

コンテナ利用システムを作る③

…運用環境

ご購入はこちら

土屋 健

プログラムの開発が終わったので，運用環境の構築と運用実験を行います。下記の順でコンテナを使ったシステム運用を体験します。ラズベリー・パイ2台を使います。

1. 運用環境の構築：コンテナ・イメージを準備する
2. 動作確認：コンテナ・イメージを各ラズベリー・

リスト1 コンテナの起動とプログラムのビルド結果

```
tsuyo@rp4:~/CQ $ docker run -it -v `pwd`/
    sharedVol:/sharedVol rust_dev:1.0
root@8calc3c34341:/# cd /sharedVol/webapi/
root@8calc3c34341:/sharedVol/webapi# cargo build
    Updating crates.io index
    Downloaded async-stream-impl v0.3.5
    省略
    Compiling webapi v0.1.0 (/sharedVol/webapi)
    Finished dev [unoptimized + debuginfo]
    target(s) in 8m 45s
root@8calc3c34341:/sharedVol/webapi#
```

リスト2 プログラムの配置結果

```
root@8calc3c34341:/sharedVol/webapi# mkdir -p /
    root/CQ/
root@8calc3c34341:/sharedVol/webapi# cp -p target/
    debug/webapi /root/CQ/
root@8calc3c34341:/sharedVol/webapi# ls -l /root/CQ/
total 75220
-rwxr-xr-x 1 root root 77059232 Apr 10 09:55 webapi
root@8calc3c34341:/sharedVol/webapi#
```

リスト3 データ・サーバのコンテナ・イメージの登録結果

```
root@8calc3c34341:/sharedVol/webapi# exit
tsuyo@rp4:~/CQ $ docker ps -a
CONTAINER ID   IMAGE          COMMAND         CREATED        STATUS          PORTS          NAMES
8calc3c34341   rust_dev:1.0   "bash"         2 hours ago   Exited (0) 3   seconds ago           practical_borg
2acd14d5e179   arm64v8/debian "bash"         9 days ago    Exited (0) 3   days ago              nostalgic_volhard
tsuyo@rp4:~/CQ $
tsuyo@rp4:~/CQ $ docker commit 8calc3c34341 datasrv
sha256:47273aeef4cafdf6666c489152ad6ba83f7ead3fffb2889769539aa83aa2c93b7
tsuyo@rp4:~/CQ $
tsuyo@rp4:~/CQ $ docker images
REPOSITORY    TAG          IMAGE ID       CREATED        SIZE
datasrv       latest      47273aeef4ca  34 seconds ago 2.61GB
rust_dev      1.0        a0f8cae68ed8  9 days ago    2.47GB
arm64v8/debian latest      f50f6cc64f28  4 weeks ago   139MB
arm64v8/ubuntu latest      2b7cc08dcdbb  6 weeks ago   69.2MB
amd64/ubuntu  latest      ca2b0f26964c  6 weeks ago   77.9MB
tsuyo@rp4:~/CQ $
```

データ・サーバをビルドしたコンテナを確認。対象はIMAGEがrust_dev:1.0のもの

datasrvという名前でコンテナ・イメージを登録

コンテナ・イメージを一覧して、datasrvが登録されていることを確認

パイに配置し，データ・サーバ・コンテナおよびセンサ・クライアント・コンテナを起動して，センサ・データの蓄積を行う

3. コンテナの入れ替え：運用を続けながらデータ・サーバの更新を行う

ステップ1：運用環境の構築

● データ・サーバのコンテナ・イメージの作成

まずは1台目のラズベリー・パイにて，データ・サーバのコンテナ・イメージを作成します。

リスト1のように，コンテナを起動し作成したRustプロジェクトをビルドします。ビルドされたプログラムを配置します(リスト2)。配置場所はどこでも構いませんが，ここでは/root/CQ/webapiとして配置します。

コンテナを停止させ，データ・サーバのイメージをdatasrvという名前で登録します(リスト3)。

● センサ・クライアントのコンテナ・イメージ

ここは1台目のラズベリー・パイで作業します。ビルド対象が異なるだけで，コンテナの起動からビルドまではほぼ同じ手順で実施します(リスト4)。ビルドされたプログラムは，/root/CQ/sensorとして