Manual de despliegue con Statefulset k8s

Versión	Fecha de publicación	Responsable	Aprobador	Resumen de cambios
1.0	28/10/2022	Dpto DS	Responsable DS	Creación del documento

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 adeque a la versión 4.4

Compatibilidad de versiones

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

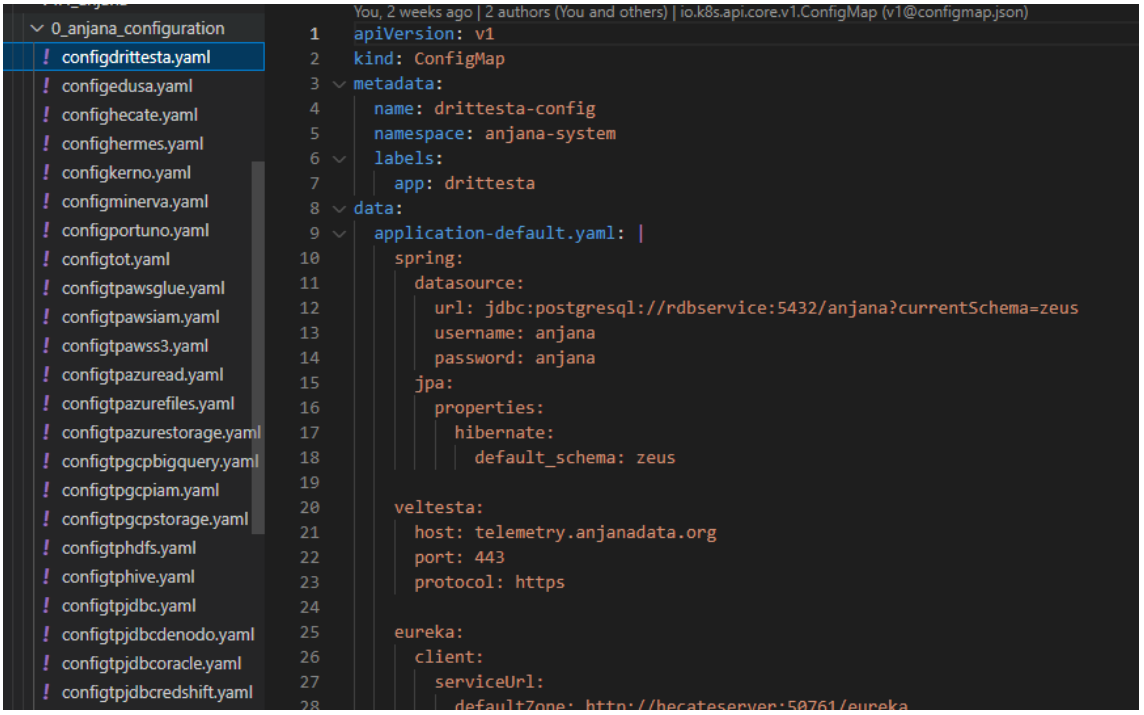
- 4.4

NOTA: En la presente versión del kit se incluyen preconfiguradas las últimas versiones 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 del despliegue a 4.s4

Paso 1 Ajustar configmaps de configuración

Debido al nuevo método de despliegue de los archivos de configuración mediante configmaps, el primer paso es exportar los antiguos archivos a los yamls de los configmaps.

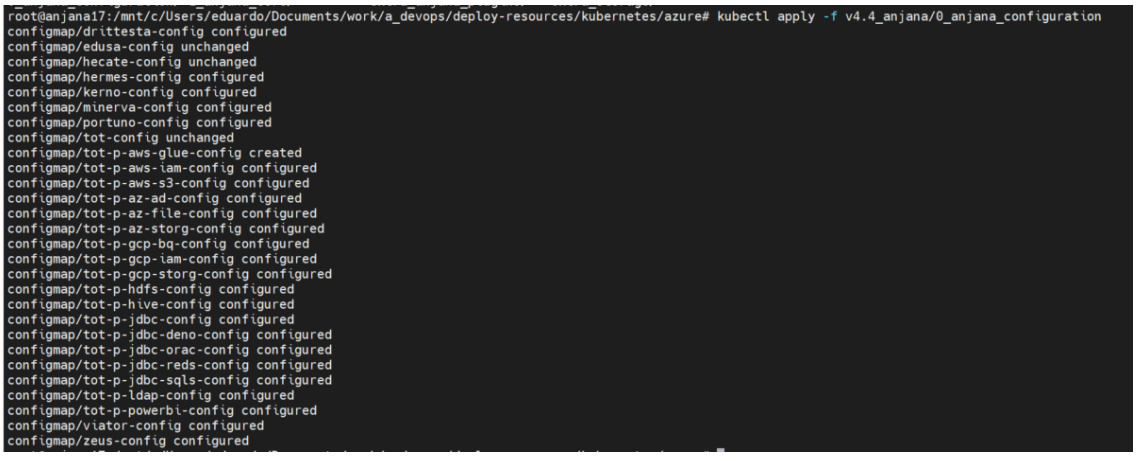


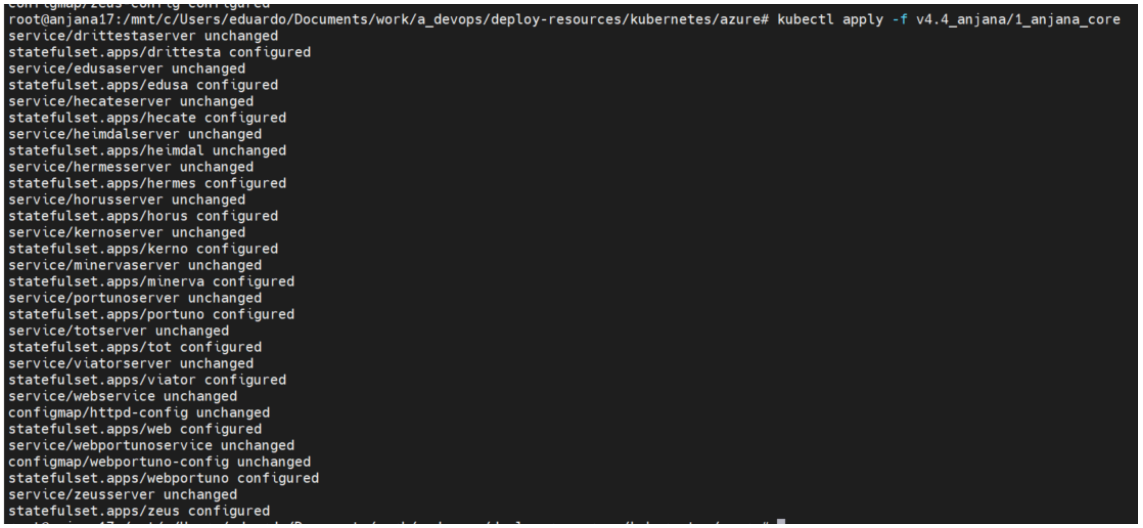
Paso 2 Recuperar personalizaciones

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 statefulset alguna image, puerto, namespaces, etc.

Paso 3 Desplegar

Al desplegar se tomará el lanzamiento como actualizaciones de los recursos existentes.





Despliegue del producto

Archivos de configuración de los microservicios

Debemos de configurar cada archivo acorde a la documentación (Anjana Data 4.x - DS - Portal and microservices config), estos archivos de configuración serán guardados en un repositorio de git, o directamente en local, en este documento, explicaremos ambos.

Opción 1: Archivos de configuración en local

En el kit descargado, veremos que hay una carpeta llamada 0_anjana_configuration , en la cual estarán los archivos de configuración de los microservicios de anjana. Si queremos la instalación con este tipo de configuración, deberemos de lanzar esta carpeta antes de los statefulsets.

Opción 2: Conectar Edusa a un repositorio externo

En esta opción, deberemos ir a la configuración de Edusa, que se encuentra en la carpeta 0_anjana_configuration, y cambiar el texto de application.yaml por uno como el siguiente.

kubernetes > Common > ! edusaserver-config.yaml > {} data > application.yaml

You, seconds ago | 1 author (You) | io.k8s.api.core.v1.ConfigMap (v1@configmap.json)

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: edusa-config
  namespace: anjanadata
  labels:
    app: edusa
    environment: demo
data:
  application.yaml: |
    server:
      port: 8888
    spring:
      profiles: default
      application:
        name: config-server
      cloud:
        config:
          server:
            git:
              uri: git@bitbucket.org:anjanadatacom/config-local.git
              default-label: develop
              skipSslValidation: true
              timeout: 10
              clone-on-start: true
              force-pull: true
              searchPaths: '{application}'
              ignoreLocalSshSettings: true
              privateKey: |
                -----BEGIN RSA PRIVATE KEY-----
                ..........................................
                -----END RSA PRIVATE KEY-----
```

Despliegue de los microservicios

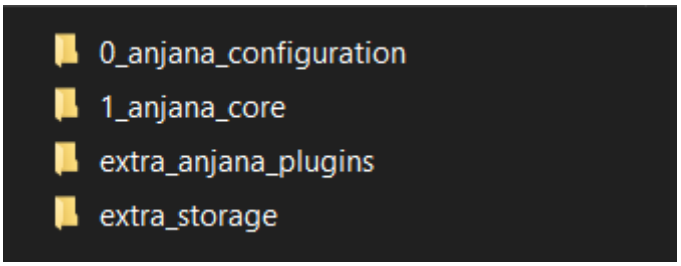
Cada archivo contiene un statefulset y un servicio que expone este servicio con una ip privada para simplificar el enrutado y las conectividades desde programas externos. Se puede cambiar este mecanismo, pero recuerda usar cualquier alternativa para exponer el producto web y la api a los usuarios y programas externos

Hay varias dependencias a considerar.

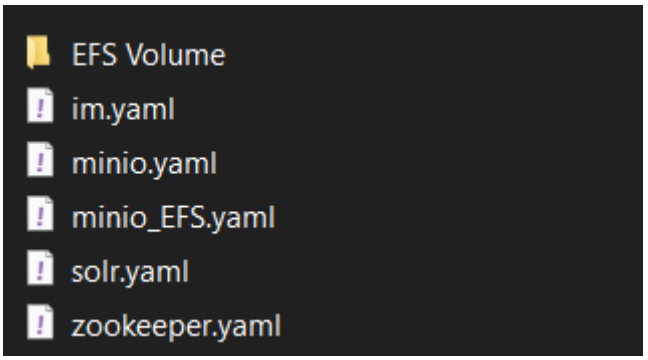
- Solr, Zk, pgsql y los programas externos deben ser lanzados antes de que se lanze anjana, ya que si no, no podrán iniciarse
- Solr depende de Zookeeper
- Todos los microservicios dependen de Edusa, ya que es quien le otorga los archivos de configuración.
- Todos los microservicios deben ser configurados antes de lanzarse.

Tenemos un tgz en Nexus con todos los yaml de los microservicios de anjana llamado kit-k8s-<version>. Con este kit, podemos descargar e instalar anjana con Statefulsets en Kubernetes

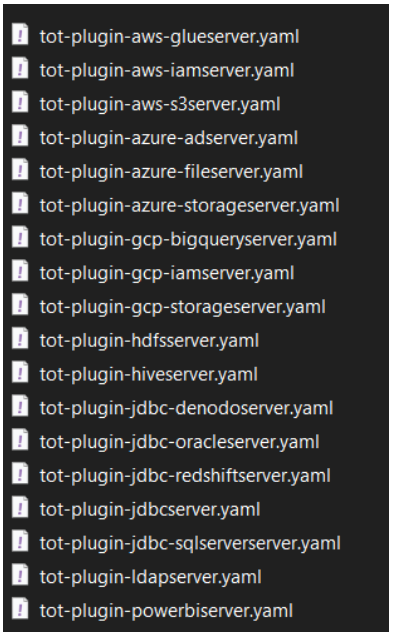
Si descomprimimos el archivo, se verá así



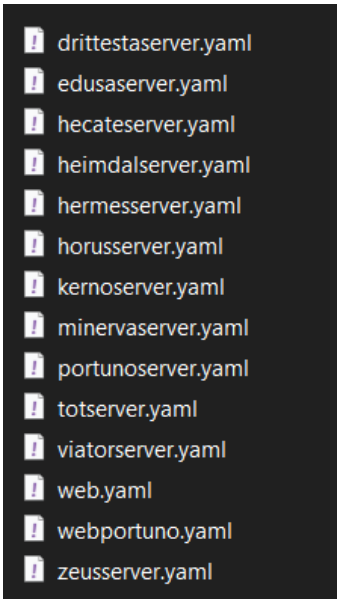
En la primera carpeta, de extra_storage, tendremos todo lo necesario para lanzar una instancia de Zk, solr y minio.



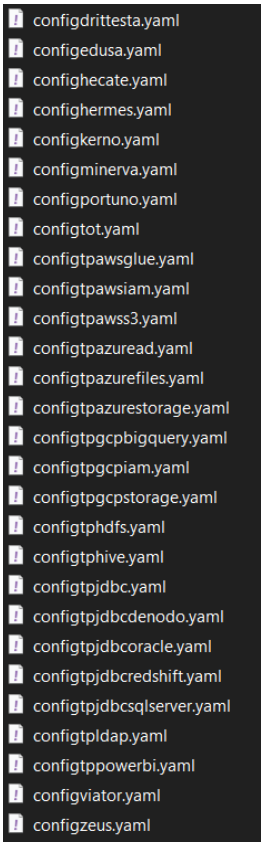
En la carpeta extra_anjana_plugins, encontraremos los Statefulset necesarios para la instalación de todos los plugins de Anjana.



En la carpeta 1_anjana_core encontraremos todos los archivos para realizar un lanzamiento de Anjana.



Y como último, en la carpeta 0_anjana_configuration encontraremos todos los archivos de configuración tanto de los microservicios de anjana como de los plugins.



Configuración de los microservicios

Una vez modificados para adecuarlos a nuestras necesidades , podemos lanzar todos los configmaps de los microservicios de la siguiente manera. Es importante hacerlo antes de lanzar los microservicios, ya que si no, estos fallarán.

```
# Para lanzar todos los archivos
kubectl apply -f 0_anjana_configuration -n anjana-system

# Para lanzar un archivo
kubectl apply -f 0_anjana_configuration/config<nombre_servicio>.yaml -n anjana-system
```

Microservicios Core de Anjana

Una vez lanzados los archivos de configuración, debemos de lanzar todos los archivos de dentro de la carpeta 1_anjana_core para levantar un anjana correctamente.

```
# Para lanzar todos los archivos
kubectl apply -f 1_anjana_core -n anjana-system
```

```
# Para lanzar un archivo
kubectl apply -f 1_anjana_core/<nombre_servicio>server.yaml -n anjana-system
```

Plugins de Anjana

De necesitar el lanzamiento de algunos de los plugins de Anjana, deberemos de, una vez lanzado su archivo de configuración, lanzar el yaml del plugin correspondiente

```
# Para lanzar todos los archivos
kubectl apply -f extra_anjana_plugins -n anjana-system

# Para lanzar un archivo
kubectl apply -f extra_anjana_plugins/<nombre_plugin>.yaml -n anjana-system
```

MinIO, Solr/Zookeeper ,Postgresql

Despliegue Solr/Zookeeper

Para desplegar estos servicios, debemos de ir donde tengamos el kit descargado, a la carpeta extra_storage, y lanzar el yaml correspondiente de Solr y Zk. Por ejemplo:

```
kubectl apply -f extra_storage/solr.yaml -n anjana-system
kubectl apply -f extra_storage/zookeeper.yaml -n anjana-system
```

Despliegue LDAP

Para desplegar estos servicios, debemos de ir donde tengamos el kit descargado, a la carpeta extra_storage, y lanzar el yaml correspondiente de IM. Por ejemplo:

```
kubectl apply -f extra_storage/im.yaml -n anjana-system
```

Despliegue Minio

Para Minio, tenemos varios modos de lanzamiento.

Tenemos el minio.yaml, el cual nos proporciona un minio con un volumen de AWS EBS, el cual tendremos que configurar previamente, y añadir el VolumeID al archivo. También debemos de añadir el accesskey y el secret_key de minio en las siguientes variables

```
        - name: 'MINIO_ROOT_USER'
          value: '<access_key>'
        - name: 'MINIO_ROOT_PASSWORD'
          value: '<secret_key>'
        ports:
        - containerPort: 9000
          name: s3service
        - containerPort: 9001
          name: minioui
        volumeMounts:
        - name: data
          mountPath: /data
      volumes:
      - awsElasticBlockStore:
          volumeID: "<volume_id>"
          fsType: ext4
        name: data
```

También tenemos otro yaml de lanzamiento de minio, llamado minio_EFS, el cual está configurado para aceptar un volumen EFS de Amazon. Para ello nos vamos a la carpeta llamada EFS Volume, y en el archivo llamado pv.yaml, debemos de añadir el volume_id del volumen EFS que hemos creado anteriormente en AWS, al igual que especificar el storage que tiene ese volumen.

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: efs-pv
  namespace: anjana-system
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: efs-sc
  csi:
    driver: efs.csi.aws.com
    volumeHandle: <volume_id>
    volumeAttributes:
      encryptInTransit: "true"
```

Cambiaremos también las variables de minio_root_user y minio_root_password en el archivo minio_EFS para adecuarlas a nuestro entorno.

```yaml
env:
- name: 'MINIO_ROOT_USER'
  value: '<access_key>'
- name: 'MINIO_ROOT_PASSWORD'
  value: '<secret_key>'
```