

```
(base) tsuchiya@macbook mydata_server % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(base) tsuchiya@macbook mydata_server % docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(base) tsuchiya@macbook mydata_server % docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
(base) tsuchiya@macbook mydata_server % docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
a9058e77ae3c   bridge   bridge    local
9e9b13b58c06   host     host      local
57be5d488354   none     null      local
(base) tsuchiya@macbook mydata_server %
リスト 1
```

```
% docker network create mynetwork
bfc045b469a4a893a9458b67d839682167f308eefec04db9d34b3848f95579cb
% docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
a9058e77ae3c   bridge   bridge    local
9e9b13b58c06   host     host      local
bfc045b469a4   mynetwork bridge    local
57be5d488354   none     null      local
%
リスト 2
```

```
docker run --name influxdb --network mynetwork -d -p 8086:8086 ¥
-e DOCKER_INFLUXDB_INIT_MODE=setup ¥
-e DOCKER_INFLUXDB_INIT_USERNAME=admin ¥
-e DOCKER_INFLUXDB_INIT_PASSWORD=admin001 ¥
-e DOCKER_INFLUXDB_INIT_ORG=MyDataServer ¥
-e DOCKER_INFLUXDB_INIT_BUCKET=my-bucket ¥
-e DOCKER_INFLUXDB_INIT_ADMIN_TOKEN="jG99-jU4YqMZixUN3IGoBCdQ7iAbekCXPkaM2SEvHXt0IXEQKMDkJppaiXYn66bnsOuaEDZ323kIp3EYIHw-zg==" ¥
influxdb:2.0
```

--name オプションでコンテナのお名前を設定します。他のコンテナからはこの名前を指定して通信できます。

--network オプションで接続するネットワークを指定します。今回は、データ・サーバ関連コンテナのみで使用する閉じたネットワーク（mynetwork）に接続します。

-p オプションは、ホストのポート番号とコンテナのポート番号をマッピングする指定で、ホストの 8086 ポートにアクセスしたらコンテナのポート 8086 にデータが送られるようになります。

-e オプションはコンテナの環境変数設定で、以下の値が設定されます。

```
+-----+-----+
| 項目      | 設定値                |
+-----+-----+
| Organization | MyDataServer          |
| API トークン | jG9- ??? YIHw-zg== |
| ユーザ      | admin                  |
| パスワード  | admin001              |
+-----+-----+
```

リスト 3

```
% docker run --name influxdb --network mynetwork -d -p 8086:8086 ¥
-e DOCKER_INFLUXDB_INIT_MODE=setup ¥
-e DOCKER_INFLUXDB_INIT_USERNAME=admin ¥
-e DOCKER_INFLUXDB_INIT_PASSWORD=admin001 ¥
-e DOCKER_INFLUXDB_INIT_ORG=MyDataServer ¥
-e DOCKER_INFLUXDB_INIT_BUCKET=my-bucket ¥
-e DOCKER_INFLUXDB_INIT_ADMIN_TOKEN="jG99-jU4YqMZixUN3IGoBCdQ7iAbekCXPkaM2SEvHXt0IXEQKMDkJppaiXYn66bnsOuaEDZ323kIp3EYIHw-zg==" ¥
influxdb:2.0
Unable to find image 'influxdb:2.0' locally
2.0: Pulling from library/influxdb
982cba7e471c: Pull complete
b02d86f59850: Pull complete
```

```
8b047e8f2e47: Pull complete
dc44b41d3b54: Pull complete
bcd83ba9d903: Pull complete
2978b95b0d4a: Pull complete
d8202613e12d: Pull complete
08d3416f8db9: Pull complete
6315db684033: Pull complete
Digest: sha256:6dd9d3894e92c31ba25c59736d00871b46b960e40e65658bd8bd62825a53f88c
```

Status: Downloaded newer image for influxdb:2.0

```
590ddd82dd48ce7147d9a4dc194823a5cb7dad42a2a3e530a75fac9f8bce3a51
```

%

リスト 4

```
docker run --name MySQL --network mynetwork -e MYSQL_ROOT_PASSWORD='MyServer@001' -e MYSQL_DATABASE=mydataserver -d mysql:8.0.29
```

リスト 5

```
% docker run --name MySQL --network mynetwork -e MYSQL_ROOT_PASSWORD='MyServer@001' -e MYSQL_DATABASE=mydataserver -d mysql:8.0.29
```

Unable to find image 'mysql:8.0.29' locally

8.0.29: Pulling from library/mysql

```
e54b73e95ef3: Pull complete
```

```
327840d38cb2: Pull complete
```

```
642077275f5f: Pull complete
```

```
e077469d560d: Pull complete
```

```
cbf214d981a6: Pull complete
```

```
7d1cc1ea1b3d: Pull complete
```

```
d48f3c15cb80: Pull complete
```

```
94c3d7b2c9ae: Pull complete
```

```
f6cfbf240ed7: Pull complete
```

```
e12b159b2a12: Pull complete
```

```
4e93c6fd777f: Pull complete
```

```
Digest: sha256:152cf187a3efc56afb0b3877b4d21e231d1d6eb828ca9221056590b0ac834c75
```

Status: Downloaded newer image for mysql:8.0.29

```
a65a47831496d29f4134583601fe6660cda55676ecc047b316c9be12cbd63902
```

%

リスト 6

```
% docker exec -i MySQL bash -c 'mysql -u root -pMyServer@001 --protocol=tcp' < SETUP/DDL
```

```
mysql: [Warning] Using a password on the command line interface can be insecure.
```

%

リスト 7

```
docker run --rm --name ethereum --network mynetwork ¥
```

```
-v `pwd`/ethereum_data:/DATA:rw ¥
```

```
ethereum/client-go ¥
```

```
--datadir /DATA/ init /DATA/genblock.json
```

-v オプションは、ホストのディレクトリとコンテナのディレクトリをマッピングする指定で、\$HOME/CQ/mydata_server/ethereum_data の領域がコンテナでは/DATAとしてアクセスできるよう設定しています。

--rm オプションは処理が終わったらコンテナを削除する指示です。コンテナを削除しても外部にあるデータ領域は残ったままです。

リスト 8

```
docker run -d --name ethereum --network mynetwork -p 8545:8545 ¥
-v `pwd`/ethereum_data:/DATA:rw ¥
ethereum/client-go ¥
--networkid "504" --nodiscover --datadir /DATA/ ¥
--http --http.addr "0.0.0.0" --http.port "8545" --http.corsdomain "*" ¥
--http.api "eth,net,web3,personal,admin,rpc,txpool,miner" ¥
--maxpeers 0 --allow-insecure-unlock
リスト 9
```

```
% mkdir ethereum_data
% cp -p SETUP/genblock.json ethereum_data
% docker run --rm --name ethereum --network mynetwork ¥
-v `pwd`/ethereum_data:/DATA:rw ¥
ethereum/client-go ¥
--datadir /DATA/ init /DATA/genblock.json
```

Unable to find image 'ethereum/client-go:latest' locally

latest: Pulling from ethereum/client-go

213ec9aee27d: Pull complete

9bd60ff9c665: Pull complete

882970670672: Pull complete

Digest: sha256:58dbd5bbec352256b4898ef56b904e24c7c976664d34a1a19f7b0f4e1bded3c2

Status: Downloaded newer image for ethereum/client-go:latest

```
INFO [09-19|03:36:08.398] Maximum peer count           ETH=50 LES=0 total=50
INFO [09-19|03:36:08.401] Smartcard socket not found, disabling err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [09-19|03:36:08.500] Set global gas cap             cap=50,000,000
INFO [09-19|03:36:08.508] Allocated cache and file handles database=/DATA/geth/chaindata cache=16.00MiB handles=16
INFO [09-19|03:36:08.853] Opened ancient database       database=/DATA/geth/chaindata/ancient/chain readonly=false
INFO [09-19|03:36:08.854] Writing custom genesis block
INFO [09-19|03:36:08.855] Persisted trie from memory database nodes=0 size=0.00B time="5.872 μs" gcnodes=0 gcsz=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [09-19|03:36:08.919] Successfully wrote genesis state database=chaindata hash=d118bd..7a409f
INFO [09-19|03:36:08.919] Allocated cache and file handles database=/DATA/geth/lightchaindata cache=16.00MiB handles=16
INFO [09-19|03:36:09.206] Opened ancient database       database=/DATA/geth/lightchaindata/ancient/chain readonly=false
INFO [09-19|03:36:09.206] Writing custom genesis block
INFO [09-19|03:36:09.207] Persisted trie from memory database nodes=0 size=0.00B time="4.362 μs" gcnodes=0 gcsz=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [09-19|03:36:09.245] Successfully wrote genesis state database=lightchaindata hash=d118bd..7a409f
```

```
% docker run -d --name ethereum --network mynetwork -p 8545:8545 ¥
-v `pwd`/ethereum_data:/DATA:rw ¥
ethereum/client-go ¥
--networkid "504" --nodiscover --datadir /DATA/ ¥
--http --http.addr "0.0.0.0" --http.port "8545" --http.corsdomain "*" ¥
--http.api "eth,net,web3,personal,admin,rpc,txpool,miner" ¥
--maxpeers 0 --allow-insecure-unlock
9acd39ca9ad117e4018bd8e6aa8d7022c4bc54e95889fd8b71fb7b7a44cd10c2
%
リスト 10
```

```
$ curl https://nodejs.org/dist/v16.17.0/node-v16.17.0-darwin-x64.tar.gz -o node-v16.17.0-darwin-x64.tar.gz
```

```
$ tar xzf node-v16.17.0-darwin-x64.tar.gz
```

```
$ export PATH=`pwd`/node-v16.17.0-darwin-x64/bin:$PATH
```

```
$ npm install keythereum
```

```
$ node
```

```
> let ac = "0x6015375f653d7b6b95ef36b0068f7425849256b8"
```

```
> let pass = "dataprovder"
```

```
> let ke = require("keythereum");
```

```
> let ks = "/Users/tsuchiya/CQ/mydata_server/ethereum_data"
```

```
> let ko = ke.importFromFile(ac, ks)
```

```
> let pk = ke.recover(pass, ko)
```

```
> pk.toString('hex')
```

```
66e069d992f1be7653cb375dc33adb2dc6df47e358e3254a91cf3e9b0d963a13
```

```
リスト 11
```

```
docker run --name MinIO --network mynetwork -p 9001:9001 quay.io/minio/minio server /data --console-address ":9001"
```

リスト 12

```
% docker run -d --name MinIO --network mynetwork -p 9001:9001 quay.io/minio/minio server /data --console-address ":9001"
```

Unable to find image 'quay.io/minio/minio:latest' locally

latest: Pulling from minio/minio

d5d2e87c6892: Pull complete

008dba906bf6: Pull complete

7f9fba2cd1df: Pull complete

3d04d9270e81: Pull complete

9e0f79aed5ef: Pull complete

060ee8abbd70: Pull complete

55f097200265: Pull complete

Digest: sha256:d1e2a053353685df57bfef981d9f47d6de2c065a705f7dad3c1b6f1be6b15356

Status: Downloaded newer image for quay.io/minio/minio:latest

db68be96ec3eca9f6005be17faafcb5d2d1ee60ad5e0208bc6d44169e06a133e

%

リスト 13

```
use mydataserver;
```

```
INSERT into ds_influxdb (id,name,address,port,token,organization) values ("InfluxDB","My
```

```
InfluxDB","influxdb",8086,"jG99-jU4YqMZixUN3IGoBCdQ7iAbekCXPkaM2SEvHXt0IXEQKMDkJppaiXYn6¥
```

```
6bnsOuaEDZ323k1p3EYIHW-zg==","MyDataServer");
```

```
INSERT into ds_storage (id,name,address,port,access_key_id,secret_access_key) values ("MinIO","Storage(MinIO)","minio",9000,"minioadmin","minioadmin");
```

```
INSERT into ds_ethereum (id,name,address,port,account,password,private_key) values ("Ethereum","Etherem private
```

```
net","ethereum",8545,"0x6015375f653d7b6b95ef36b0068f7425849256b8"¥
```

```
,"dataproducer","66e069d992f1be7653cb375dc33adb2dc6df47e358e3254a91cf3e9b0d963a13");
```

リスト 14

```
% docker exec -i MySQL bash -c 'mysql -u root -pMyServer@001 --protocol=tcp' < DOCKER/config.ddl
```

```
mysql: [Warning] Using a password on the command line interface can be insecure.
```

%

リスト 15

```
1 FROM python:3.7
```

```
2 RUN pip install fastapi uvicorn influxdb-client SQLAlchemy==1.4.39 mysqlclient web3 boto3 python-multipart
```

```
3
```

```
4 COPY ./mydata_server/
```

```
5 COPY ./LIB/config.json.cont /mydata_server/LIB/config.json
```

```
6 COPY ./SETUP/config.sh.cont /mydata_server/config.sh
```

```
7
```

```
8 ENV PYTHONPATH="/mydata_server/LIB:/mydata_server/APP:$PYTHONPATH" ¥
```

```
9 MYDATASERVER_APP_CONFIG_PATH=/mydata_server/APP ¥
```

```
10 MYDATASERVER_LIB_CONFIG_PATH=/mydata_server/LIB ¥
```

```
11 MYDATASERVER_TOP=/mydata_server/
```

```
12
```

```
13 CMD ["python", "/mydata_server/mydata_server.py"]
```

リスト 16

```
% docker build -f DOCKER/dataserver.dockerfile -t dataserver .
```

```
[+] Building 5.6s (10/10) FINISHED
```

```

=> [internal] load build definition from dataserver.dockerfile          0.3s
=> => transferring dockerfile: 47B                                     0.1s
=> [internal] load .dockerignore                                       0.1s
=> => transferring context: 2B                                         0.0s
=> [internal] load metadata for docker.io/library/python:3.7          3.5s
=> [1/5] FROM docker.io/library/python:3.7@sha256:2341ac5eadd71f1e6481afe854af572f5ec1b78fa3aea2293dba65942108e663 0.0s
=> [internal] load build context                                       0.5s
=> => transferring context: 140.21kB                                    0.5s
=> CACHED [2/5] RUN pip install fastapi uvicorn influxdb-client SQLAlchemy==1.4.39 mysqlclient web3 boto3 python-multipart 0.0s
=> [3/5] COPY ./mydata_server/                                         0.5s
=> [4/5] COPY ./LIB/config.json.cont /mydata_server/LIB/config.json  0.1s
=> [5/5] COPY ./SETUP/config.sh.cont /mydata_server/config.sh        0.1s
=> exporting to image                                                 0.2s
=> => exporting layers                                               0.2s
=> => writing image sha256:556fbf6ead22c818b99f2378304e00452303765ba955cbf45ea9410e0389aabb 0.0s
=> => naming to docker.io/library/dataserver                          0.0s

```

```
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

```
%
```

```
リスト 17
```

```
% docker run -d --name mydataserver --network mynetwork -p 8000:8000 dataserver
```

```
1c2c4326125e6e67c384ac958c4a32c612a8aa43e0409004ed90cfedd927d312
```

```
%
```

```
リスト 18
```

```
% docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
1c2c4326125e	dataserver	"python /mydata_serv..."	58 seconds ago	Up 51 seconds	0.0.0.0:8000->8000/tcp	mydataserver
db68be96ec3e	quay.io/minio/minio	"/usr/bin/docker-ent..."	14 minutes ago	Up 14 minutes	9000/tcp, 0.0.0.0:9001->9001/tcp	MinIO
9acd39ca9ad1	ethereum/client-go	"geth --networkid 50..."	53 minutes ago	Up 53 minutes	8546/tcp, 0.0.0.0:8545->8545/tcp, 30303/tcp, 30303/udp	ethereum
a65a47831496	mysql:8.0.29	"docker-entrypoint.s..."	About an hour ago	Up About an hour	3306/tcp, 33060/tcp	MySQL
590ddd82dd48	influxdb:2.0	"/entrypoint.sh infl..."	About an hour ago	Up About an hour	0.0.0.0:8086->8086/tcp	influxdb

```
% docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dataserver	latest	556fbf6ead22	4 minutes ago	1.13GB
ethereum/client-go	latest	ee278e08b7a2	33 hours ago	50.2MB
quay.io/minio/minio	latest	9ff24ad6f42a	44 hours ago	227MB
mysql	8.0.29	33037edcac9b	2 months ago	444MB
influxdb	2.0	6e977ef79a26	3 months ago	342MB

```
%
```

```
リスト 19
```