

Сделал следующее:

Сделал следующее:

Создание минимальной root файловой системы с помощью debootstrap (для Ubuntu/Debian):

Директория

```
mkdir rootfs
```

Минимальная система

```
sudo debootstrap --variant=minbase focal rootfs http://archive.ubuntu.com/ubuntu/
```

Упаковка

```
sudo tar -C rootfs -cf rootfs.tar .
```

Установка capabilities для java

Узнаем где джава находится (фактический исполняемый файл java)

```
readlink -f $(which java)
```

Выставляем capabilities:

```
sudo setcap cap_sys_admin,cap_setgid,cap_setuid+ep /usr/lib/jvm/java-11-openjdk-amd64/bin/java
```

Запуск в нашем минидокере

Проверяем что .jar файл нашего мини-докера и rootfs.tar находятся в одной директории:

```

root@cv4222321: /# cd
total 129608
drwxr-xr-x 21 root root 4096 Oct 31 18:58 .
drwxr-xr-x 21 root root 4096 Oct 31 18:58 ..
lrwxrwxrwx 1 root root 7 Feb 16 2024 bin -> usr/bin
drwxr-xr-x 3 root root 4096 Jul 2 18:27 boot
dr-xr-xr-x 2 root root 4096 Feb 17 2024 cdrom
drwxr-xr-x 6 root root 380 Oct 31 18:49 dev
drwxr-xr-x 103 root root 4096 Oct 25 21:02 etc
drwxr-xr-x 3 root root 4096 Jul 2 18:25 home
lrwxrwxrwx 1 root root 7 Feb 16 2024 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Feb 16 2024 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Feb 16 2024 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Feb 16 2024 libx32 -> usr/libx32
drwx----- 2 root root 16384 Jul 2 18:22 lost+found
drwxr-xr-x 2 root root 4096 Feb 16 2024 media
drwxr-xr-x 4 root root 4096 Oct 25 21:26 mnt
-rw-r--r-- 1 root root 4601268 Oct 31 18:58 my-docker.jar
drwxr-xr-x 2 root root 4096 Feb 16 2024 opt
dr-xr-xr-x 167 root root 0 Oct 31 16:57 proc
drwx----- 10 root root 4096 Oct 31 18:10 root
drwxr-xr-x 17 root root 4096 Oct 31 18:49 rootfs
drwxr-xr-x 17 root root 4096 Oct 31 18:50 rootfs3
-rw-r--r-- 1 root root 128030720 Oct 31 18:51 rootfs3.tar
drwxr-xr-x 29 root root 840 Oct 31 18:30 run
lrwxrwxrwx 1 root root 8 Feb 16 2024/sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Feb 16 2024 srv
dr-xr-xr-x 13 root root 0 Oct 31 16:57 sys
drwxrwxrwt 12 root root 4096 Oct 31 18:45 tmp
drwxr-xr-x 14 root root 4096 Feb 16 2024 usr
drwxr-xr-x 12 root root 4096 Jul 2 18:28 var
root@cv4222321: /#

```

Запускаем:

```
/usr/bin/java -jar my-docker.jar --mem 200M --cpu 0.5 --root rootfs.tar "/bin/bash"
```

Встретившиеся проблемы:

1. Несовместимость версий скомпилированного my-docker.jar и java на linux-е - взять перекомпилировать мини-докер
2. Проблема следующего характера:

```

root@cv4222321: /# /usr/bin/java -jar my-docker.jar --mem 200M --cpu 0.5 --
root rootfs.tar "/bin/bash"
java.io.FileNotFoundException: /sys/fs/cgroup/mydocker-
1730390374960/memory.max (Permission denied)
    at java.base/java.io.FileOutputStream.open0(Native Method)
    at
java.base/java.io.FileOutputStream.open(FileOutputStream.java:298)
    at java.base/java.io.FileOutputStream.<init>

```

```
(FileOutputStream.java:237)
    at java.base/java.io.FileOutputStream.<init>
(FileOutputStream.java:187)
    at java.base/java.io.FileWriter.<init>(FileWriter.java:96)
    at ru.axothy.utils.CgroupUtils.setMemoryLimit(CgroupUtils.java:16)
    at ru.axothy.Minidocker.main(Minidocker.java:40)
```

Видим что не найден определенный файл:

Введем команду и посмотрим что имеется у нас:

```
ls -la /sys/fs/cgroup/mydocker-1730390374960
```

Вывод

```
root@cv4222321:/# ls -la /sys/fs/cgroup/memory/mydocker-1730390374960
total 0
drwxr-xr-x 2 root root 0 Oct 31 18:59 .
drwxr-xr-x 9 root root 0 Oct 31 18:25 ..
-r--r--r-- 1 root root 0 Oct 31 19:01 cgroup.controllers
-r--r--r-- 1 root root 0 Oct 31 19:01 cgroup.events
-rw-r--r-- 1 root root 0 Oct 31 19:01 cgroup.freeze
--w----- 1 root root 0 Oct 31 19:01 cgroup.kill
-rw-r--r-- 1 root root 0 Oct 31 19:01 cgroup.max.depth
-rw-r--r-- 1 root root 0 Oct 31 19:01 cgroup.max.descendants
-rw-r--r-- 1 root root 0 Oct 31 19:01 cgroup.procs
-r--r--r-- 1 root root 0 Oct 31 19:01 cgroup.stat
-rw-r--r-- 1 root root 0 Oct 31 19:01 cgroup.subtree_control
-rw-r--r-- 1 root root 0 Oct 31 19:01 cgroup.threads
-rw-r--r-- 1 root root 0 Oct 31 19:01 cgroup.type
-rw-r--r-- 1 root root 0 Oct 31 19:01 cpu.pressure
-r--r--r-- 1 root root 0 Oct 31 19:01 cpu.stat
-rw-r--r-- 1 root root 0 Oct 31 19:01 io.pressure
-rw-r--r-- 1 root root 0 Oct 31 19:01 memory.pressure
root@cv4222321:/#
```

Если мы не видим что-то вроде memory.max (который отвечает за установку максимального объема памяти для cgroup в системе cgroups v2) и прочего, нужно решить проблему (проблема не в пермишках)

Определяем версию cgroups

```
mount | grep cgroup
```

У меня v2 и я решал след образом. По умолчанию видимо не установлены такие контроллеры как mem, cpu...

```
cat /sys/fs/cgroup/mydocker-1730388452766/cgroup.controllers
```

Вывод команды должен дать что-то вроде memory cpu io

Поэтому для cgroup надо было добавить следующее (точнее для родительского cgroup):

```
echo "+memory" | sudo tee /sys/fs/cgroup/cgroup.subtree_control
```

аналогично добавляем и +cpu

добавили в родительский cgroup строчку memory, соответственно, все дочерние создаваемые "mydocker-таймстемп" будут иметь файлы memory.max и тд... тоже и для cpu

Результаты

Проверим что наш докер контейнер распаковал rootfs.tar

```
ls -la /tmp/mydocker/
```

```
root@cv4222321:~# ls -la /tmp/mydocker/
total 16
drwxr-xr-x  4 root root 4096 Oct 31 20:28 .
drwxrwxrwt 13 root root 4096 Oct 31 20:26 ..
drwxr-xr-x  3 root root 4096 Oct 31 20:26 mydocker-1730395592108
drwxr-xr-x  3 root root 4096 Oct 31 20:28 mydocker-1730395695539
root@cv4222321:~#
```

Видим что после запусков у нас создались директории для каждого докер-контейнера, зайдём в один из них:

```
ls -la /tmp/mydocker/mydocker-1730395695539/rootfs/
```

```
root@cv4222321:~# ls -la /tmp/mydocker/mydocker-1730395695539/rootfs/
total 68
drwxr-xr-x 17 root root 4096 Oct 31 20:28 .
drwxr-xr-x  3 root root 4096 Oct 31 20:28 ..
-rw-r--r--  1 root root    0 Oct 31 20:28 bin
drwxr-xr-x  2 root root 4096 Oct 31 20:28 boot
drwxr-xr-x  4 root root 4096 Oct 31 20:28 dev
drwxr-xr-x 31 root root 4096 Oct 31 20:28 etc
drwxr-xr-x  2 root root 4096 Oct 31 20:28 home
-rw-r--r--  1 root root    0 Oct 31 20:28 lib
-rw-r--r--  1 root root    0 Oct 31 20:28 lib32
-rw-r--r--  1 root root    0 Oct 31 20:28 lib64
-rw-r--r--  1 root root    0 Oct 31 20:28 libx32
drwxr-xr-x  2 root root 4096 Oct 31 20:28 media
drwxr-xr-x  2 root root 4096 Oct 31 20:28 mnt
drwxr-xr-x  2 root root 4096 Oct 31 20:28 opt
drwxr-xr-x  2 root root 4096 Oct 31 20:28 proc
drwxr-xr-x  2 root root 4096 Oct 31 20:28 root
drwxr-xr-x  4 root root 4096 Oct 31 20:28 run
-rw-r--r--  1 root root    0 Oct 31 20:28 sbin
drwxr-xr-x  2 root root 4096 Oct 31 20:28 srv
drwxr-xr-x  2 root root 4096 Oct 31 20:28 sys
drwxr-xr-x  2 root root 4096 Oct 31 20:28 tmp
drwxr-xr-x 14 root root 4096 Oct 31 20:28 usr
drwxr-xr-x 11 root root 4096 Oct 31 20:28 var
root@cv4222321:~#
```

По итогу видим нашу систему.

И таких можно создавать бесконечно, за все отвечает программа на Java

```
root@cv4222321:~# ls -la /tmp/mydocker/
total 148
drwxr-xr-x 37 root root 4096 Oct 31 22:49 .
drwxrwxrwt 13 root root 4096 Oct 31 20:26 ..
drwxr-xr-x 3 root root 4096 Oct 31 20:26 mydocker-1730395592108
drwxr-xr-x 3 root root 4096 Oct 31 20:28 mydocker-1730395695539
drwxr-xr-x 3 root root 4096 Oct 31 20:40 mydocker-1730396435426
drwxr-xr-x 3 root root 4096 Oct 31 20:40 mydocker-1730396451830
drwxr-xr-x 3 root root 4096 Oct 31 20:41 mydocker-1730396489472
drwxr-xr-x 3 root root 4096 Oct 31 20:43 mydocker-1730396627044
drwxr-xr-x 3 root root 4096 Oct 31 20:50 mydocker-1730397010970
drwxr-xr-x 3 root root 4096 Oct 31 20:53 mydocker-1730397190924
drwxr-xr-x 3 root root 4096 Oct 31 21:00 mydocker-1730397641513
drwxr-xr-x 3 root root 4096 Oct 31 21:07 mydocker-1730398054662
drwxr-xr-x 3 root root 4096 Oct 31 21:08 mydocker-1730398090545
drwxr-xr-x 3 root root 4096 Oct 31 21:12 mydocker-1730398326060
drwxr-xr-x 3 root root 4096 Oct 31 21:13 mydocker-1730398426419
drwxr-xr-x 3 root root 4096 Oct 31 21:16 mydocker-1730398584840
drwxr-xr-x 3 root root 4096 Oct 31 21:16 mydocker-1730398611883
drwxr-xr-x 3 root root 4096 Oct 31 21:17 mydocker-1730398662052
drwxr-xr-x 3 root root 4096 Oct 31 21:18 mydocker-1730398686196
drwxr-xr-x 3 root root 4096 Oct 31 21:18 mydocker-1730398691408
drwxr-xr-x 3 root root 4096 Oct 31 21:18 mydocker-1730398697193
drwxr-xr-x 3 root root 4096 Oct 31 21:18 mydocker-1730398699281
drwxr-xr-x 3 root root 4096 Oct 31 21:54 mydocker-1730400889732
drwxr-xr-x 3 root root 4096 Oct 31 21:56 mydocker-1730400985487
drwxr-xr-x 3 root root 4096 Oct 31 22:00 mydocker-1730401247101
drwxr-xr-x 3 root root 4096 Oct 31 22:00 mydocker-1730401251585
drwxr-xr-x 3 root root 4096 Oct 31 22:05 mydocker-1730401525066
drwxr-xr-x 3 root root 4096 Oct 31 22:09 mydocker-1730401798144
drwxr-xr-x 3 root root 4096 Oct 31 22:10 mydocker-1730401853663
drwxr-xr-x 3 root root 4096 Oct 31 22:15 mydocker-1730402109136
drwxr-xr-x 3 root root 4096 Oct 31 22:16 mydocker-1730402187395
drwxr-xr-x 3 root root 4096 Oct 31 22:18 mydocker-1730402319230
drwxr-xr-x 3 root root 4096 Oct 31 22:24 mydocker-1730402652673
drwxr-xr-x 3 root root 4096 Oct 31 22:25 mydocker-1730402725231
drwxr-xr-x 3 root root 4096 Oct 31 22:45 mydocker-1730403922894
drwxr-xr-x 3 root root 4096 Oct 31 22:48 mydocker-1730404115679
drwxr-xr-x 3 root root 4096 Oct 31 22:49 mydocker-1730404140557
root@cv4222321:~#
```