

まずはコンテナを動かしてみよう！ ～コンテナ版IRISで 新機能を試す方法のご紹介～

インターシステムズジャパン株式会社
教育サービス担当
飯島美穂子

2023年12月19日

本ウェビナーの主な目標



- 本ウェビナーでは、コンテナに興味があるけれどまだ使用したことのない方、また、コンテナ版IRISを使用してみたい方を対象に、最短でコンテナを動かすために必要な基本情報をご説明します。
- また、以下内容についての説明と実演を通して、コンテナ利用方法の大枠をつかんでいただくことを目的としています。
 - コンテナ版IRISを動かすための必要な準備
 - Dockerコンテナとイメージの違い
 - コンテナを動かしてみる
 - 自分流にアレンジしたイメージを作る方法

コンテナはお手軽



- ローカルの環境を汚さずに使えます。
 - ちょっとしたテストで他DBや新バージョンのIRISに接続しないといけない！という場合、ローカルにインストールせずコンテナを開始するだけで試すことができます。
- いらなくなったらすぐ消せます（跡形もなく消せる）
 - テスト終了後、アンインストール、ディレクトリ消去など行うことなく、dockerコマンドを利用して使用したコンテナを跡形もなく消せます。
- 他の人に動作確認をお願いするような場合、コンテナベースのサンプルにしておくと環境作成の手順を簡単にすることができます。
 - ソフトウェアや必須ライブラリのインストールなどの初期設定がある場合、その手順を含めることができるため、同じ環境での動作確認のお願いやサンプル配布が簡単です。
- 他の方の意見もぜひご覧ください：[難しいこと抜きでまずはコンテナを触ってみよう](#)
 - [#AWSDevLiveShow](#)



仮想マシンも同じことができるのでは？

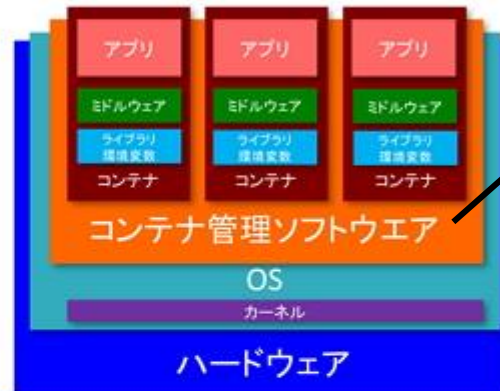
● 違いは？

コンテナ型仮想化

ハイパーバイザー型仮想化



コンテナ型仮想化



仮想マシンは、**環境ごとゲストOSが必要**なのに対して、**コンテナは1つのOSで稼働できる**

隔離されたアプリケーション実行環境を提供する

各仮想マシンに1つのゲストOSが必要

1つのOS上で複数のコンテナを稼働

テストにおいて実行環境の差異を考慮する必要がない

- 開発環境下ではOSやDBのパリテーションが多くツールもさまざまなものが存在
- テスト対象は多岐にわたり、それぞれに対応したテスト環境の準備に手間
- 各環境を準備するための知識を学ぶことが必要

Dockerによってそうした負担から解放され、テスト環境を簡単に構築できるようになり、時間とコストを削減できる

- ✓ 処理のオーバーヘッドが少なくリソース効率が良い
- ✓ 起動・停止が早い
- ✓ デプロイサイズが小さく軽量

コンテナは起動・停止が早い
デプロイサイズが小さく軽量

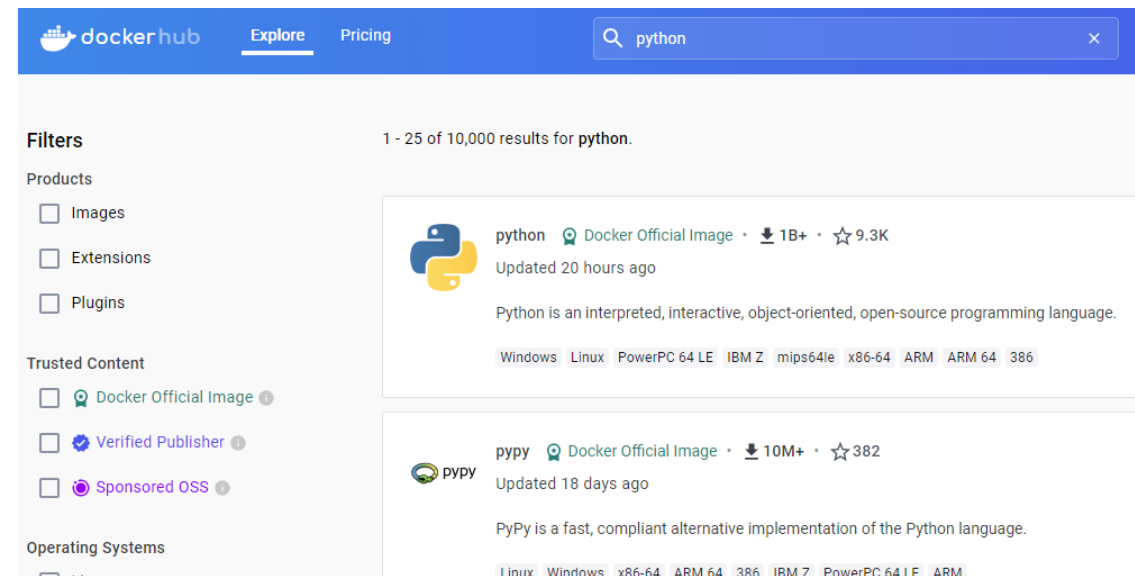
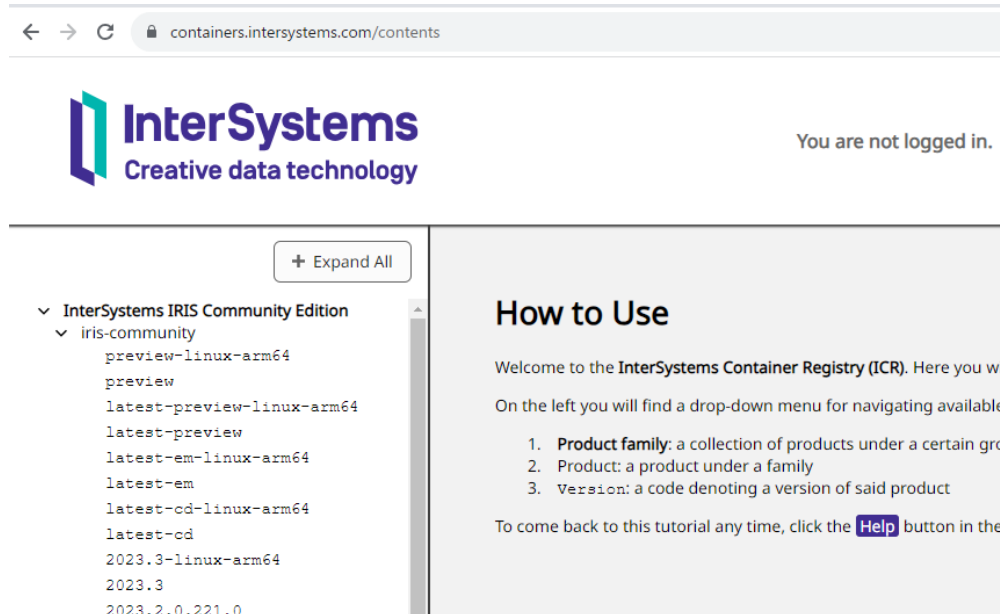


コンテナ版IRISを 動かすために必要な 準備

コンテナ版IRISを動かすために必要な準備



- Linux環境の準備
- Dockerのインストール
- コンテナの元になる**イメージ**をダウンロード
 - コンテナ版IRISのイメージはインターシステムズが用意する「コンテナレジストリ」に用意があります。
 - ちなみに、一般的なイメージはDocker Hubなどにあります。



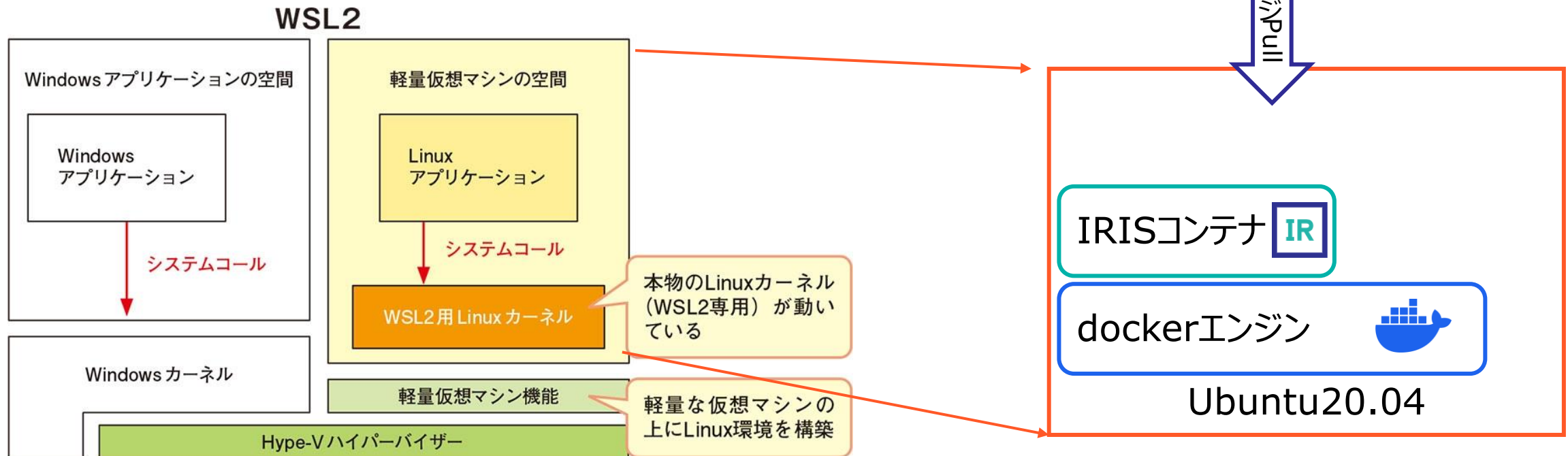


Dockerのコンテナとイメージの違いは？

- Dockerのコンテナは、イメージを使用して生成するアプリケーションなどが動作する場所そのもの
- イメージは、アプリケーションを動かすために必要な環境を保存したもので、持ち運びできる読み取り専用のテンプレート（コンテナの元）

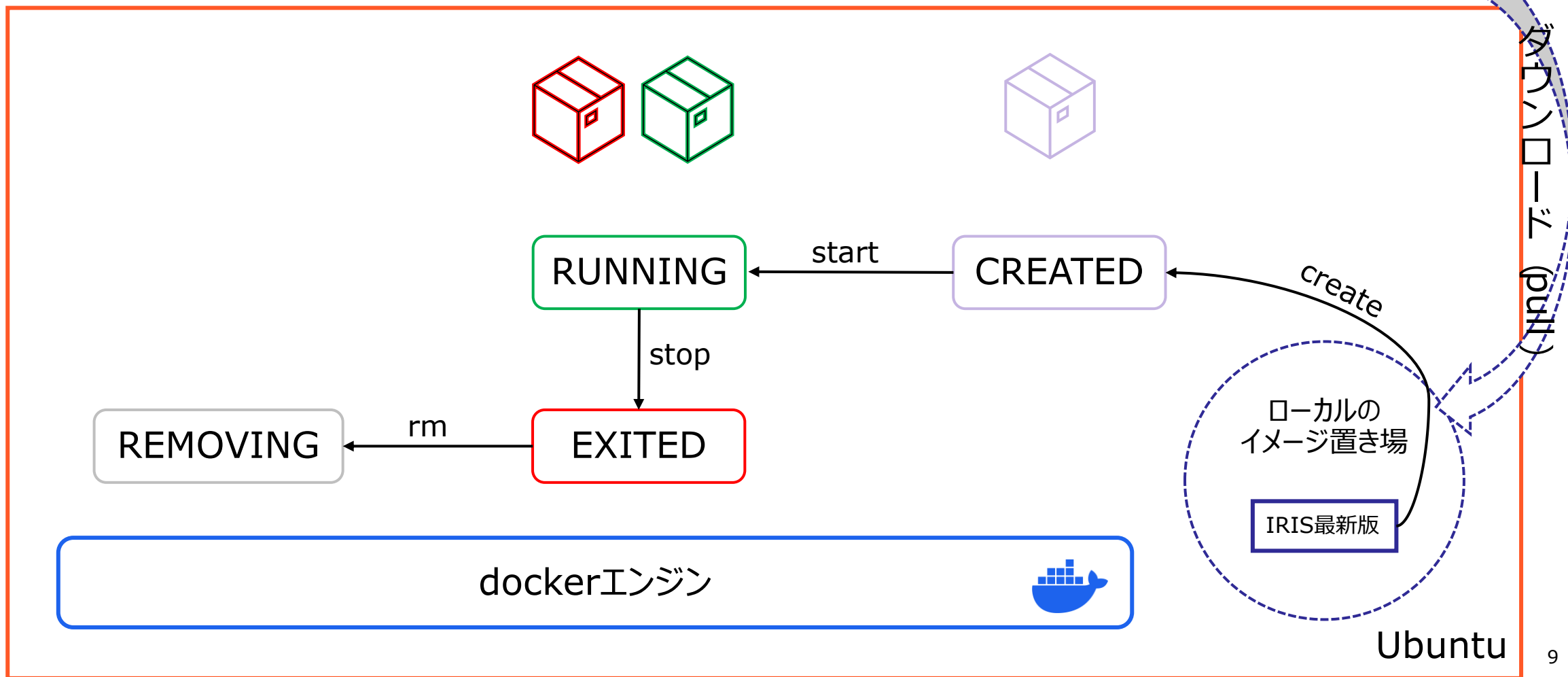
実演環境で考えてみると..

- コンテナの元となる**イメージ**とは？
 - コンテナを開始するために必要な（最低限の）下準備をまとめて保存してある**テンプレート**とお考え下さい。
 - 最新のコンテナ版IRISのイメージには、Ubuntu22.04 + IRIS2023.3 + openjdk "1.8.0_392" + Python3.10.12 が含まれています。



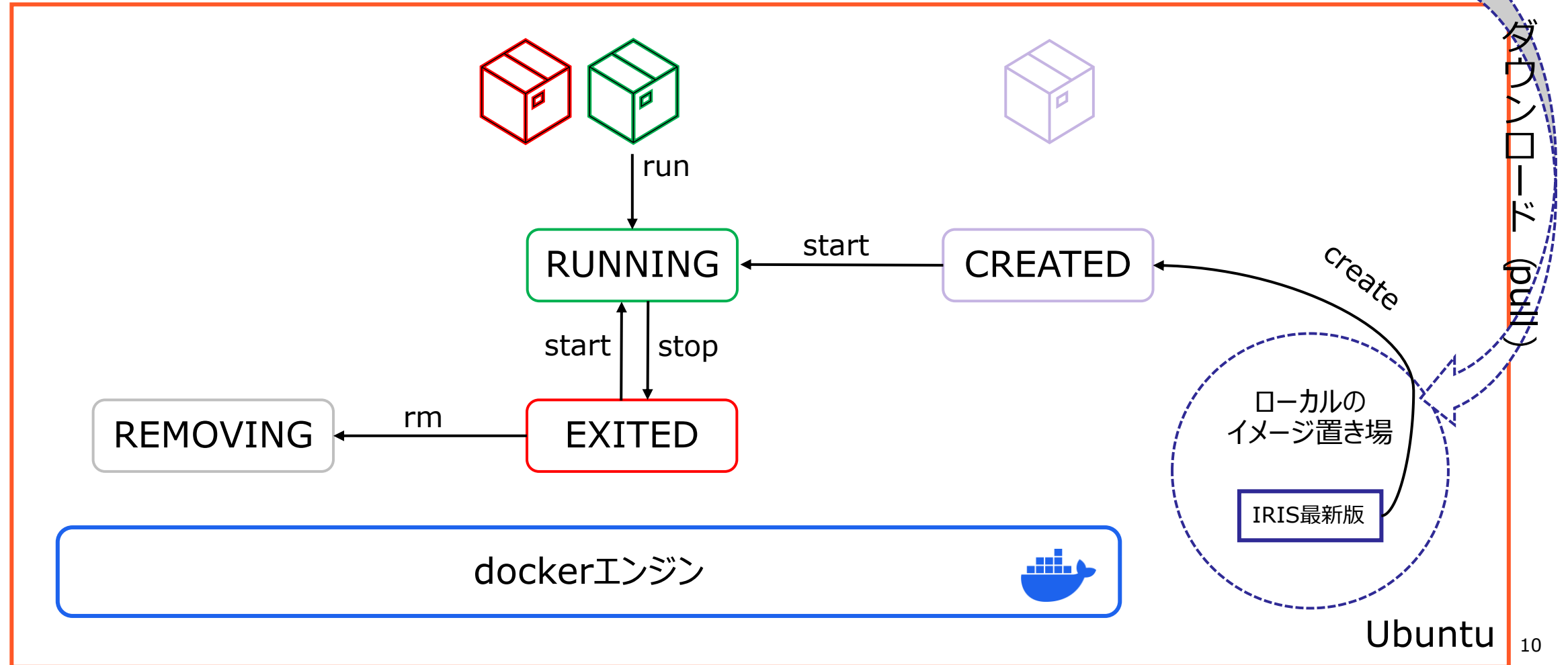
コンテナを動かすには？

- dockerのコンテナ操作コマンドを利用します。
(矢印の黒字部分が操作コマンド名)



コンテナが動くまで（実演の方法）

- docker run コマンドを使います。





では、動かしてみます

コンテナを開始できた後、最新機能である「Foreign Table」の使用例をご覧ください。



自分流にアレンジ したイメージ

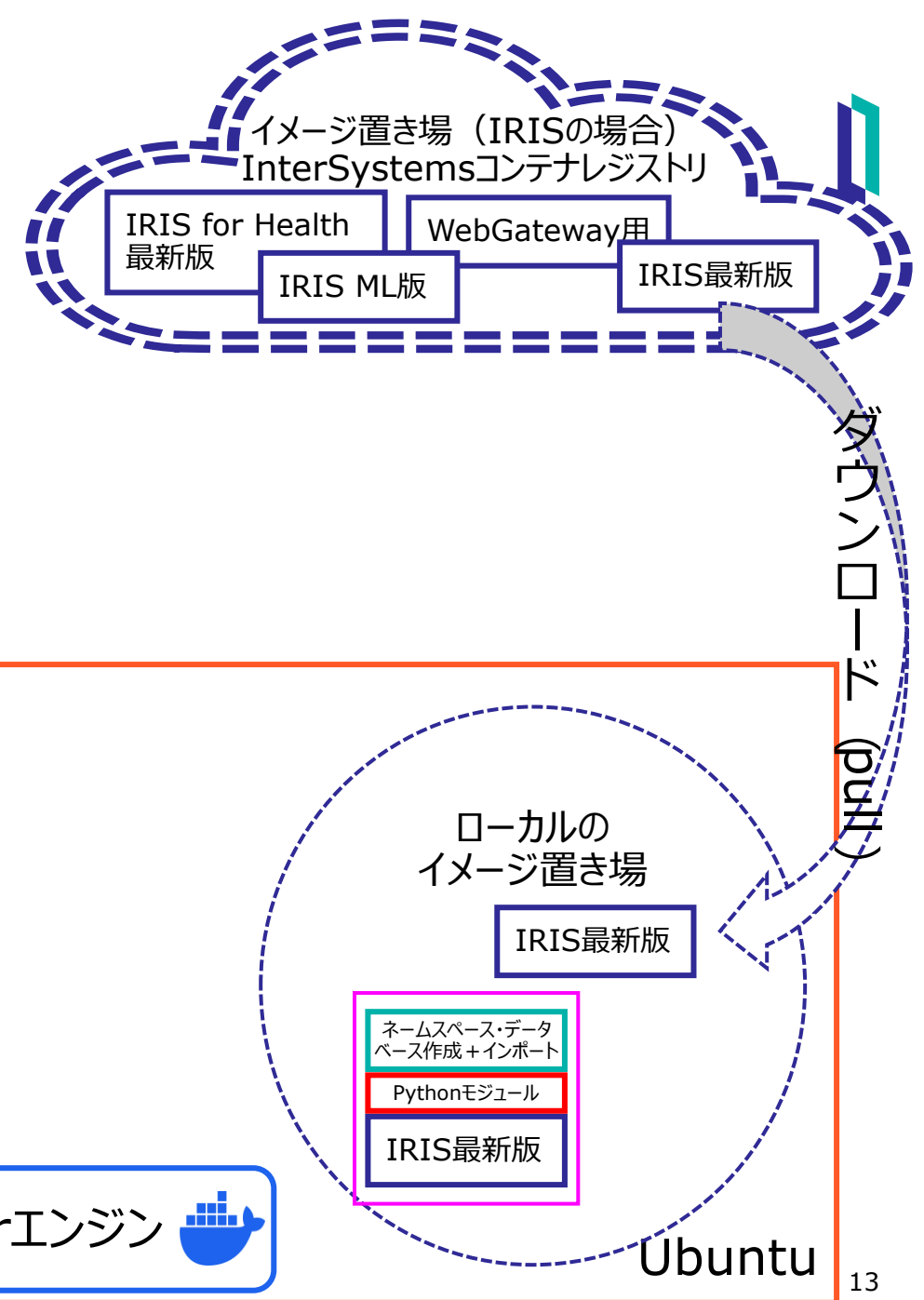
定番の環境があれば、自分流にアレンジしたイメージを事前に用意しておくことができます。

IRISを例に具体例を交えてご紹介します。

オリジナルイメージの作成

- 「あったらいいな」を組み込んだオリジナルイメージを作成することができます。
- 例えば・・・
 - ネームスペース／データベースを作ってアプリケーションで利用するコード類をインポートしておきたい
 - 初期実行があれば実行しておきたい
 - Embedded Pythonで使うモジュールを予めインストールしておきたい
- でも、どうやって作るの？

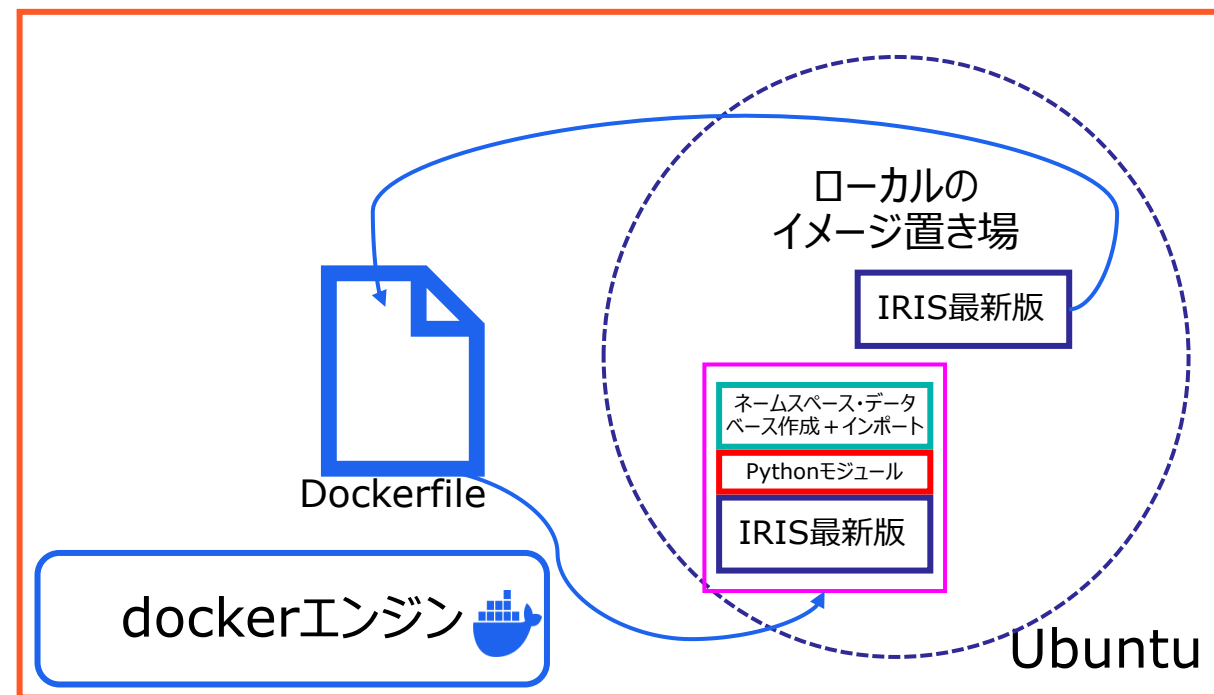
Dockerfileを使います



Dockerfileでできること



- ベースとなるイメージを利用して自分流にアレンジしたコンテナイメージを作成するための各種設定を記述できます。
 - ソフトウェアやライブラリのインストール、IRISなら構成設定を行ったり定義やデータのインポートなど指定できます。
- 作成したイメージを使用してコンテナを開始するだけで必要な初期設定が済んだコンテナを開始できます。
- Dockerfileと関連ファイル一式があれば、別環境で同じコンテナを開始することも簡単に行えます。



```
docker image build . --tag myiris:simple
```



```
.
├── Dockerfile
└── buildsrc
    ├── Installer.cls
    ├── iris.script
    └── requirements.txt
```

実演に使用するDockerfile



Dockerfile > ...

```
1  #イメージのタグはこちら (https://containers.intersystems.com/contents) でご確認ください
2  ARG IMAGE=containers.intersystems.com/intersystems/irishealth-community:latest-cd
3  FROM $IMAGE
4
5  USER root
6  # コンテナ内のワークディレクトリを /opt/try に設定 (後でここにデータベースを作成予定)
7  WORKDIR /opt/try
8  RUN chown ${ISC_PACKAGE_MGRUSER}:${ISC_PACKAGE_IRISGROUP} /opt/try
9
10 USER ${ISC_PACKAGE_MGRUSER}
11
12 # ファイルのコピー
13 COPY buildsrc/ .
14 COPY src src
15
16 # iris.scriptに記載された内容を実行
17 RUN iris start IRIS \
18     && iris session IRIS < iris.script \
19     && iris stop IRIS quietly \
20     ## pipでPythonモジュールインストール
21     && pip install -r requirements.txt
22
```

ベースとなるコンテナ
イメージを指定します。

コンテナ内でデータベースを配置するディレクトリ
を作成し、IRISのプロセスが利用できるように権
限を変更しています。

コンテナ内でIRISに関連する操作をする場合は、
IRIS専用ユーザirisownerに切り替え実行します。

ビルド時に使用するファイルをホストからコピーします。

IRIS開始後、IRISにログインし (iris session IRIS) iris.scriptに記
載されているコマンドを実行し、終了後IRISを停止しています。
(iris.scriptの中身は実演の中でご紹介します)

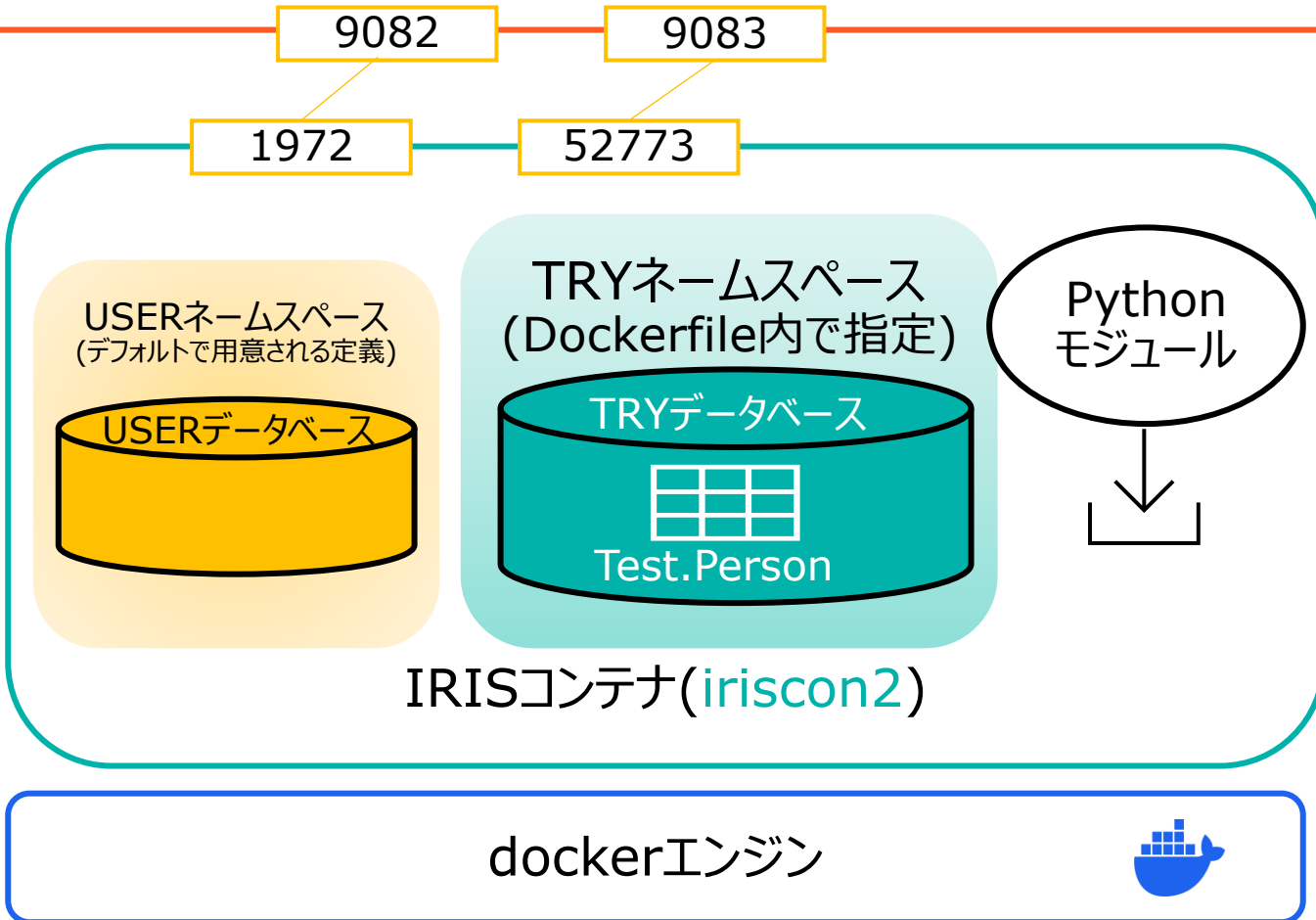
pythonのモジュールをインストールしています。

```
buildsrc > requirements.txt
1  requests
2  beautifulsoup4
```

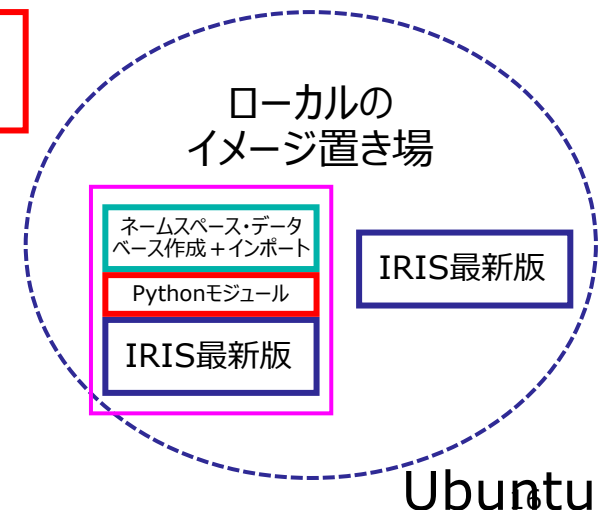
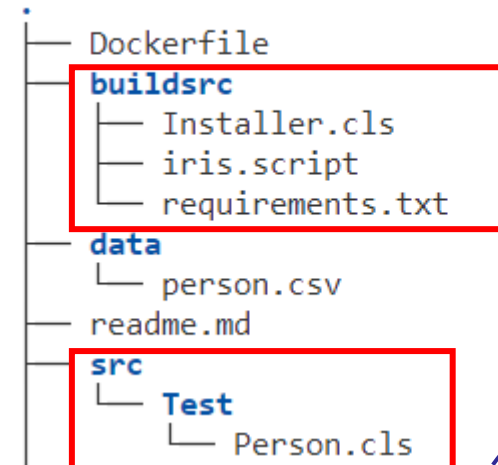
デモで作成する環境

```
docker image build . --tag myiris:simple
```

```
docker run --name iriscon2 -d -p 9082:1972 -p 9083:52773 myiris:simple
```



```
isjedu@JP7430MIIJIMA:~/container/TryContainer$ tree
```





では、動かしてみます

ビルドに使用しているファイルの中身をご覧いただきながら
オリジナルイメージの作成とそのイメージを利用したコンテナ
の開始をご覧いただきます。



コンテナは「ちょっと試したい」にぴったり

- 公開されているイメージそのままでも自分流にアレンジしたイメージでも、コンテナ開始手順は同じ
- 手順が簡単→配布も簡単→試すのも簡単
- **ただし、コンテナを削除する＝コンテナ内に大事なデータを置いておくとすべて失われます！**
 - そんな時のために、`--volume`を使ってホスト側ディレクトリにマウントしておく方法や、IRISの場合は、**永続的な%SYS (Durable %SYS)**という設定を使い、%SYSの環境と、指定ディレクトリ以下に配置したデータベースをコンテナが削除されても永続的に保存できる仕組みを用意しています。
- 開催日程未定ですが、本日ご覧いただいた内容を一緒に試したり、「永続的な %SYS」を使う方法など、「ミニハンズオン」を開催できたらと考えています。
- ご興味ある方、アンケートの「ミニハンズオン参加希望」にチェックをお願いします。
 - 開催が確定しましたらご連絡させていただきます。

本日使用したコマンドなど



- <https://github.com/iijimam/TryContainer>

