**Estructura de Pipelines**

Principalmente el sistema de los pipelines se divide en 5 pipelines. Un pipeline principal para crear los recursos y desplegar los contenidos y cuatro pipelines de contenido para los cuatro recursos (Aks, Azure Function, Azure web app, Azure sql), para sí hace falta modificar el contenido de un de los recursos, entonces de esta manera no hará falta volver a desplegar toda la infraestructura de cero.

**Principal-pipeline**

Este es el pipeline principal, este pipeline es el pipeline responsable de crear todos los recursos, i también desplegar todos los contenidos de cada recurso, este pipeline llama a todos los stages que están definidos dentro del proyecto, el fichero que contiene este pipeline es el fichero principal-pipeline.yml, y tiene un trigger programado para lanzarse si se cambia un de los ficheros principal-pipeline.yml, main.bicep o el directorio bicep-templates.

**Aks-pipeline**

Este pipeline es el pipeline responsable de desplegar los contenidos dentro del aks, este pipeline contiene el stage deploy\_k8s\_content y está definido dentro del fichero aks-pipeline.yml, y su trigger se lanza en función si se cambia el fichero aks-pipeline.yml o el directorio content/ aks-content/.

**Azure-function-pipeline**

La funcionalidad de este pipeline es desplegar el contenido dentro del azure function y este pipeline solamente contiene un stage que se llama deploy\_function\_content y el pipeline está definido dentro del fichero function-pipeline.yml, y el trigger de este pipeline se lanza si se cambia el fichero function-pipeline.yml o el directorio content/ function-content/.

**Azure-webapp-pipeline**

La funcionalidad de este pipeline es el desligué del contenido del azure webapp y este pipeline solamente contiene un stage que se llama deploy\_webapp\_content y el pipeline está definido dentro del fichero webapp-pipeline.yml y su trigger se lanza si se cambia el fichero webapp-pipeline.yml o el directorio content/webapp-content/.

**Azure-sql-pipeline**

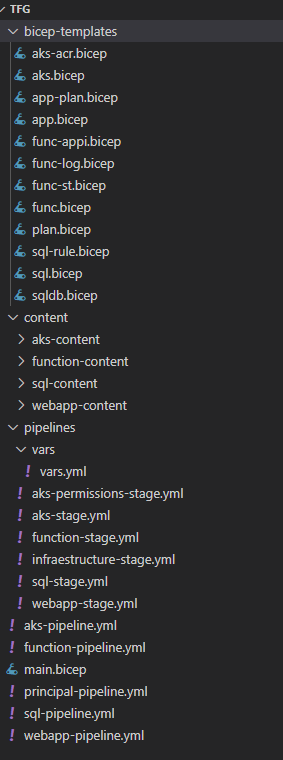
Y finalmente tenemos este pipeline que es básicamente un pipeline para desplegar la base de datos que como los otros pipelines de contenidos tiene una stage que se llama deploy\_sql\_content y el pipeline está definido dentro del fichero sql-stage.yml y tiene trigger encima del fichero sql-pipeline.yml o el directorio content/sql-content/.

**Azure Repository Structure**

En el repositorio de azure nosotros tenemos todos los ficheros importantes para nuestro proyecto, básicamente alla hay todo, y nosotros hemos decidido de tirar por este via por dos razones principales:

1. Para proteger nuestros ficheros y datos para no perderlos
2. Ya que usamos el cloud de azure, nos resultara mucho mas fácil usar azure repository que otro repository , ya que también usamos azure pipelines

En el direcory root de nuestro repository encontramos es tres carpetas y cinco ficheros , que acontinuacion vamos a explicar de que consta cada uno



**Los seis ficheros son**

primero los cinco ficheros que defienen los cinco pipelines que necesitamos para disenya nuestro sistema de integración contiuna (principal-pipeline.yml, aks-pipeline.yml, function-pipeline.yml, webapp-pipeline.yml, sql-pipeline.yml,main.bicep) y main.bicep que es el fichero principal que llama a las diferentes plantillas para desplegar la infraestructura de recusos dentro de azure.

**Los tres directories son los siguientes**:

bicep-template

En este directorio hay 11 ficheros, que contienen los diferentes plantillas de los diferentes recusos usados para desplegar la infraestructura, las plantillas están definidas en bicep, y todas se llaman desde el main.bicep a contiuacion tenemos la lista de los ficheros y el contenido de cada fichero

* aks-acr.bicep: Este fichero contiene la plantilla que se usa para crear el registro de contenidors que usar el aks para obtener la imagen para los pods
* aks.bicep: Aqui esta definida la planilla del aks para desplegarse
* app-plan.bicep: Aqui esta definida el app service plan del webapp para definir los caracteristiques de la maquina que hostea el webapp
* app.bicep: Aqui estan las configuracio del webapp
* func-appi.bicep: Este fichero se usa para el appliacation insighith que se usa con el azure function
* func-log.bicep: Este fichero se usa para configurar el log analytic para el azure function
* func-st.bicep: Aquí se crea el storage account que se usa para que el azure function puede guradar sus datos
* func.bicep: Este fichero se usa para configurar el azure function
* plan.bicep: aqui se configura el app service plan para el azure function y para el webapp
* sql-rule.bicep: Este fichero es para configurar los accesos en el firewall de la sql que por defecto todo es negado.
* sql.bicep: aquí se configura la base de datos dentro del servidor
* sqldb.bicep: Aquí es el servidor sql que sostiene la base de datos

content

Este directorio contiene 4 subdirectorios aks-content, function-content, sql-content, webapp-content, que cada uno de estos cuatro subdirectorios contiene el contenido que se tiene que usar en los recursos (azure kubernetes service, azure function, azure webapp, azure sql).

aks-content: aquí hay 6 subdirectorios cada uno es para un componente diferete dentro del aks.

* App: este subdirectorio se usa para desplegar la aplicación del cliente dentro del aks.
* docker-images: este subdirectiorio contiene 2 imágenes de docker una para la aplicación del cliente y otra para el exportador de métricas que usar prometheus para traer las métricas de azure
* ingress-ngnix: aquí esta definido el contenido que se usar para desplegar el controlador de ingress-ngnix
* kube-prometheus-stack: aquí están las definiciones de los diferentes componenete que se usen para prometheus y grafana para correr dentro del cluster
* prometheus-exporter: este es el subdirectorio que define el despligue del exportador de métricas que usa primetheus dentro de azure

Pipelines

dentro de este directorio hay un subdirectorio que solamente tiene un fichero que se llama var.yml que básicamente es la lista de los variables que se usen en todo el sistema y también tiene

* vars: es un sibdirectirio que contiene un fichero que se llama vars.yml que alla están difinidas todos las variables de toda el sistema
* aks-permissions-stage.yml: aquí esta definida el stage que actua para dar el cluster de k8s los permisos necesarios para colectar las métricas dentro de grafana
* aks-stage.yml: es un fichero donde esta definido el stage responable de subri el contenido de aks en el mismo
* function-stage.yml: este fichero es el responsable de subir la función definida del usuario dentro del aure function
* infraestructure-stage.yml: aquí está definida el stage que deplega el código de bicep para desplegar toda la infraestructura
* sql-stage.yml: en este fichero esta definida la stage resposable de subrir la base de datos en el sql service de azure
* webapp-stage.yml: este stage es responable de de subir el código de la aplicación web en el web app