



# MANUAL DE USO DE OPERADOR

<b>Release Notes</b>	<b>1</b>
<b>Compatibilidad de versiones</b>	<b>1</b>
<b>Upgrade operador a versión 4.04</b>	<b>1</b>
<b>Descargar operador/controller</b>	<b>3</b>
<b>Despliegue del operador/controller</b>	<b>5</b>
Ajustar repositorio Docker, imagen y versión	5
Despliegue del operador de Anjana	6
Credenciales del repositorio Docker	7
Despliegue CRD de configuraciones	7
Despliegue CRD de los microservicios de Anjana	8
Despliegue CRD de los plugins de Anjana	10
Ajustar en Edusa el origen de la configuración	12
• Perfil native y carpeta local	12
• Perfil default y repositorio Git	13

Versión	Fecha de publicación	Responsable	Aprobador	Resumen de cambios
1.0	28/10/2022	Dpto DS	Responsable DS	Creación del documento
2.0	15/12/2023	Dpto DS	Responsable DS	Nueva URL repositorio Docker dr-releases.anjanadata.org

## Release Notes

- Se ha implementado el despliegue de los archivos de configuración de los microservicios en Edusa a través de configmaps.
- Se ha implementado la posibilidad de despliegue de los plugins a través del operador de Anjana.
- Se ha cambiado la configuración para que se adecue a la versión 4.4

## Compatibilidad de versiones

La presente versión del operador puede ser usada para versiones de Anjana Data:

- 4.4

**NOTA:** En la presente versión del kit se incluyen preconfiguradas la última versión de bugfix de cada elemento tratado en el momento de la publicación, pudiendo ser publicadas versiones independientes de dichos elementos en fechas posteriores a la publicación del presente kit. Recuerde revisar y ajustar las versiones de los elementos a desplegar a la última versión de bugfix disponible.

## Upgrade operador a versión 4.04

Para hacer upgrade de una versión anterior a la actual versión hay que tener en cuenta los siguientes puntos:

- En versiones anteriores los .yaml de configuración estaban en un volumen acoplado al pod de Edusa. Hay que copiar esos .yaml de configuración a cada CRD de configuración de cada microservicio que tiene la versión actual.

```

  anjana-operator
  > cicdcq
  > deploy_for_clients
  > deploy_crds
  > deploy_crds_config
  ! anjana_v1_configdritttesta.yaml
  ! anjana_v1_confighecate.yaml
  ! anjana_v1_confighermes.yaml
  ! anjana_v1_configkerno.yaml
  ! anjana_v1_configminerva.yaml
  ! anjana_v1_configportuno.yaml
  ! anjana_v1_configtot.yaml
  ! anjana_v1_configtpawsglue.yaml
  ! anjana_v1_configtpawsglue.yaml

You, 2 weeks ago | 2 authors (You and others)
1  apiVersion: anjana.my.domain/v1
2  kind: Configdritttesta
3  metadata:
4    name: configdritttesta-sample
5    namespace: anjana-system
6  spec:
7    dritttesta_config: |
8      spring:
9        datasource:
10       url: jdbc:postgresql://rdbservice:5432/anjana?currentSchema=zeus
11       username: anjana
12       password: anjana
13     jpa:
14       properties:
15         hibernate:

```

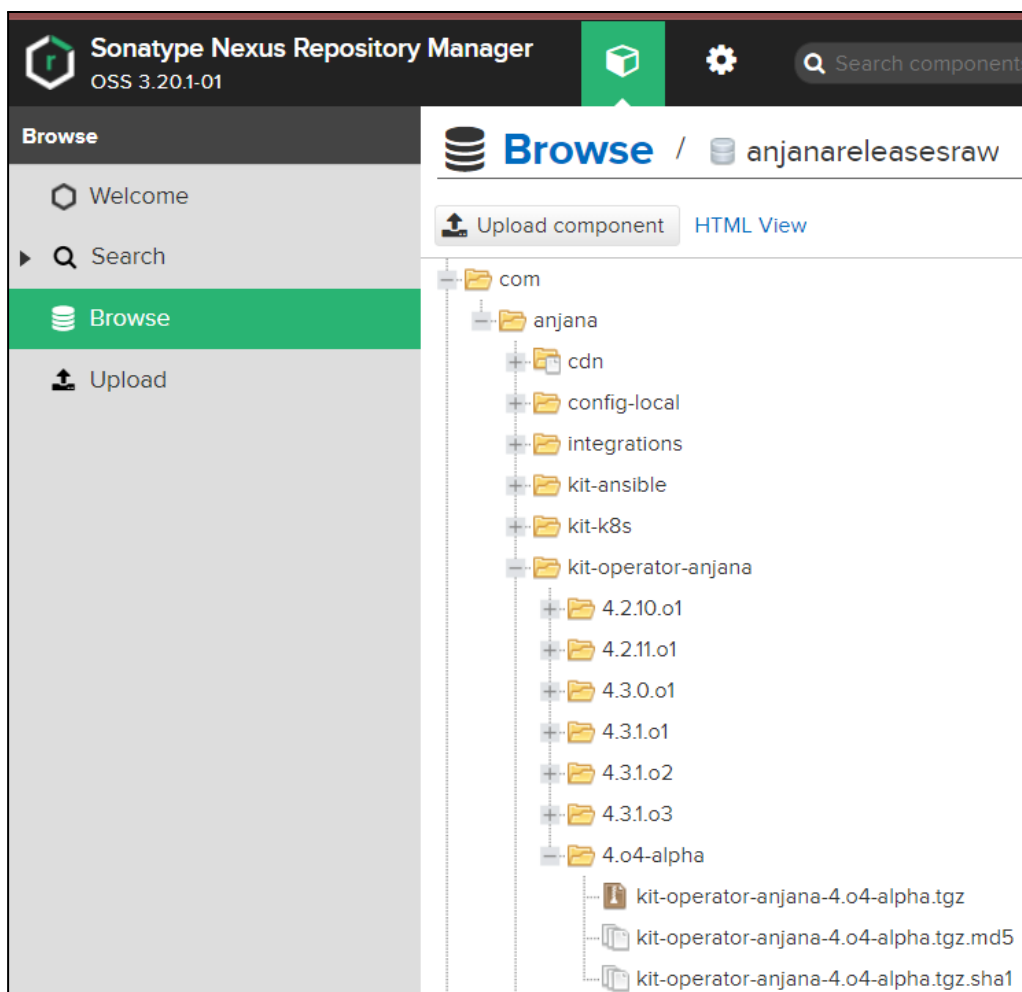
- A continuación, ya se puede repetir el proceso como si fuera la primera vez ya que irá sobrescribiendo el viejo operador por el nuevo. La configuración se desplegará como configmaps y volúmenes en Edusa.

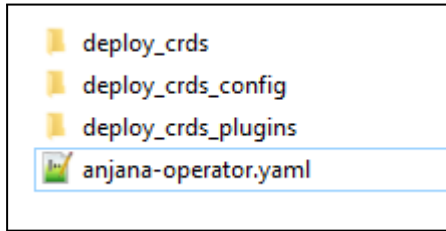
# Descargar operador/controller

El kit se publica en el repositorio raw "releasesraw" de Anjana.

Contenido del kit:

- Contiene un recurso yaml de kubernetes que despliega el operador de Anjana
- Carpeta de despliegues de configmaps con la configuración de cada microservicio
- Carpeta de despliegues de los CRD de cada microservicio de Anjana
- Carpeta de despliegues de los CRD de cada plugin de Anjana





# Despliegue del operador/controller

Los pasos que vamos a seguir para el despliegue de Anjana con el operador son:

1. Ajustar repositorio Docker, imagen y versión
2. Crear secret con credenciales del repositorio Docker
3. Despliegue del recurso yaml del operador de Anjana
4. Despliegue de los configmaps con la configuración de los microservicios
5. Despliegue de los CRD de los microservicios de Anjana
6. Ajustar en Edusa el origen de la configuración de los microservicios.

Disponible el origen en dos formatos:

- Ficheros con la configuración en local
- Repositorio git y generación de un secret de credenciales

## Ajustar repositorio Docker, imagen y versión

Se puede usar un repositorio de Docker propio, distinto al de Anjana, así como la imagen y la versión a desplegar.

Esto debemos de ajustarlo en los recursos yaml que hay en la carpeta `deploy_crds` y en yaml del operador.

NOTA: ajustar el nombre del secret con las credenciales al repositorio Docker en todos los yaml.

NOTA: el argumento `size`: especifica el número de pods de cada statefulSet

```
apiVersion: anjana.my.domain/v1
kind: Edusa
metadata:
  name: edusa-sample
  namespace: anjana-system
spec:
  size: 1
  docker_repository_url: releasesdr.anjanadata.org
  docker_repository_port: 11000
  docker_image: edusa
  docker_version: 4.4.0
  secret_credentials_docker_repository:
    - name: anjanadr
```

```
image: releasesdr.anjanadata.org:11000/anjana-controller:4.04
imagePullPolicy: Always
livenessProbe:
  httpGet:
    path: /healthz
    port: 6789
    initialDelaySeconds: 15
    periodSeconds: 20
name: manager
readinessProbe:
  httpGet:
    path: /readyz
    port: 6789
    initialDelaySeconds: 5
    periodSeconds: 10
securityContext:
  allowPrivilegeEscalation: false
imagePullSecrets:
- name: anjanadr
securityContext:
```

## Despliegue del operador de Anjana

Desplegamos anjana-operator.yaml que creará los recursos necesarios del controller/operador.

```
kubectl apply -f anjana-operator.yaml
```

```
azureuser@k8s:~/kit-operator-anjana-4.x$
azureuser@k8s:~/kit-operator-anjana-4.x$ kubectl apply -f anjana-operator.yaml
namespace/anjana-system created
customresourcedefinition.apiextensions.k8s.io/drittesta.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/edusas.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/grafanas.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/hecatas.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/heimdals.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/hermes.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/horus.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/kernoos.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/minervae.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/portunoos.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/prometheuses.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/tots.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/viators.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/webportunoos.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/webs.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/zeus.anjana.my.domain created
serviceaccount/anjana-controller-manager created
role.rbac.authorization.k8s.io/anjana-leader-election-role created
clusterrole.rbac.authorization.k8s.io/anjana-manager-role created
clusterrole.rbac.authorization.k8s.io/anjana-metrics-reader created
clusterrole.rbac.authorization.k8s.io/anjana-proxy-role created
rolebinding.rbac.authorization.k8s.io/anjana-leader-election-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/anjana-manager-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/anjana-proxy-rolebinding created
configmap/anjana-manager-config created
service/anjana-controller-manager-metrics-service created
deployment.apps/anjana-controller-manager created
azureuser@k8s:~/kit-operator-anjana-4.x$
```

Se crearán los recursos que pueden ver en la siguiente imagen, junto con los crd que luego provisionaremos.

```

azureuser@k8s:~/kit-operator-anjana-4.x$
azureuser@k8s:~/kit-operator-anjana-4.x$ kubectl get all --all-namespaces
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
anjana-system pod/anjana-controller-manager-fb8db9977-kftqm  2/2     Running   0           5m37s
kube-system  pod/calico-kube-controllers-547686d897-ghmwr  1/1     Running   0           16m
kube-system  pod/calico-node-gm6z6                       1/1     Running   0           16m
kube-system  pod/coredns-74ff55c5b-gvrkf                 1/1     Running   0           16m
kube-system  pod/coredns-74ff55c5b-hjgkr                 1/1     Running   0           16m
kube-system  pod/etcd-k8s                                 1/1     Running   0           16m
kube-system  pod/kube-apiserver-k8s                       1/1     Running   0           16m
kube-system  pod/kube-controller-manager-k8s             1/1     Running   0           16m
kube-system  pod/kube-proxy-ckcdc                         1/1     Running   0           16m
kube-system  pod/kube-scheduler-k8s                     1/1     Running   0           16m

NAMESPACE   NAME                                     TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
anjana-system service/anjana-controller-manager-metrics-service  ClusterIP    10.186.184.11 <none>        8443/TCP         5m37s
default     service/kubernetes                               ClusterIP    10.96.0.1     <none>         443/TCP          17m
kube-system service/kube-dns                                 ClusterIP    10.96.0.10    <none>        53/UDP,53/TCP,9153/TCP 17m

NAMESPACE   NAME                                     DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
kube-system daemonset.apps/calico-node                1         1         1       1             1           kubernetes.io/os=linux 16m
kube-system daemonset.apps/kube-proxy                 1         1         1       1             1           kubernetes.io/os=linux 17m

NAMESPACE   NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
anjana-system deployment.apps/anjana-controller-manager  1/1     1             1           5m37s
kube-system  deployment.apps/calico-kube-controllers    1/1     1             1           16m
kube-system  deployment.apps/coredns                    2/2     2             2           17m

NAMESPACE   NAME                                     DESIRED   CURRENT   READY   AGE
anjana-system replicaset.apps/anjana-controller-manager-fb8db9977  1         1         1       5m37s
kube-system  replicaset.apps/calico-kube-controllers-547686d897  1         1         1       16m
kube-system  replicaset.apps/coredns-74ff55c5b                2         2         2       16m
azureuser@k8s:~/kit-operator-anjana-4.x$

```

## Credenciales del repositorio Docker

Se necesita un secret del tipo docker-registry para poder hacer pull del contenedor de Docker.

```

kubectl create secret docker-registry <name_secret> --docker-server=dr-releases.anjanadata.org
--docker-username=<user> --docker-password=<password> --docker-email=<mail> --namespace
anjana-system

```

```

azureuser@k8s:~/kit-operator-anjana-4.x$
azureuser@k8s:~/kit-operator-anjana-4.x$ kubectl --namespace anjana-system create secret docker-registry anjanadr --docker-server=releasesdr.anjanadata.org:11000
--docker-username= --docker-password= --docker-email=
secret/anjanadr created
azureuser@k8s:~/kit-operator-anjana-4.x$

```

**NOTA:** Aunque no se utilice hay que crear un secret como el ejemplo para que no falle Edusa

```

kubectl create secret generic privatekey-configserver -n anjana-system
--from-literal=PRIVATEKEY-CONFIGSERVER='-----BEGIN RSA PRIVATE KEY-----
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX=
-----END RSA PRIVATE KEY-----'

```

## Despliegue CRD de configuraciones

A continuación, con el controller desplegado podemos ejecutar los recursos que levantan los configmaps con las configuraciones para los microservicios de Anjana.

```

# Desplegar todos de golpe
kubectl apply -f deploy_crds_config -n anjana-system

# Desplegar uno a uno
kubectl apply -f deploy_crds_config/anjana_v1_configdritttesta.yaml -n anjana-system

```



**NOTA:** Hay que tener en cuenta que Edusa no funcionará correctamente si se despliega antes de haber desplegado todas las configuraciones de esta manera. En caso de haberlo hecho hay que tirar el pod de Edusa para que se vuelva a levantar y sea capaz de ver las configuraciones que ya hay desplegadas.

## Despliegue CRD de los microservicios de Anjana

Tras esto, podemos ejecutar los recursos que levantan los CRD de los microservicios de Anjana.

```
# Desplegar todos de golpe
kubectl apply -f deploy_crds -n anjana-system

# Desplegar uno a uno
kubectl apply -f deploy_crds/anjana_v1_edusa.yaml -n anjana-system
```

```
azureuser@k8s:~/kit-operator-anjana-4.x$
azureuser@k8s:~/kit-operator-anjana-4.x$ kubectl apply -f deploy_crds
drittesta.anjana.my.domain/drittesta-sample created
edusa.anjana.my.domain/edusa-sample created
grafana.anjana.my.domain/grafana-sample created
hecate.anjana.my.domain/hecate-sample created
heimdal.anjana.my.domain/heimdal-sample created
hermes.anjana.my.domain/hermes-sample created
horus.anjana.my.domain/horus-sample created
kerno.anjana.my.domain/kerno-sample created
minerva.anjana.my.domain/minerva-sample created
portuno.anjana.my.domain/portuno-sample created
prometheus.anjana.my.domain/prometheus-sample created
tot.anjana.my.domain/tot-sample created
viator.anjana.my.domain/viator-sample created
web.anjana.my.domain/web-sample created
webportuno.anjana.my.domain/webportuno-sample created
zeus.anjana.my.domain/zeus-sample created
azureuser@k8s:~/kit-operator-anjana-4.x$
```

Every 2.0s: kubectl get all -n anjana-system

NAME	READY	STATUS	RESTARTS	AGE
pod/anjana-controller-manager-fb8db9977-ngj86	2/2	Running	0	17m
pod/drittista-0	1/1	Running	0	3m26s
pod/edusa-0	1/1	Running	0	7m16s
pod/grafana-b74667c6c-crbxb	1/1	Running	0	16m
pod/hecate-0	1/1	Running	0	5m16s
pod/hermes-0	1/1	Running	0	5m25s
pod/horus-0	1/1	Running	0	5m16s
pod/kerno-0	1/1	Running	0	5m25s
pod/minerva-0	1/1	Running	0	5m25s
pod/portuno-0	1/1	Running	0	5m24s
pod/prometheus-cb97fc757-xl9vd	1/1	Running	0	17m
pod/tot-0	1/1	Running	0	5m25s
pod/viator-0	1/1	Running	0	5m25s
pod/web-0	2/2	Running	0	17m
pod/webportuno-0	1/1	Running	0	17m
pod/zeus-0	1/1	Running	0	5m26s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/anjana-controller-manager-metrics-service	ClusterIP	10.102.181.139	<none>	8443/TCP	23m
service/drittistaserver	ClusterIP	10.104.173.5	<none>	8095/TCP	17m
service/edusaserver	ClusterIP	10.97.209.186	<none>	8888/TCP	17m
service/grafanaserver	LoadBalancer	10.106.184.105	<pending>	3000:30925/TCP	17m
service/hecatesserver	ClusterIP	10.110.109.90	<none>	50761/TCP	17m
service/hermesserver	ClusterIP	10.106.29.179	<none>	8087/TCP	17m
service/horussserver	ClusterIP	10.106.186.14	<none>	9999/TCP	17m
service/kernoserver	ClusterIP	10.108.166.43	<none>	8081/TCP	17m
service/minervaserver	ClusterIP	10.109.82.77	<none>	8084/TCP	17m
service/portunoserver	ClusterIP	10.102.32.41	<none>	8998/TCP	17m
service/prometheussserver	LoadBalancer	10.96.20.40	<pending>	9090:32761/TCP	17m
service/totserver	ClusterIP	10.100.49.70	<none>	15000/TCP	17m
service/viatorsserver	ClusterIP	10.100.160.220	<none>	8085/TCP	17m
service/webportunoservice	LoadBalancer	10.107.85.11	<pending>	8080:30395/TCP	17m
service/webservice	LoadBalancer	10.100.5.104	<pending>	8080:32497/TCP,9117:30540/TCP	17m
service/zeussserver	ClusterIP	10.100.23.22	<none>	8088/TCP	17m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/anjana-controller-manager	1/1	1	1	23m
deployment.apps/grafana	1/1	1	1	16m
deployment.apps/prometheus	1/1	1	1	17m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/anjana-controller-manager-fb8db9977	1	1	1	23m
replicaset.apps/grafana-b74667c6c	1	1	1	16m
replicaset.apps/prometheus-cb97fc757	1	1	1	17m

NAME	READY	AGE
statefulset.apps/drittista	1/1	17m
statefulset.apps/edusa	1/1	17m
statefulset.apps/hecate	1/1	17m
statefulset.apps/hermes	1/1	17m
statefulset.apps/horus	1/1	17m
statefulset.apps/kerno	1/1	17m
statefulset.apps/minerva	1/1	17m
statefulset.apps/portuno	1/1	17m
statefulset.apps/tot	1/1	17m
statefulset.apps/viator	1/1	17m
statefulset.apps/web	1/1	17m
statefulset.apps/webportuno	1/1	17m
statefulset.apps/zeus	1/1	17m

## Despliegue CRD de los plugins de Anjana

Con el controller previamente desplegado y las configuraciones aprovisionadas también es posible el despliegue de los plugins de tot mediante el siguiente comando:

```
# Desplegar todos de golpe
kubectl apply -f deploy_crds_plugins -n anjana-system

# Desplegar uno a uno
kubectl apply -f deploy_crds_plugins/anjana_v1_totpluginawss3.yaml -n anjana-system
```

En caso de que se quiera instanciar el mismo plugin repetidas veces habrá que editar el CRD correspondiente al plugin que se quiere desplegar para añadir un bloque igual al existente, descomentando el comando de arranque y alterando de forma necesaria el puerto, nombre y perfil de configuración para evitar errores en el despliegue debido a que éstos ya existirán cuando se instancie el segundo plugin. El resto de alteraciones en la configuración son opcionales.

Tras haber hecho eso el fichero debería quedar de la siguiente forma:

```
spec:
  instances:
  -
    ●name: totpawss3-1
    ●port: 15007
    size: 1
    docker_repository_url: releasesdr.anjanadata.org
    docker_repository_port: 11000
    docker_image: tot-plugin-aws-s3
    docker_version: 4.4.0
    secret_credentials_docker_repository:
      - name: anjanadr
    container_command:
      - /tot-plugin-aws-s3launcher
      - java
      - -Djava.awt.headless=true
      - -Xmx256m
      - -javaagent:/xjar-agent-hibernate.jar
      - -jar
      - /tot-plugin-aws-s3.jar
      - --spring.profiles.active=totpawss3-1
      - --spring.cloud.config.failFast=true
      - --spring.config.import=configserver:http://edusaserver:8888
  -
    ●name: totpawss3-2
    ●port: 15107
    size: 1
    docker_repository_url: releasesdr.anjanadata.org
    docker_repository_port: 11000
    docker_image: tot-plugin-aws-s3
    docker_version: 4.4.0
    secret_credentials_docker_repository:
      - name: anjanadr
    container_command:
      - /tot-plugin-aws-s3launcher
      - java
      - -Djava.awt.headless=true
      - -Xmx256m
      - -javaagent:/xjar-agent-hibernate.jar
      - -jar
      - /tot-plugin-aws-s3.jar
      - --spring.profiles.active=totpawss3-2
      - --spring.cloud.config.failFast=true
      - --spring.config.import=configserver:http://edusaserver:8888
```

## Ajustar en Edusa el origen de la configuración

Edusa se despliega con un perfil específico y con un configmap donde se especifica el origen de la configuración. Las dos alternativas actuales son:

1. Perfil native y una carpeta local con la configuración de cada microservicio
2. Perfil default y un repositorio git con la configuración de cada microservicio

Requerimientos manuales para ajustarlo :

- **Perfil native y carpeta local**

**Por defecto es el lanzamiento de Edusa, no hace falta tocar ni configmap ni comando de ejecución.**

```
cat <<EOF | kubectl replace --force -f -
apiVersion: v1
kind: ConfigMap
metadata:
  name: edusa-config-git
  namespace: 'anjana-system'
  labels:
    app: edusa
data:
  application.yaml: |
    server:
      port: 8888
    spring:
      profiles: native
      application:
        name: config-server
    cloud:
      config:
        server:
          native:
            search-locations:
              - file:/opt/data/configrepo
              - file:/opt/data/configrepo/{application}
              - file:/opt/data/configrepo/{application}/{profile}
EOF
```

- **Perfil default y repositorio Git**

El despliegue va a necesitar un secret con una variable de entorno de la llave ssh que acceso de lectura para hacer checkout del repositorio Git. Si en unos pasos anteriores se ha creado el secret sin los valores correctos se recomienda borrar el secret y crearlo correctamente.

```
kubectl create secret generic privatekey-configserver -n anjana-system
--from-literal=PRIVATEKEY-CONFIGSERVER='-----BEGIN RSA PRIVATE KEY-----
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX==
-----END RSA PRIVATE KEY-----'
```

```
azureuser@k8s:~/kit-operator-anjana-4.x$ kubectl create secret generic privatekey-configserver
-n anjana-system --from-literal=PRIVATEKEY-CONFIGSERVER='-----BEGIN RSA PRIVATE KEY-----
[REDACTED]
[REDACTED]
-----END RSA PRIVATE KEY-----'
secret/privatekey-configserver created
azureuser@k8s:~/kit-operator-anjana-4.x$
```

Ajustar el comando para levante con el perfil default:

- Cambiar native por default

```
root@k8s:~/deploy_for_clients#
root@k8s:~/deploy_for_clients# cat deploy_crds/anjana_v1_edusa.yaml
apiVersion: anjana.my.domain/v1
kind: Edusa
metadata:
  name: edusa-sample
spec:
  size: 1
  docker_repository_url: releasesdr.anjanadata.org
  docker_repository_port: 11000
  docker_image: edusa
  docker_version: 4.2.7
  secret_credentials_docker_repository:
    - name: anjanadr
  container_command:
    - /edusalauncher
    - /usr/bin/java
    - -Xmx256m
    - -jar
    - /edusa.jar
    - --spring.profiles.active=native
    - --spring.config.location=file:///application.yaml
root@k8s:~/deploy_for_clients#
```

Ajustar configmap para que coja la configuración de un repositorio git:

```
cat <<EOF | kubectl replace --force -f -
apiVersion: v1
kind: ConfigMap
metadata:
  name: edusa-config-git
  namespace: 'anjana-system'
  labels:
    app: edusa
data:
  application.yaml: |
    server:
      port: 8888
    spring:
      profiles: default
      application:
        name: <config_branch>
    cloud:
      config:
        server:
          git:
            uri: <url_repo_git> # git@bitbucket.org:repo_company/repo.git
            default-label: develop
```

```
skipSslValidation: true
timeout: 10
clone-on-start: true
force-pull: true
searchPaths: '{application}'
ignoreLocalSshSettings: true
privateKey: "${PRIVATEKEY-CONFIGSERVER}"
```

EOF