Artículo
Mathew Lambert · 25 mar, 2020 Lectura de 6 min

Uso de Docker con tu repositorio de desarrollo de InterSystems IRIS

¡Hola Comunidad!

Creo que hoy en día todo el mundo guarda el código fuente de sus proyectos en repositorios como Github, GitLab, bitbucket, etc. Lo mismo sucede con proyectos de InterSystems IRIS, se pueden ver algunos ejemplos en Open Exchange.

¿Qué hacemos cada vez que empezamos o continuamos nuestro trabajo con un repositorio en particular con la plataforma de datos InterSystems?

Necesitamos una máquina de InterSystems IRIS local, configurar el entorno para el proyecto e importar el código fuente.

Por eso, cada desarrollador hace lo siguiente:

1. Bajarse el código del repositorio
2. Instalar/Ejecutar una instalación de IRIS local
3. Crear un nuevo namespace/database para un proyecto
4. Importar el código a ese nuevo namespace
5. Configurar el resto del entorno
6. Iniciar/continuar codificando el proyecto

Si pasas tu repositorio a Docker, estos pasos podrían reducirse a 3:

1. Bajarse el código del repositorio
2. Ejecutar docker-compose build
3. Iniciar/continuar desarrollando el proyecto

Ventaja - evitar los pasos 3, 4 y 5, que pueden llevar varios minutos y de vez en cuando generan algún dolor de cabeza.

Puedes dockerizar (casi) cualquiera de tus repositorios de InterSystems en unos pocos pasos. ¡Comenzamos!

Cómo dockerizar el repositorio y lo que significa

Básicamente la idea es tener Docker instalado en tu máquina, que compilará el código y el entorno en un contenedor, que a su vez se ejecutará en Docker y funcionará tal como el desarrollador lo introdujo en el primer lugar. Así nos olvidamos de preguntarnos "¿Cuál es la versión del sistema operativo?" o "¿Qué más tenía en esta instalación de IRIS?".

Cada vez tenemos una página limpia (o un contenedor de IRIS limpio) que podremos usar para configurar el entorno (namespaces, bases de datos, web-apps, usuarios/roles) e importar código a una base de datos limpia y recién creada.

¿Este procedimiento de "dockerizar" dañará gravemente tu repositorio actual?

No. Necesitará añadir 2 o 3 nuevos archivos en la raíz del repositorio y seguir unas pocas reglas que podrás configurar por ti mismo.

Requisitos previos

Descargar e instalar docker.

Descargar e instalar la imagen de IRIS docker. En este ejemplo, usaré la vista previa completa de InterSystems IRIS: iris:2019.1.0S.111.0, que puedes descargar de WRC-preview., mira los detalles.

Si trabajas con una instancia que requiere una clave, coloca el iris.key en el lugar que usarás todo el tiempo. Yo la pongo en mi Mac en el directorio Home.

Dockerizar el repositorio

Para dockerizar tu repositorio, deberás añadir tres archivos a su carpeta raíz.

Este es un ejemplo de un repositorio dockerizado - el proyecto ISC-DEV, que ayuda a importar/exportar código fuente desde la base de datos IRIS. Este repositorio tiene un Dockerfile adicional, docker-compose.yml e installer.cls que describiré a continuación.

Primero está Dockerfile, que será usado por el comando build de docker-compose.

Dockerfile

Este Dockerfile copia installer.cls y el código fuente desde la carpeta /cls del repositorio a la carpeta /src en el contenedor.

También ejecuta algunos ajustes de configuración, que otorgan al usuario admin el rol %All, contraseña infinita "SYS", introduce la autorización de nivel de sistema operativo y ejecuta el %Installer.

¿Qué contiene %Installer?

```
Class App.Installer

{


XData MyInstall [ XMLNamespace = INSTALLER ]

{

<Manifest>

  <Default Name="NAMESPACE" Value="ISCDEV"/>

  <Default Name="DBNAME" Value="ISCDEV"/>

  <Default Name="APPPATH" Dir="/opt/app/" />

  <Default Name="SOURCESPATH" Dir="${APPPATH}src" />
```

```
  <Default Name="RESOURCE" Value="%DB_${DBNAME}" />



  <Namespace Name="${NAMESPACE}" Code="${DBNAME}-CODE" Data="${DBNAME}-DATA" Create="
yes" Ensemble="0">

    <Configuration>

      <Database Name="${DBNAME}-CODE" Dir="${APPPATH}${DBNAME}-CODE" Create="yes" Res
ource="${RESOURCE}"/>

      <Database Name="${DBNAME}-DATA" Dir="${APPPATH}${DBNAME}-DATA" Create="yes" Res
ource="${RESOURCE}"/>

    </Configuration>

    <Import File="${SOURCESPATH}" Recurse="1"/>

  </Namespace>



</Manifest>

}



ClassMethod setup(ByRef pVars, pLogLevel As %Integer = 3, pInstaller As %Installer.In
staller, pLogger As %Installer.AbstractLogger) As %Status [ CodeMode = objectgenerato
r, Internal ]

{

  Return ##class(%Installer.Manifest).%Generate(%compiledclass, %code, "MyInstall")

}



}
```

Crea el namespace/database ISCDEV e importa el código desde la carpeta fuente -/src.

A continuación viene el archivo docker-compose.yml, que se usará cuando ejecutemos el contenedor con el comando up de docker-compose.

```
version: '2.4'

services:

  iris:

    build: .
```

```
    restart: always

    ports:

      - 52773:52773

    volumes:

      - ~/iris.key:/usr/irissys/mgr/iris.key
```

Esta configuración le indicará a Docker en qué puerto esperamos que IRIS trabaje en nuestro host. Primero (52773) es un host, segundo es el puerto interno del contenedor de un contenedor (52773)

En la sección de volúmenes, docker-compose.yml otorga acceso a una clave de IRIS en tu máquina dentro del contenedor en el lugar, donde IRIS la buscará:

```
- ~/iris.key:/usr/irissys/mgr/iris.key
```

Para empezar a codificar con este repositorio, hay que hacer lo siguiente:

1. Clonar/hacer un pull de git del repositorio hacia cualquier directorio local.

2. Abrir el terminal en este directorio y ejecutar

```
user# docker-compose build
```

esto compilará el contenedor.

3. Ejecutar el contenedor IRIS con tu proyecto

```
user# docker-compose up -d
```

Abre tu IDE favorito, conéctate al servidor en localhost://52773 y desarrolla con InterSystems IRIS Data Platforms ;)

Puedes usar estos 3 archivos para dockerizar tu repositorio. Simplemente tienes que introducir el nombre correcto del código fuente en Dockerfile, el o los namespace(s) adecuados en Installer.cls y el sitio de iris.key en docker-compose.yml y usar los beneficios de los contenedores Docker en tus tareas diarias de desarrollo con InterSystems IRIS.

#Contenedorización #Docker #Entorno de desarrollo #Mejores prácticas #InterSystems IRIS

URL de fuente:https://es.community.intersystems.com/post/uso-de-docker-con-tu-repositorio-de-desarrollo-de-intersystems-iris