MANUAL DE USO DE OPERADOR

Versión	Fecha de publicación	Responsable	Aprobador	Resumen de cambios
1.0	30/11/2023	Dpto DS	Responsable DS	Creación del documento

Release Notes

- Se ha implementado el despliegue de los nuevos plugins a través de statefulset.
- Se ha cambiado la configuración para que se adeque a la versión 23.1

Compatibilidad de versiones

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

- 23.1

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.o5

Para actualizar se tendrá que relanzar el operador con todos los CRDs, tanto de configuración como microservicios. Para ello, seguimos los siguientes pasos

Paso 1 Recuperar personalizaciones

Los cambios o personalizaciones que se hicieran en el anterior despliegue deben ajustarse en el nuevo kit para no perderlo. Por ejemplo si se editó algún servicio o de los crds alguna imagen, puerto, namespaces, etc.

Paso 2 Cambio url y puerto repositorio docker

En la versión anterior se usaba el repositorio releasesdr.anjanadata.org:11000 pero ahora hemos movido a dr-releases.anjanadata.org por lo que hay que borrar el secret apuntando al anterior repositorio y crear el nuevo secret apuntando al nuevo:

```
kubectl --namespace anjana-system create secret docker-registry anjanadr
--docker-server=dr-releases.anjanadata.org --docker-username=<user_repo_docker>
--docker-password=<password_repo_docker>
```

Paso 3 Desplegar

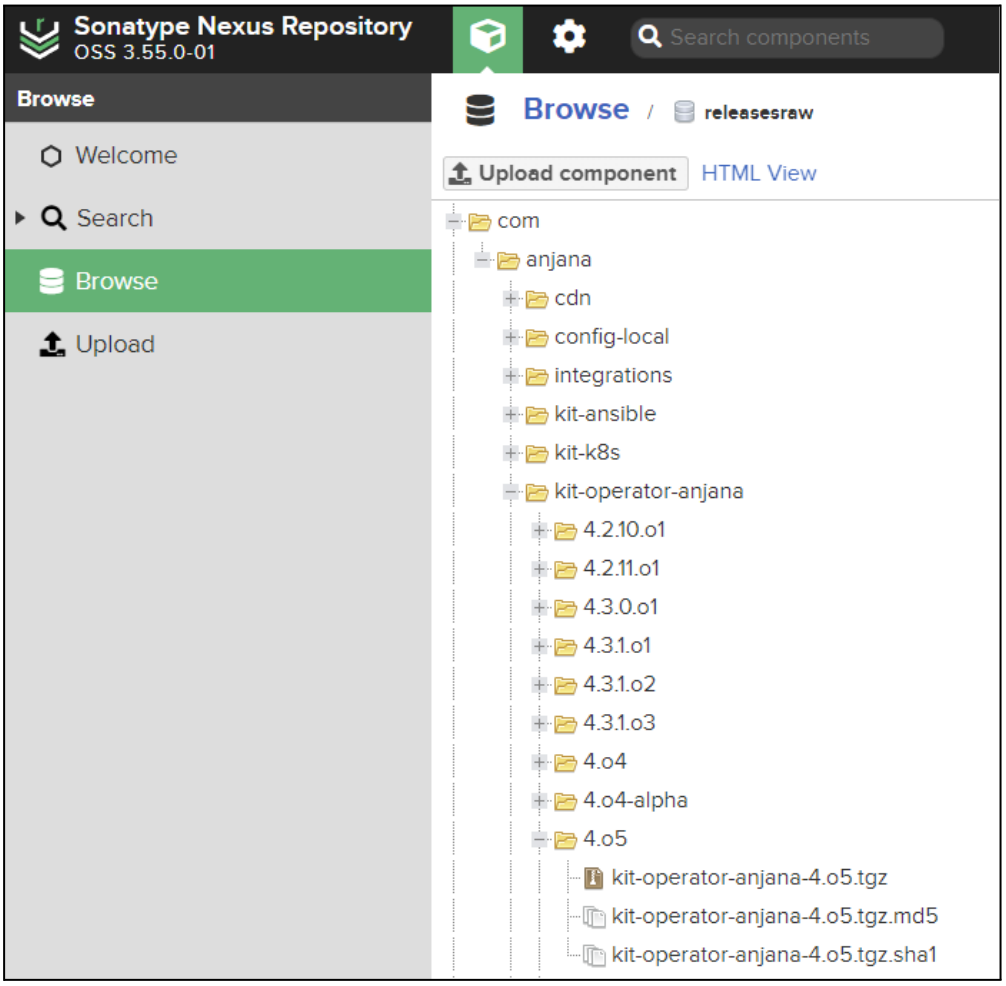
Al desplegar el archivo anjana-operator.yaml , se tomará el lanzamiento como actualizaciones de los recursos existentes, únicamente creando un nuevo pod de controlador y tirando el viejo. Una vez hecho, ya podemos lanzar los CRD con normalidad.

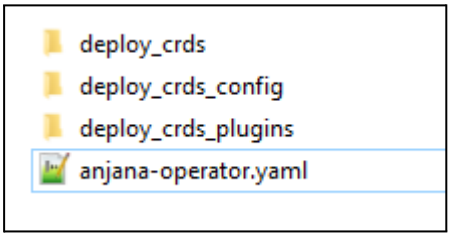
Descargar operador/controller

El kit se publica en el repositorio raw “anjanareleasesraw” 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

```
kubectl --namespace anjana-system create secret docker-registry anjanadr
--docker-server=dr-releases.anjanadata.org --docker-username=<user_repo_docker>
--docker-password=<password_repo_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

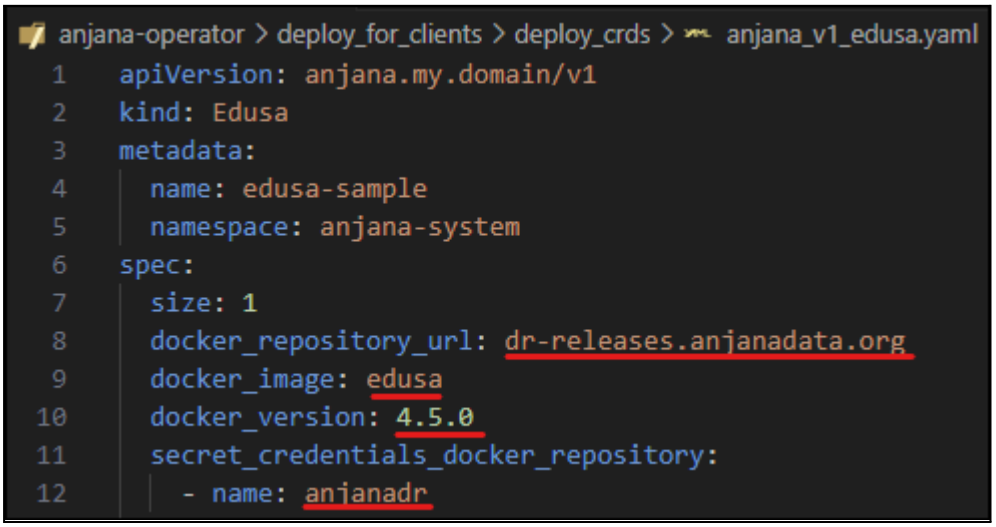
Ajustar repositorio Docker, imagen y versión

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



```
anjana-operator > deploy_for_clients > anjana-operator.yaml > {} spec > {} template > {
3566          image: dr-releases.anjanadata.org/anjana-controller:4.o5
3567          imagePullPolicy: Always
3568          livenessProbe:
3569            httpGet:
3570              path: /healthz
3571              port: 6789
3572            initialDelaySeconds: 15
3573            periodSeconds: 20
3574          name: manager
3575          readinessProbe:
3576            httpGet:
3577              path: /readyz
3578              port: 6789
3579            initialDelaySeconds: 5
3580            periodSeconds: 10
3581          securityContext:
3582            allowPrivilegeEscalation: false
3583        imagePullSecrets:
3584        - name: anjanadr
3585        securityContext:
3586          runAsNonRoot: true
3587        serviceAccountName: anjana-controller-manager
3588        terminationGracePeriodSeconds: 10
```

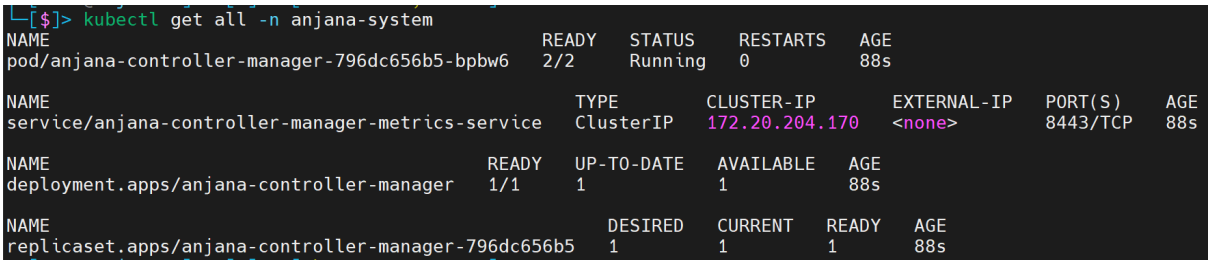
Despliegue del operador de Anjana

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

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

```
└─[$]> kubectl apply -f ../deploy_for_clients/anjana-operator.yaml -n anjana-system
namespace/anjana-system created
customresourcedefinition.apiextensions.k8s.io/configdrittesta.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/confighecates.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/confighermes.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/confighorus.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configkernoes.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configminervae.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configportunoes.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtots.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpaqtivae.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpawsglues.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpawsiams.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpawss3s.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpazureads.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpazurefiles.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpazurestorages.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpgcpbigqueries.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpgcpiams.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpgcpstorages.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtphdfs.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtphives.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpjdbcdenodoes.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpjdbcoracles.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpjdbcredshifts.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpjdbcs.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpjdbcsqlservers.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtpldaps.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtppowerbis.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtprangers.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configtptableaus.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configviators.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/configzeus.anjana.my.domain 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/hecates.anjana.my.domain created
customresourcedefinition.apiextensions.k8s.io/heimdals.anjana.my.domain created
```

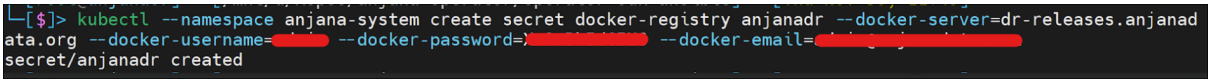
Se crearán los recursos que pueden ver en la siguiente imagen, junto con los crd que luego provisionaremos.



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
```



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-----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx==
-----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_configdrittesta.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
```

```
└─[$]> kubectl apply -f deploy_crds -n anjana-system
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
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
solr.anjana.my.domain/solr-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
```

```
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/drittesta-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/drittestaserver                             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/hecateserver                                ClusterIP      10.110.109.90    <none>        50761/TCP                    17m
service/hermesserver                                ClusterIP      10.106.29.179    <none>        8087/TCP                     17m
service/horusserver                                 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/prometheusserver                            LoadBalancer   10.96.20.40      <pending>     9090:32761/TCP               17m
service/totserver                                   ClusterIP      10.100.49.70     <none>        15000/TCP                    17m
service/viatorserver                                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/zeusserver                                  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/drittesta    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
```

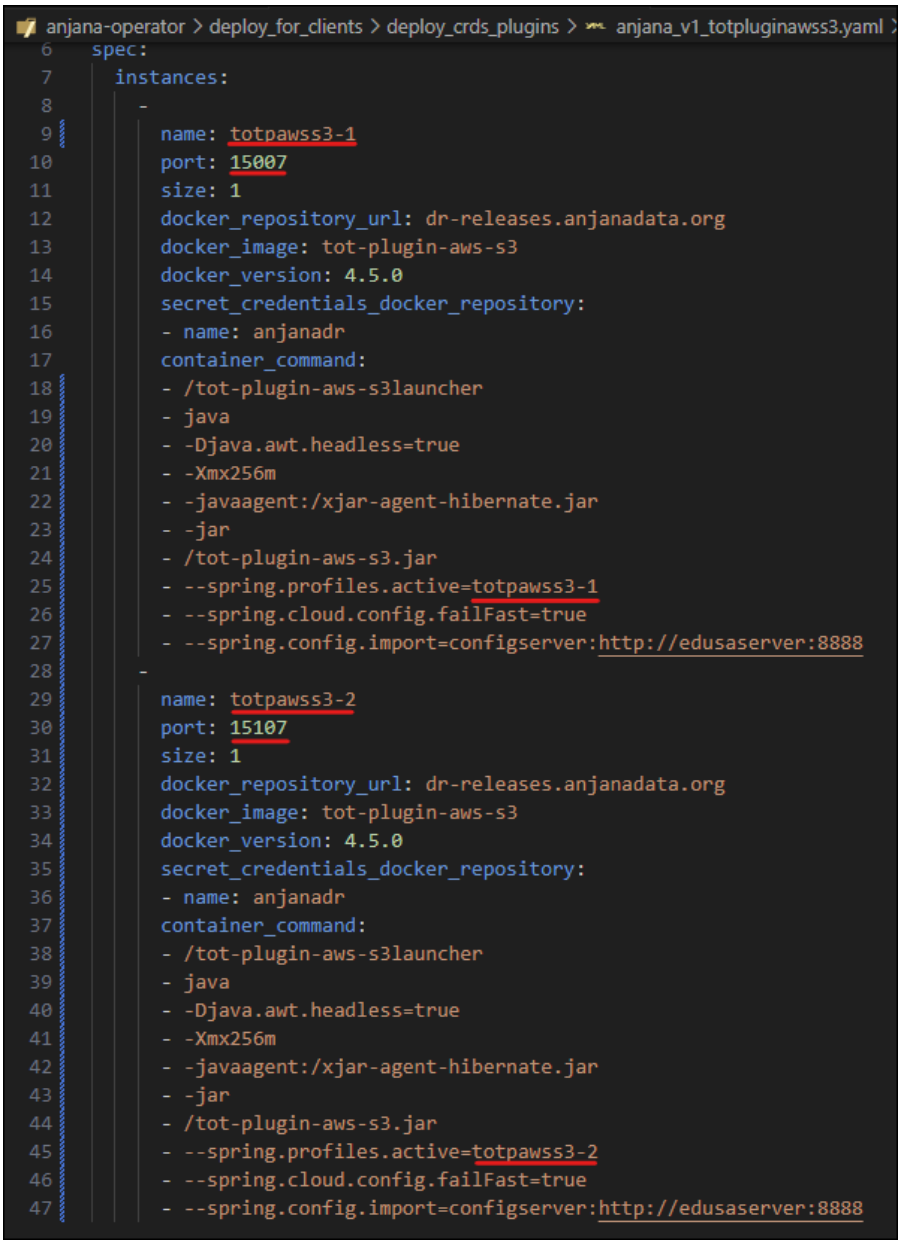
Despliegue CRD de los plugins de Anjana

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:



Ajustar en Edusa el origen de la configuración

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

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

Requerimientos manuales para ajustarlo :

- ## Perfíl native y carpeta local

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

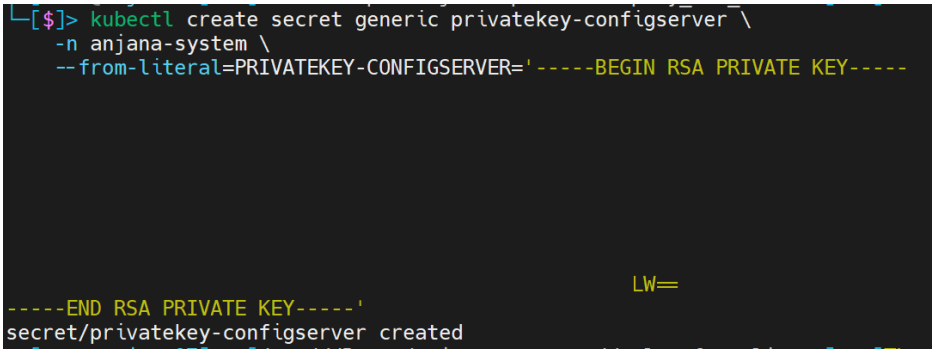
```
cat <<EOF | kubectl replace --force -f -

apiVersion: v1
kind: ConfigMap
metadata:
  name: edusa-config-git
  namespace: 'anjana-system'
  labels:
    app: edusa
data:
  application.yaml: |
    logging:
      pattern:
    server:
      port: 8888
    spring:
      config:
        activate:
          on-profile: native
      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-----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx==
-----END RSA PRIVATE KEY-----'
```



Una vez recreado el secreto hay que ajustar dos cosas para configurar el origen como repositorio git:

- Cambiar native por default como el perfil de configuración de edusa
- Ajustar configmap para que coja la configuración de un repositorio git

Ambos ajustes se pueden a continuación:

```
cat <<EOF | kubectl replace --force -f -

apiVersion: v1
kind: ConfigMap
metadata:
  name: edusa-config-git
  namespace: 'anjana-system'
  labels:
    app: edusa
data:
  application.yaml: |
    logging:
      pattern:
    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: trues
              privateKey: "${PRIVATEKEY-CONFIGSERVER}"
EOF
```