記事
Toshihiko Minamoto · 2020年12月25日 8m read

Docker for Windows で InterSystems IRIS コンテナを使用する

InterSystems は自社が提供する InterSystems IRIS の Dockerイメージについて、Linux 環境での使用のみをサポートしています。 Docker for Windows は Linux プラットフォームのようにコンテナをネイティブプロセスとして実行するのではなく、Windows の仮想化機能である Hyper-V で実行される Linux VM を作成してコンテナをホストします。 このような追加レイヤーによって複雑度が増しているため、InterSystems は現時点で Docker for Windows をサポートすることができません。

ただし、テストやその他の特殊な目的のために InterSystems の IRIS ベースのコンテナを Docker for Windows 環境で実行する必要があることは私たちも理解しています。 この記事では、InterSystems が提供するコンテナイメージの使用に関して InterSystems が認識している Docker for Windows と Docker for Linux の違いについて説明します。 その他にも予期しない問題が発生する可能性はあります。 Windows プラットフォームで InterSystems IRIS のイメージとコンテナを使用する場合は、参考のために Docker ドキュメント（特に「Getting started with Docker for Windows」）にアクセスできることを確認してください。

Docker for Windows 環境のコンテナで外部永続ストレージを操作する場合は、Windows および Linux のファイルシステムとファイル操作の両方が関係してくるため、ここでは主にストレージに関係する違いを説明しています。

InterSystems が提供するイメージを使用して InterSystems IRIS を Docker コンテナで実行する場合の一般的な情報については、「Running InterSystems IRIS in Docker Containers」と「First Look: InterSystems IRIS in Docker Containers」を参照してください。

ディスクドライブを共有する

Windows では Docker が存在するディスクドライブを共有し、Docker に操作対象となるすべてのストレージへのアクセスを許可する必要があります。 1 台以上のドライブを共有するには、次の手順に従ってください（この手順は前の項目の手順と組み合わせることができます）。

1. システムトレイの Docker アイコンを右クリックし、[**Settings ...**] を選択します。
2. [**Shared Drives**] タブを選択し、ストレージが配置されているドライブを選択してから [**Apply**] をクリックします。 ドライブがすでに選択されている場合は（デフォルトでは C ドライブが選択されています）、チェックボックスをオフにして [**Apply**] をクリックし、ドライブを選択してから [**Apply**] をもう一度クリックします。
3. プロンプトが表示されたら、ログイン資格情報を入力します。
4. 変更を適用すると、Docker が自動的に再起動します。再起動しなかった場合はシステムトレイの Docker アイコンを右クリックし、[**Restart**] を選択します。

コンテナ内で外部ファイルをコピーする

Docker を使用する場合は、一般的には外部ファイルシステム内のディレクトリをコンテナ内のボリュームとしてマウントし、それをコンテナ内のソフトウェアが必要とするすべての外部ファイル用の場所として使用するのが便利です。 例えば、あるボリュームをマウントし、**iris-main** プログラムの **--key** オプションとパスワード変更スクリプトがそれぞれアクセスする外部ディレクトリにある InterSystems IRIS のライセンスキー（**iris.key**）や所定の InterSystems IRIS のパスワードを含むファイルを配置できます（「*Running InterSystems IRIS in Containers*」の「The iris-main Program」と「Changing the InterSystems IRIS Password」を参照してください）。 ただし、Docker for Windows ではファイル操作とアクセス許可の違いにより、マウントされた外部ボリューム上のファイルがコンテナ内のプログラムによって適切に使用されない場合があります。 多くの場合はプログラムでコンテナ内のファイルをコピーしてからそのコピーを使用することで、アクセス許可の問題を解決できます。

例えば、**iris-main --before** オプションはコンテナ内の InterSystems IRIS インスタンスのパスワードを変更するためによく使用されます。以下に例を示します。

```
--before "changePassword.sh /external/password.txt"
```

この方法を使っても Windows で意図したとおりにパスワードを変更できない場合は、以下を試してください。

```
--before "cp /external/password.txt /external/passwordcopied.txt && \
changePassword.sh /external/passwordcopied.txt"
```

名前付きボリュームを使用する

動的に変更されるファイルが多数存在する場合、個別にマウントしてすべてのファイルをコピーできたとしても、コンテナ内に Windows ファイルシステムを直接マウントすると問題が発生する可能性があります。 InterSystems IRIS の場合、これは特にインスタンス固有データを永続的に保管するための Durable %SYS 機能（「Running InterSystems IRIS in Containers」の「Durable %SYS for Persistent Instance Data」を参照してください）とジャーナルファイルの保管の両方に当てはまります。 この問題は名前付きボリュームをマウントすることで解決できます。名前付きボリュームは、システム上のコンテナをホストしている Linux VM のファイルシステムにマウントポイントを持つストレージボリュームです。 VM はファイルシステムベースであるため、このようなボリュームの内容は Docker やシステムがダウンした場合でも VM の他の部分と一緒に Windows ディスクに保存されます。

例えば InterSystems IRIS コンテナを実行中に Durable %SYS を有効化する場合、一般的には外部ボリュームをマウントし、**--env** オプションを使用して **ISCDATADIRECTORY** 環境変数をそのボリュームの場所に設定します。以下に例を示します。

```
docker run ... \
--volume /nethome/pmartinez/irisexternal:/external \
--env ISCDATADIRECTORY=/external/durable/
```

この方法は Docker for Windows では使えません。代わりに、**docker volume create** コマンドで名前付きボリュームを作成し、Durable %SYS ディレクトリをそこに配置する必要があります。また、Linux とは異なり、**ISCDATADIRECTORY** には最上位の Durable %SYS ディレクトリである **/irissys** を指定する必要があります。 したがって、Windows では次のようなオプションになります。

```
docker volume create durable
docker run ... \
--volume durable:/durable \
--env ISCDATADIRECTORY=/durable/irissys/
```

インスタンスのジャーナルファイルにこの方法を使用するには、上記の Durable %SYS の例と同じように名前付きボリュームを作成してマウントし、任意の構成方法（**^JOURNAL** ルーチン、管理ポータルの [Journal Settings] ページ、または **iris.cpf** ファイル）を使用して現在および代替のジャーナルディレクトリを名前付きボリュームの場所に設定します。 現在のジャーナルディレクトリと代替のジャーナルディレクトリを分離するには、それぞれに対応する名前付きボリュームを作成してマウントする必要があります（この方法は***十分に検証されておらず***、ジャーナルファイルが危険にさらされる可能性がありますのでご注意ください）。

VM 内の名前付きボリュームにある Linux ファイルシステムのマウントポイントを検出するには、次のように **docker volume inspect** コマンドを使用します。

```
docker volume inspect durabledata
[
  {
  "CreatedAt": "2018-05-04T12:11:54Z",
  "Driver": "local",
```

```
  "Labels": null,
  "Mountpoint": "/var/lib/docker/volumes/durabledata/data",
  "Name": "durabledata",
  "Options": null,
  "Scope": "local"
  }
]
```

コマンドの比較

上記の内容をすべて考慮し、「*First Look: InterSystems IRIS in Docker Containers*」の「Run and Investigate the InterSystems IRIS-based Container」に記載されている Linux プラットフォームでの実行を想定した最終的な **docker run** コマンドと Docker for Windows を使用した同等の **docker run** コマンドを比較すると次のようになります。

Linux

```
$ docker run --name iris3 --detach --publish 52773:52773 \
  --volume /nethome/pmartinez/irisexternal:/external \
  --env ISCDATADIRECTORY=/external/durable \
  --env ICMSENTINELDIR=/external iris3:test --key /external/iris.key \
  --before "changePassword.sh /external/password.txt"
```

Windows

```
C:\Users\pmartinez>docker volume create durable
C:\Users\pmartinez>docker volume create journals
C:\Users\pmartinez>docker run --name iris3 --detach --publish 52773:52773 \
--volume durable:/durable\
--volume journals:/journals
--env ISCDATADIRECTORY=/durable/irissys \
--env ICMSENTINELDIR=/durable iris3:test --key /external/iris.key \
--before "cp /external/password.txt /external/passwordcopied.txt && \
changePassword.sh /durable/passwordcopied.txt"
```

Docker for Windows での InterSystems 提供コンテナの使用に関する情報をご提供可能な方は、こちらにコメントを追加するか、記事の投稿をお願いします！

#Docker #Microsoft Windows #コンテナ化 #デプロイ #InterSystems IRIS

ソースURL:https://jp.community.intersystems.com/post/docker-windows-%E3%81%A7-intersystems-iris-%E3%82%B3%E3%83%B3%E3%83%86%E3%83%8A%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%99%E3%82%8B