Configuración bróker Mosquitto

# Introducción

A la hora de desplegar el bróker Mosquitto, tendremos que realizar una serie de pasos para su configuración. Esto no pasa con otros elementos como la base de datos eXist. Aun así, gracias a esta configuración podremos crear sesiones persistentes y el bróker podrá realizar nuevas funcionalidades, como guardar la información para los suscriptores que se hayan desconectado, hasta que se vuelvan a conectar.

# Creando los volúmenes

Como ya comentamos en el tutorial de cómo crear elementos persistentes, tendremos que crear los volúmenes que vaya a necesitar antes de su despliegue, ya que en el despliegue es cuando solicitará estos volúmenes. El broker Mosquitto utilizará 3 volumenes: *mosquitto-data* (para guardar los datos que se envían al broker), *mosquitto-config* (para guardar los archivos de configuración del broker) y *mosquitto-log* (para guardar lo que enseña por consola el bróker, quien se ha conectado, desconectado…)

Por lo tanto, crearemos estos tres volúmenes con los archivos YAML, este es un ejemplo de esos tres:

apiVersion: v1

kind: PersistentVolume

metadata:

name: mosquitto-data

labels:

type: local

spec:

storageClassName: mosquitto-data

capacity:

storage: 500Mi

accessModes:

- ReadWriteOnce

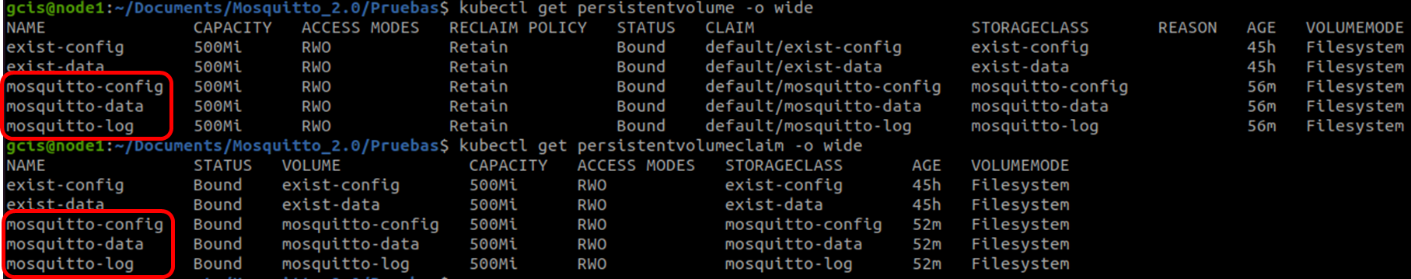
hostPath:

path: "/mosquitto/data"

# Primer despliegue

Ahora, utilizando los archivos de despliegue creados por Kompose a partir del archivo *docker-compose.yml* desplegaremos Mosquitto en el cluster. Podemos hacerlo todo a la vez, o primero los *PersistentVolumeClaim* para enlazarlos con los volúmenes creados anteriormente y después desplegar el archivo para el Deployment y Service.

Si lo hacemos paso a paso, al crear los “*claim*” podremos ver como se han enlazado con los volúmenes ya existentes:



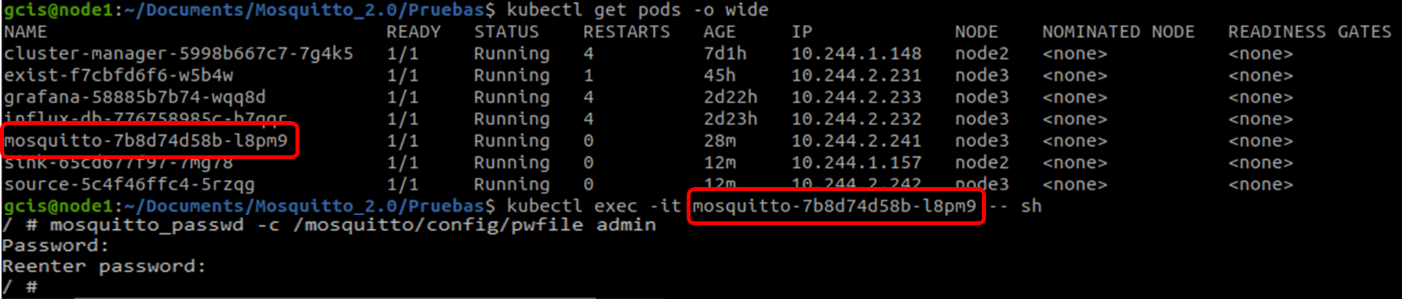
Ahora, desplegaremos los archivos Deployment y Service. Hay que tener en cuenta que el archivo Service hay que modificarlo si se acaba de crear con Kompose. Tendremos que añadirle los *nodePort* para exponerlo fuera del cluster.

# Configuración del bróker

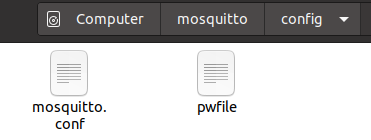
Una vez este desplegado, vamos a configurar el bróker para habilitarle conexiones externas y sesiones persistentes y también para añadirle seguridad creando un usuario y contraseña. Vamos a comenzar con la seguridad. Para crear un nuevo usuario entraremos dentro del pod y ejecutaremos el comando de creación de usuario:

> kubectl exec –it <pod name> -- sh

> / # mosquitto\_passwd –c /mosquitto/config/pwfile <nuevo usuario>



Ahora, tendremos que buscar donde se ha creado el volumen y entrar en la carpeta *config*, para modificar el archivo *mosquitto.conf*. En nuestro caso, el volumen se ha creado en el nodo 3, por lo que si vamos al directorio “/”, podremos ver la carpeta *mosquitto*. Si entramos a la carpeta *config* podremos ver que se nos ha creado el archivo con el nuevo usuario.



Ya que está en una carpeta en la cual necesitaremos editar el archivo con permisos de administrador. A la hora de editarlo, añadiremos estas líneas:

persistence true

persistence\_location /mosquitto/data/

log\_dest file /mosquitto/log/mosquitto.log

password\_file /mosquitto/config/pwfile

listener 1883

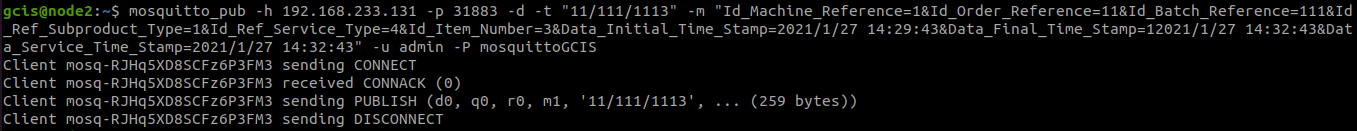
Las primeras dos líneas son para poder crear sesiones persistentes. La tercera fila es para indicarle donde está el archivo con el usuario nuevo que hemos creado. Por último, el cuarto comando es para que elementos externos se puedan conectar al bróker.

Por último paso, tendremos reinicializar el componente Mosquitto, para ellos podremos eliminar el Deployment y volverlo a crear:

> kubectl delete –f mosquitto-deployment.yaml

> kubectl apply –f mosquitto-deployment.yaml

Ahora, ya podremos conectarnos al bróker:



Ya que todo el log se guarda en un archivo, si queremos comprobar si se han conectado correctamente, podemos comprobarlo con un solo comando:

> kubectl exec -it <pod name> -- cat /mosquitto/log/mosquitto.log