**Indicaciones para despliegue de proyecto integrador**

**(Challenge – Java 03)**

**Paso 1:** Descargar las imágenes subidas a un repositorio público en DockerHub (no se consideró el despliegue de las imágenes en un Azure Conteiner Registry, debido a que cada una pesaba más de 450MB y el servicio tiene un costo por alojar las imagenes).

Realizar un **docker pull** a las siguientes imágenes:

**Nota:** BS - bussines, UX - expereince

* **ajorge2893/config-server** (servicio de Spring Cloud Config Server)
* **ajorge2893/eureka-server** (servicio de Spring Cloud Eureka)
* **ajorge2893/servicing-payment** (servicios de API BS Service Payment)
* **ajorge2893/favorites** (servicios de API BS Favorites)
* **ajorge2893/payments** (servicios de API UX Payments)
* **ajorge2893/api-gateway** (servicio de Spring Cloud Gateway)

**Paso2:** Descargar las fuentes de los servicios que se encuentran en el repositorio público:

[**https://github.com/JorgeAlamo/challenge-java-03.git**](https://github.com/JorgeAlamo/challenge-java-03.git)

**Paso3:** Desplegar un servicio de Azure Kubernetes en Azure Portal y de no contar con un Resource Group, crearlo previamente dentro de alguna Suscripción que se tenga disponible.

**Nota:** es necesario esperar a la creación de cada uno de los recursos/componentes antes de continuar.

Creación de Resource Group

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Creación de Servicio de Kubernetes, se crea dentro del Resource Group mencionado anteriormente-

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Paso 4:** Instalar la herramienta Azure CLI en el dispositivo local para interactuar con el AKS desde consola de comandos. Enlace a la documentación de Microsoft respecto a la instalación.

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>

**Paso 5:** Acceder al componente AKS creado previamente, en la sección de **Información General**, dar clic sobre el botón **Connectar**. Considerar los 2 primeros comandos que se muestran en la ventana emergente, se utilizarán en los siguientes pasos.

Conexión a servicio AKS.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 6:** En el dispositivo local iniciar una consola de comandos (es necesario ya tener instalado Azure CLI) y ejecutar el comando:

**Nota:** antes de realiza el login, considerar eliminar la carpeta .kube que se encuentra en el fichero de users (C: -> usuarios -> nombre.usuario) para evitar problemas al realizar la acción.

* az login

Al ejecutarlo se abrirá el navegador por defecto y se solicitará realizar el login con la cuenta de Azure correspondiente a la suscripción en la que se han desplegado los componentes.

**Paso 7:** Una vez realizado el login, ejecutar los comandos identificados en el paso 5 de forma ordenada. Considerar el siguiente ejemplo de referencia:

* **az account set --subscription 75fgg6c3-b489-4a45-9634-260d4515685Yf**
* **az aks get-credentials --resource-group rgchallenge --name akschallenge**

**Paso 8:** En el dispositivo local, desde la consola posicionarse sobre la carpeta en la que se descargaron las fuentes de repositorio **(paso 4)**, e ingresar a la carpeta **07 – manifest**.

Consola asociada a AKS mediante Azure CLI y login

Texto

Descripción generada automáticamente

Estructura de carpetas

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Dentro se encontrarán los siguientes archivos según la imagen:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Paso 9:** Ejecutar en consola los siguientes comandos en orden:

**Nota:** considerar contar con las extensiones necesarias para ejecutar los comandos kubectl, respecto a la implementación se realizó en un dispositivo con Docker Desktop que ya cuanta con lo requerido para utiliza el comando.

* **kubectl apply -f mongo-db.yml**
* **kubectl apply -f mysql-db.yml**
* **kubectl apply -f redis-cache.yml**
* **kubectl apply -f config-server.yml**
* **kubectl apply -f eureka-server.yml**

**Paso 10:** Ejecutar el comando para crear un deployment y servicio correspondiente a Keycloak con el siguiente comando:

* **kubectl apply -f keycloak-auth.yml**

Ejecución de comandos en consola

Texto

Descripción generada automáticamente

Validar el despliegue de los deployments con el comando:

* **Kubectl get deployments**

Texto

Descripción generada automáticamente

Validar el despliegue de los servicios con el comando:

* **Kubectl get services**

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

Apuntar el valor de EXTERNAL-IP correspondiente al servicio keycloak-auth, se utilizará en el siguiente paso.

**Paso 11:** Con el valor de la ip publica para el servicio keycloak-auth, modifica los siguientes archivos en la carpeta **07 – manifest**:

* **favorites.yml**
* **servicing-payment.yml**
* **payments.yml**
* **api-gateway.yml**

En la sección de env para el valor SERVICES\_URL\_KEYCLOAK, colocar la ip pública obtenida en el paso anterior (realiza la acción para cada uno de los archivos listados ates de realizar el despliegue).

Texto

Descripción generada automáticamente

**Paso 12:** Realizado el cambio, ejecutar los siguientes comandos:

* **kubectl apply -f favorites.yml**
* **kubectl apply -f servicing-payment.yml**
* **kubectl apply -f payments.yml**
* **kubectl apply -f api-gateway.yml**

Texto

Descripción generada automáticamente

Validar el despliegue de los deployments con el comando:

* **Kubectl get deployments**

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Validar el despliegue de los servicios con el comando:

* **Kubectl get services**

Imagen que contiene Calendario

Descripción generada automáticamente

Considerar las ip públicas (EXTERNAL-IP) de los servicios keycloak-auth (generar token de seguridad) y api-gateway (acceder a las apis) para interactuar con los servicios. Adicionalmente mencionar que estas ips pueden cambiar en cada implementación, así que tenerlo en cuanta al momento de realizar las pruebas, se adjunta un postman con los endpoint necesarios para validar en el repositorio de Github con las fuentes del proyecto.

**Paso 13:** Configurar Keycloak accediendo a las ip pública del servicio keycloak-auth en un navegador por el puerto 8080. Posteriormente dar clic en **Administration Console**.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Acceder a la herramienta con las credenciales admin para user y password.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Crear un nuevo “realm”, dar clic al botón **Add realm**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Colocar en el campo **Name** el valor **oauth-secure**. Dar clic en **Create**.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Crear un nuevo cliente, acceder al apartado **Clients**.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En el campo **Client ID** colocar el valor **service-gateway**, el resto de los campos similar a la imagen.

Captura de pantalla de un celular

Descripción generada automáticamente

Posterior a la creación de cliente, se posicionará en la pestaña de **Settings**. Seleccionar la opción **Access Type ID** en **confidencial,** colocar en **ON** las opciones de **Authorization Enabled** y **Service Accounts Enabled**, en el campo **Valid Redirect URLs** colocar **\*** y para finalizar guardar los cambios dando clic en el botón **Save**. Similar a las imágenes adjuntas.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

A continuación, en la pestaña **Credentials**, copiar el valor del **secret** que será necesario posteriormente para poder generar el token de autorización al momento de realizar las pruebas.

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

Como siguiente paso corresponde acceder a la sección de **Users** y dar clic en el botón de **Add user.**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En el formulario de registro colocar en el campo **Username** el valor **gateway** y dar clic en **Save**.

Captura de pantalla de un celular

Descripción generada automáticamente

Se habilitarán pestañas en el formulario, acceder a **Credentials**. Marcar la opción **Temporary** en **OFF** y colocar como password el valor **gateway** (colocar el mismo password en el campo de confirmación).

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Dar clic en el botón Set password y confirmar la acción en la ventana emergente.

**Nota**: es importante que se coloque la opción **Temporary** en **OFF** para generar los tokens de seguridad sin inconvenientes.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Realizados los pasos indicados en este manual se tendrá lo necesario para realizar las pruebas de los servicios solicitados en el reto. Tener en cuanta que los valores resaltados en el presente documento serán necesarios para poder autenticarse y consumir los servicios.