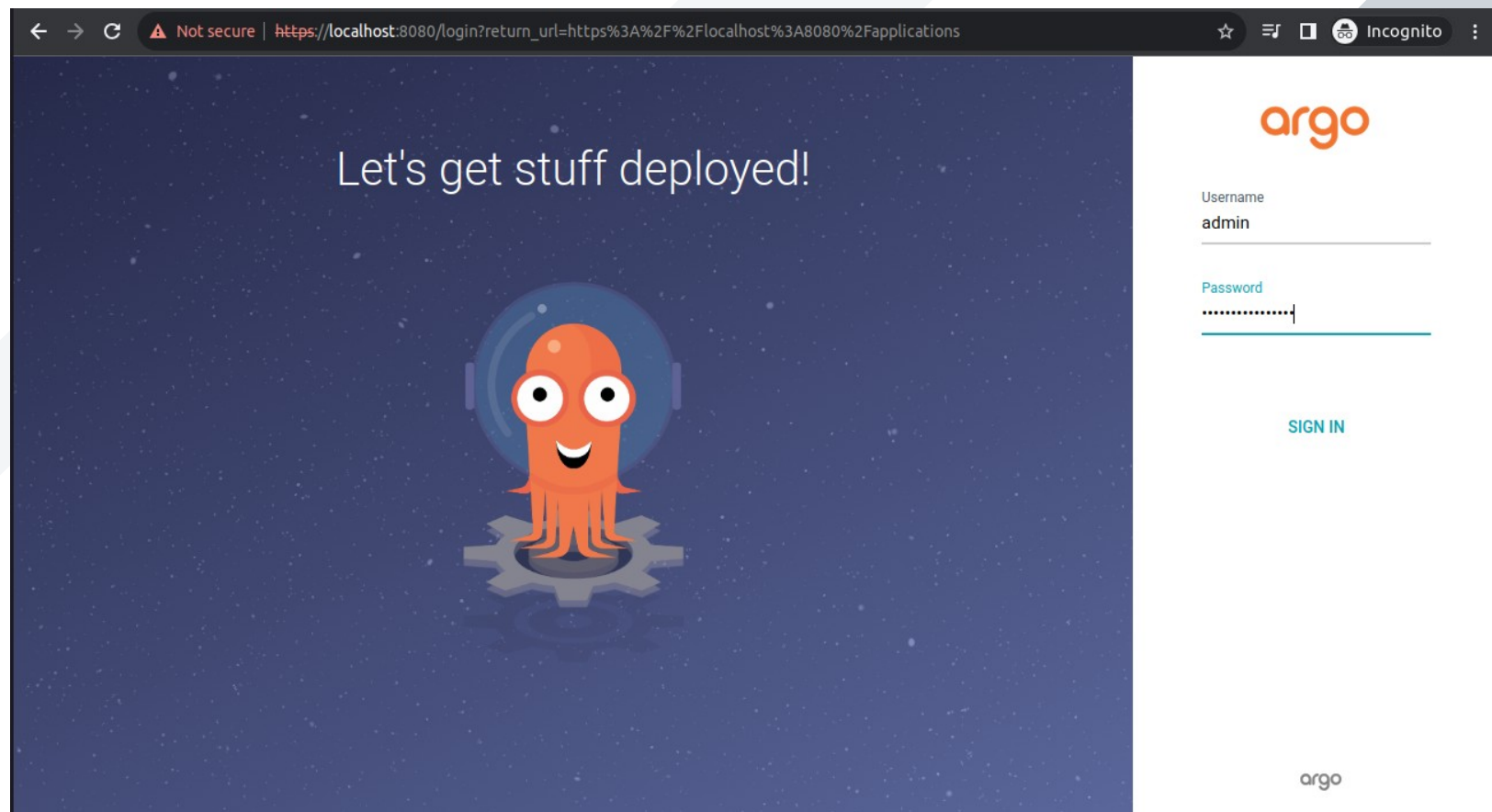
3. Se genera la clave para ingresar a la consola de Argo CD mediante el comando ``kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d``.

```
dev@dev-ThinkPad-T530:~$ kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d  
re02IfR5x4i00jgTdev@dev-ThinkPad-T530:~$
```

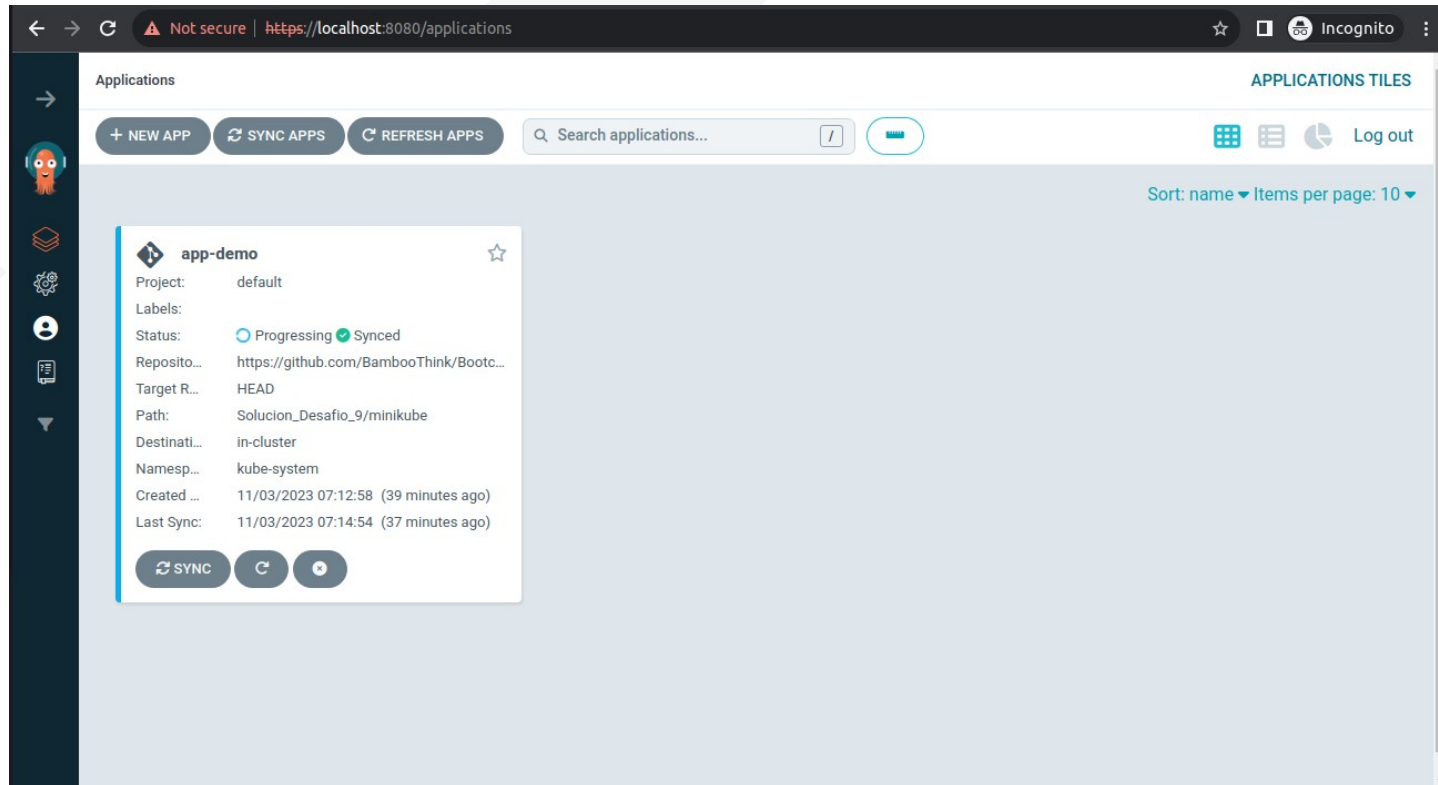
Luego, se ejecuta el comando ``kubectl port-forward svc/argocd-server -n argocd 8080:443`` que permite acceder a la consola de Argo CD mediante el navegador.

```
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.

The screenshot displays the Argo CD web interface in a browser window. The address bar shows the URL `https://localhost:8080/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=&s...`. The interface includes a sidebar with navigation links: Applications, Settings, User Info, and Documentation. Below these are filters for 'Favorites Only' and 'SYNC STATUS' (Unknown, Synced, OutOfSync) and 'HEALTH STATUS' (Unknown, Progressing, Suspended, Healthy).

The main content area shows the 'Applications' list with a table of application details for 'app-demo':

Project	Labels	Status	Repos...	Target...	Path	Destin...	Name...	Create...	Last S...
default		Progressing ✓ Synced	https://github.com/BambooThink/...	HEAD	Solucion_Desafio_9/minikube	in-cluster	kube-system	11/02/2023 20:30:57 (4 minutes...)	11/02/2023 20:33:42 (a minute ...)

Below the table are buttons for 'SYNC', 'REFRESH', and 'DELETE'.

A 'SYNCHRONIZE' modal is open on the right, titled 'Synchronizing application manifests from <https://github.com/BambooThink/BootcampDevOps2023>'. The modal contains the following options:

- Revision: HEAD
- ☐ PRUNE ☐ DRY RUN ☐ APPLY ONLY ☐ FORCE
- SYNC OPTIONS:
 - ☐ SKIP SCHEMA VALIDATION ☐ AUTO-CREATE NAMESPACE
 - ☐ PRUNE LAST ☐ APPLY OUT OF SYNC ONLY
 - ☐ RESPECT IGNORE DIFFERENCES ☐ SERVER-SIDE APPLY
- PRUNE PROPAGATION POLICY: foreground
- ☐ REPLACE ⚠
- ☐ RETRY
- SYNCHRONIZE RESOURCES: [all](#) / [out of sync](#) / [none](#)
- ☒ /PERSISTENTVOLUMECLAIM/ARGOCD/MONGO-DATA ✓
- ☒ /SECRET/ARGOCD/MONGO-SECRET ✓

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.

← → ↻ ⚠ Not secure | <https://localhost:8080/applications/argocd/app-demo?view=tree&resource=> 🔍 ☆ 🏠 Incognito ⋮

Applications / [app-demo](#) APPLICATION DETAILS TREE

APP DETAILS APP DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH **Progressing**

SYNC STATUS **Synced to HEAD (a8d000f)**

Auto sync is not enabled.
Author: Marco Vanegas <orimel@gmail.com>
Comment: Ajustes

LAST SYNC **Sync OK to a8d000f**

Succeeded a minute ago (Fri Nov 03 2023 07:14:54 GMT+0300)
Author: Marco Vanegas <orimel@gmail.com>
Comment: Ajustes

desafio9
rs 0 minutes

mongo-data
pvc 1 minute

mongo-secret
secret 1 minute

mongo-service
svc 1 minute

myapp-service
svc 1 minute

myapp
deploy 1 minute rev1

mongo
sts 1 minute

mongo-service
ep 1 minute

mongo-service-dmtp4
endpointslice 1 minute

myapp-service
ep 1 minute

myapp-service-ft/c9
endpointslice 1 minute

myapp-bcbdc098/
rs 1 minute rev1

mongo-U
pod 1 minute running 1/1

mongo- /88d0b4d9
controllerrevision 1 minute rev1

myapp-bcbdc098/-Zl2pd
pod 1 minute running 1/1

myapp-bcbdc098/-4rlrp
pod 1 minute running 1/1

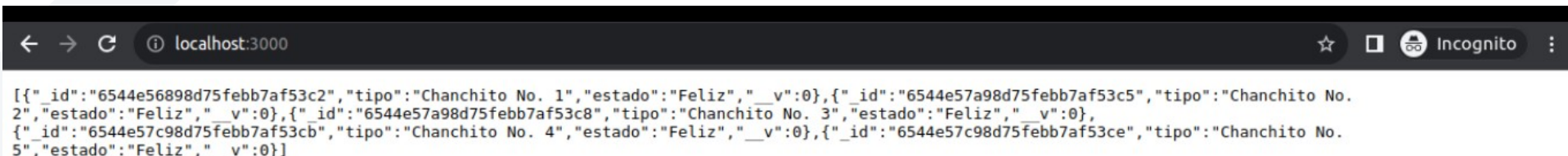
myapp-bcbdc098/-hctct
pod 1 minute running 1/1

myapp-bcbdc098/-nmvd
pod 1 minute running 1/1

myapp-bcbdc098/-zgxcq
pod 1 minute running 1/1

```
dev@dev-ThinkPad-T530:~/Desktop/Bootcamp DevOps/Desafios/Solucion_Desafios/Solucion_Desafio_9$ kubectl get pods -n desafio9
NAME                                READY   STATUS    RESTARTS   AGE
mongo-0                              1/1     Running   0           2m34s
myapp-65c5dc6987-2l2pd               1/1     Running   0           2m33s
myapp-65c5dc6987-4rlrp               1/1     Running   0           2m33s
myapp-65c5dc6987-hcfct               1/1     Running   0           2m33s
myapp-65c5dc6987-nrnvd               1/1     Running   0           2m34s
myapp-65c5dc6987-zgxcq               1/1     Running   0           2m33s
dev@dev-ThinkPad-T530:~/Desktop/Bootcamp DevOps/Desafios/Solucion_Desafios/Solucion_Desafio_9$
```

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