

手順概要

1. Dockerファイルを作成
- ↓
2. コンテナイメージの作成
- ↓
3. コンテナの動作確認
- ↓
4. イメージをリポジトリにアップ
- ↓
5. 残作業

1. Dockerファイルを作成

- ①作業ディレクトリに移動してDockerfileを作成
- ②サンプル用のHTMLページを作成

```
$ cd /home/chopper/docker/  
$ touch testpage.html  
$ touch Dockerfile
```

Dockerfile

```
FROM ubuntu  
  
RUN apt-get update && \  
    apt-get install -y apache2  
  
COPY testpage.html /var/www/html/  
  
EXPOSE 80  
CMD ["apachectl", "-D", "FOREGROUND"]
```

2. コンテナイメージの作成

- ①Dockerfileからイメージを作成 : `docker build -t <作成するイメージ名:タグ名> <材料フォルダのパス>`

```
docker build -t test-apache:ver01 .
```

- ②ビルドしたイメージを確認 : `docker image ls`

```
$ docker image ls  
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE
```

test-apache ver01 159394ebdbde 40 minutes ago 236MB

③イメージからコンテナを作成 : docker run

```
$ docker run --name app0001 -d -p 8081:80 test-apache:ver01
99e634a617d1bd90fa7c6c4561b3b11bf2bb2b607ef2a35244f61aa142f34606
```

```
$ docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|-------------------|--------------------------|--------------------|-----------------|
| 99e634a617d1 | test-apache:ver01 | "apachectl -D FOREGR..." | About a minute ago | Up About a minu |

- ・ tag名を省略した場合、デフォルトで"latest"が補完される

3. コンテナの動作確認

```
$ curl -I http://localhost:8081
HTTP/1.1 200 OK
～略～
```

4. イメージをリポジトリにアップ

イメージのアップ方法は二つあり、今回は①の方法で実施。

- ①git cloneで Dockerfileを含むファイル群をCodeCommit等のgitリポジトリにアップする
- ②docker push で コンテナイメージをECR等のDockerリポジトリにアップする

5. 残作業

①Dockerを停止する : docker container stop

```
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
99e634a617d1   test-apache:ver01  "apachectl -D FOREGR..."  2 hours ago   Up 2 hours   0.0.0.0:80->8081
$
$ docker container stop app0001
app0001
$
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
```

②コンテナを削除 : docker image rm

```
$ docker container rm app0001
app0001
$
$ docker container ls
CONTAINER ID   IMAGE          COMMAND         CREATED        STATUS        PORTS        NAMES
$
```

③コンテナイメージを削除 : docker image rm

```
$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
test-apache   ver01     159394ebbdbe   3 hours ago   236MB
$
$ docker image rm test-apache:ver01
Untagged: test-apache:ver01
Deleted: sha256:159394ebbdbe1e90689eafc88c3f32d5df54062714b32af0b46854fdae5528
$
$
$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
$
```

④不要キャッシュを削除 : docker system df、docker builder prune

```
$ docker system df
TYPE                TOTAL        ACTIVE        SIZE        RECLAIMABLE
Images              0            0            0B          0B
Containers          0            0            0B          0B
Local Volumes       11           0            1.59GB      1.59GB (100%)
Build Cache         7            0            316.3MB     316.3MB

$ docker builder prune
WARNING! This will remove all dangling build cache. Are you sure you want to continue? [y/N] y
ID                                RECLAIMABLE    SIZE        LAST ACCESSED
w4jinffc1iyyv4xoripxndoty         true           158.1MB      4 hours ago
yc6hdsK9m53o0bi4n14ayyrg2         true           157B         2 hours ago
2mfszr80kfo29qa3x2dr3xw06*        true           72B          2 hours ago
nx1enrqmmy35aqdc3cdaug7dv*        true           157B         2 hours ago
a56nka3bne6xr64vs2ltrt0wr*        true           0B           2 hours ago
l91y19je04yx4zmzjtceacw5b         true           158.1MB      3 hours ago
orx2z97zjz97u36n1fxg06i69         true           0B           3 hours ago
Total: 316.3MB

$ docker system df
TYPE                TOTAL        ACTIVE        SIZE        RECLAIMABLE
Images              0            0            0B          0B
Containers          0            0            0B          0B
Local Volumes       11           0            1.59GB      1.59GB (100%)
Build Cache         0            0            0B          0B
$
```