SPEI Core - Manual de Instalación

Historial de Cambios

Fecha	N° de revisión	Descripción de cambios
11/01/2022	01	*Documento inicial*
21/02/2022	02	*Reestructuración de contenido*

Contenido

Introducción

Este microservicio, implementa la funcionalidad de autenticación y autorización centralizada para los demás componentes que conforman la solución.

Requisitos

Para la ejecución de este servicio, es necesario contar con los siguientes componentes:

- *spei-core.yml* → Archivo base de configuración.
- *spei-core-log4j.yml* → Archivo base de configuración de log.
- *ef-spei-core-[version].jar* → Archivo java correspondiente al microservicio. El campo [version] especifica la versión del componente.
- *keystore.p12* → Archivo de almacenamiento de llaves en el que deberá existir el juego de llaves pública/privada para identificarse con los demás microservicios.
- *truststore.p12* → Archivo de almacenamiento de llaves donde serán guardados los certificados públicos validos para establecer comunicación con el microservicio.
- La propiedad `server.ssl.key-alias` debe coincidir con los alias del certificado
- La propiedad `eureka.instance.hostname` debe coincidir con los CN del los certificados

Para mayor información sobre los almacenes de certificados, consultar el documento: Crypto Tools

Recomendaciones

Se recomienda crear la imagen considerando:

- Utilizar la imagen `eclipse-temurin:17.0.6_10-jre-ubi9-minimal` o similar para ejecutar el servicio dentro del contenedor.
- Crear los directorios en la imagen y los volúmenes correspondientes para:
  - /app/bin → Este directorio, almacenara las clases y recursos necesarios por el aplicativo.
  - /app/logs → Este directorio, deberá contar con permisos de lectura/escritura
  - /app/certs → Este directorio, deberá contar con permisos de lectura
  - /app/config → Este directorio, deberá contar con permisos de lectura
- Extraer la aplicación mediante el comando:

```
RUN java -Djarmode=layertools -jar ef-svc-reg.jar extract
```

- Utilizar variables de ambiente para la configuración de propiedades.
- Almacenar el archivo `spei-core.yml` dentro del volumen `/app/config` del servicio `config server`.

Configuración

Para el correcto funcionamiento de este microservicio, será necesario modificar el archivo de propiedades en formato yaml. Este archivo y su configuración default se muestra a continuación:

```yaml
server:
  ssl:
    key-alias: spei-core
    client-auth: none

spring:
  sql:
    init:
      mode: never
  datasource:
    url: database_url
    username: database_user
    password: database_user_password
    driver-class-name: database_driver_class_name
    hikari:
      maximum-pool-size: database_max_pool_size
  jpa:
    properties:
      hibernate:
        cache:
          region:
            factory_class: infinispan
          use_second_level_cache: true
          infinispan:
            statistics: true
            com.lgec.enlacefi.spei3.core.ordenes.envios.model.OrdenEnviadaEntity:
              cfg: replicated-entity
              memory.size: 0
              expiration.lifespan: 3600000 # 1 hour
              expiration.max_idle: 3600000 # 10 minutes
            com.lgec.enlacefi.spei3.core.ordenes.envios.model.OrdenRecibidaEntity:
              cfg: replicated-entity
              memory.size: 0
              expiration.lifespan: 3600000
              expiration.max_idle: 3600000
```

```
          generate_statistics: false
          jakarta:
            persistence:
              sharedCache:
                mode: ENABLE_SELECTIVE
    open-in-view: false
  liquibase:
    change-log: classpath:/db/changelog/db.changelog-master.xml
  security:
    oauth2:
      client:
        registration:
          ef-sso:
            client-id: spei-core
            client-authentication-method: private_key_jwt
            authorization-grant-type: client_credentials
        provider:
          ef-sso:
            issuer-uri: http://auth_server:auth_port/realms/spei
        extensions:
          credentials:
            spei-core:
              keystore: /path_to/keystore.p12
              keystore-password: ks_password
              alias: spei-core

logging:
  config: /path_to/spei-core-log4j.yml

eureka:
  instance:
    hostname: instance_host_app

infinispan:
  embedded:
    configXml: cache/infinispan-spei-core.xml

ef:
  spei:
    core:
      adapter:
        alias-firmas-adapter: alias_adapter
```

Para mayor información sobre las propiedades utilizadas en el archivo de configuración, consultar el documento Descripción de Propiedades.

Despliegue

Para desplegar el contenedor del microservicio, se deberá ejecutar el comando:

```
ENTRYPOINT java [jvm_opts] \
    --add-opens java.base/java.security=ALL-UNNAMED \
    --add-opens java.base/java.util.concurrent=ALL-UNNAMED \
    --add-opens java.base/java.math=ALL-UNNAMED \
    --add-opens java.base/javax.crypto.spec=ALL-UNNAMED \
    --add-opens java.base/sun.security.x509=ALL-UNNAMED \
    --add-opens java.base/sun.security.pkcs=ALL-UNNAMED \
    --add-opens java.base/java.security.spec=ALL-UNNAMED \
```

```
    --add-opens java.base/sun.security.util=ALL-UNNAMED \
    --add-opens java.base/javax.crypto=ALL-UNNAMED \
    --add-opens jdk.crypto.ec/sun.security.ec=ALL-UNNAMED \
    --add-opens java.base/java.lang=ALL-UNNAMED \
    --add-opens java.base/java.util=ALL-UNNAMED \
    --add-opens java.base/java.time.format=ALL-UNNAMED \
    org.springframework.boot.loader.JarLauncher \
    --server.port=[service_port] \
    --config.url=[config_server] \
    --tls.ks=file:[path_to]/[ks_name] --tls.ks.pwd=[ks_pwd] \
    --tls.ts=file:[path_to]/[ts_name] --tls.ts.pwd=[ts_pwd] \
    [program_opts]
```

Donde:

- *[config_server]* → Corresponde a la url donde se encuentra desplegado el servicio `config server.`
- *[path_to]* → ruta donde se ubican los archivos de configuración / almacenes de certificados.
  - Por ejemplo, para Linux:
    - /user_home/spei-core/config
    - /user_home/security
  - Para Windows
    - C:\SOME_USER\spei-core\config
    - C:\SOME_USER\security
- *[ks_name]* - nombre del almacen keystore en formato p12
- *[ks_pwd]* - password del archivo keystore
- *[ts_name]* - nombre del almacen truststore en formato p12
- *[ts_pwd]* - password del archivo truststore
- *[service_port]* → Puerto que utilizará el servicio para levantar.
- *[jvm_opts]* → Variable que permite establecer opciones adicionales a la jvm.
- *[program_opts]* → Variable que permite establecer opciones adicionales al microservico.

Una vez ejecutado el comando con los valores correspondientes a las variables, antes mencionadas, se podrá ver en el log del microservicio, una salida parecida a:

```
Started Spei3CoreApplication in 66.8 seconds (process running for 69.233)
Processor [ordenes-enviadas-projections] claimed 16 new segments for processing
```

Una vez iniciado el microservicio, éste se registrará de forma automática en eureka.