

1. 「コンテナ構築概要」を参考にして、Windows 上、AWS Ubuntu 上に Debian コンテナを構築してください。

コンテナ構築概要

- Debian コンテナはバージョン10以上
- Debian コンテナに JDK と gcc をインストールする(Dockerfile で定義してください)
openjdk-11-jdk をインストールしてください。

ローカル上のフォルダ構成

-/Docker/debian_bench/

に作成した compose.yaml、Dockerfile を配置する

-/Docker/debian_bench/src

添付されたベンチマーク用ソースプログラムを配置する

BenchC.c、BenchJava.java

-/Docker/debian_bench/src と Debian 内の/usr/local/src のマウントを取る

📁 コンテナ構築過程、添付された C 言語、Java のベンチマーク用のプログラムのベンチマーク結果を画面キャプチャして提出してください。

compose.yaml、Dockerfile を提出してください。

```
C:\Users\> docker compose up -d
[+] Building 121.0s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 419B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/debian:11.3
=> CACHED [1/4] FROM docker.io/library/debian:11.3
=> [internal] load build context
=> => transferring context: 1.64kB
=> [2/4] COPY ./src/BenchC.c /usr/local/src
=> [3/4] COPY ./src/BenchJava.java /usr/local/src
=> [4/4] RUN apt update && apt-get -y install openjdk-11-jdk && apt-get -y install vim && apt -y install gcc
=> exporting to image
=> exporting layers
=> => writing image sha256:72d9bc1264810cbaf2cd0706b0993adec41ef9d83f4a7a7b86c89ee843be5c8c
=> => naming to docker.io/library/debian_bench-debian
[+] Running 2/2
 - Network debian_bench_default Created
 - Container debian_bench-debian-1 Started
```

```
root@3b2911dbe518:/usr/local/src# gcc -o BenchC BenchC.c
root@3b2911dbe518:/usr/local/src#
root@3b2911dbe518:/usr/local/src#
root@3b2911dbe518:/usr/local/src# ./BenchC
C      -> 18.9387 secs.
root@3b2911dbe518:/usr/local/src#
```

```
root@3b2911dbe518:/usr/local/src# javac BenchJava.java
root@3b2911dbe518:/usr/local/src#
root@3b2911dbe518:/usr/local/src#
root@3b2911dbe518:/usr/local/src# java BenchJava
Java  -> 6.3960 secs.
root@3b2911dbe518:/usr/local/src#
```

Docker 実習 14

```
ubuntu@ip-172-31-3-122:~/Docker/debian_bench$ docker compose up -d
[+] Building 87.7s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 419B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/debian:11.3
=> CACHED [1/4] FROM docker.io/library/debian:11.3@sha256:859ea45db307402ee024b153c7a63ad4888eb4751921abbef68679fc73c4c739
=> [internal] load build context
=> => transferring context: 1.64kB
=> [2/4] COPY ./src/BenchC.c /usr/local/src
=> [3/4] COPY ./src/BenchJava.java /usr/local/src
=> [4/4] RUN apt update && apt-get -y install openjdk-11-jdk && apt-get -y install vim && apt -y install gcc
=> exporting to image
=> => exporting layers
=> => writing image sha256:84445f4a399acc4dbef0705170c3d86bedff5768a58a98a910fc464e25cb1b56
=> => naming to docker.io/library/debian_bench-debian
[+] Running 2/2
? Network debian_bench_default Created
? Container debian_bench-debian-1 Started
ubuntu@ip-172-31-3-122:~/Docker/debian_bench$
```

```
root@3a511b10ffb8:/usr/local/src# gcc -o BenchC BenchC.c
root@3a511b10ffb8:/usr/local/src#
root@3a511b10ffb8:/usr/local/src#
root@3a511b10ffb8:/usr/local/src#
root@3a511b10ffb8:/usr/local/src# .BenchC
bash: .BenchC: command not found
root@3a511b10ffb8:/usr/local/src# ./BenchC
C      -> 19.9950 secs.
root@3a511b10ffb8:/usr/local/src#
```

```
root@3a511b10ffb8:/usr/local/src# javac BenchJava.java
root@3a511b10ffb8:/usr/local/src#
root@3a511b10ffb8:/usr/local/src#
root@3a511b10ffb8:/usr/local/src# java BenchJava
Java   -> 5.7190 secs.
root@3a511b10ffb8:/usr/local/src#
```


