

---docker学习笔记---

docker 是什么？

Docker是基于Go语言实现的云开源项目。

Docker的主要目标是“Build, Ship and Run Any App,Anywhere”，也就是通过对应用组件的封装、分发、部署、运行等生命周期的管理，使用户的APP（可以是一个WEB应用或数据库应用等等）及其运行环境能够做到“一次镜像，处处运行”。

Linux容器技术的出现就解决了这样一个问题，而 Docker 就是在它的基础上发展过来的。将应用打成镜像，通过镜像成为运行在Docker容器上面的实例，而 Docker容器在任何操作系统上都是一致的，这就实现了跨平台、跨服务器。只需要一次配置好环境，换到别的机器上就可以一键部署好，大大简化了操作。

容器和虚拟机比较

虚拟机

虚拟机（virtual machine）就是带环境安装的一种解决方案。

它可以在一种操作系统里面运行另一种操作系统，比如在Windows10系统里面运行Linux系统CentOS7。应用程序对此毫无感知，因为虚拟机看上去跟真实系统一模一样，而对于底层系统来说，虚拟机就是一个普通文件，不需要了就删掉，对其他部分毫无影响。这类虚拟机完美的运行了另一套系统，能够使应用程序，操作系统和硬件三者之间的逻辑不变。

虚拟机的缺点：

- 资源占用多
- 冗余步骤多
- 启动慢

容器

Linux容器(Linux Containers, 缩写为 LXC)

Linux容器是与系统其他部分隔离开的一系列进程，从另一个镜像运行，并由该镜像提供支持进程所需的全部文件。容器提供的镜像包含了应用的所有依赖项，因而在从开发到测试再到生产的整个过程中，它都具有可移植性和一致性。

Linux 容器不是模拟一个完整的操作系统而是对进程进行隔离。有了容器，就可以将软件运行所需的所有资源打包到一个隔离的容器中。容器与虚拟机不同，不需要捆绑一整套操作系统，只需要软件工作所需的库资源和设置。系统因此而变得高效轻量并保证部署在任何环境中的软件都能始终如一地运行。

启动速度快，占用体积小

对比

- 传统虚拟机技术是虚拟出一套硬件后，在其上运行一个完整操作系统，在该系统上再运行所需应用进程；
- 容器内的应用进程直接运行于宿主的内核，容器内没有自己的内核且也没有进行硬件虚拟。因此容器要比传统虚拟机更为轻便。
- 每个容器之间互相隔离，每个容器有自己的文件系统，容器之间进程不会相互影响，能区分计算资源。

docker的特点

一次构建，随处运行

更快的应用交付和部署

传统的应用开发完成后，需要提供一堆安装程序和配置说明文档，安装部署后需根据配置文档进行繁杂的配置才能正常运行。Docker化之后只需要交付少量容器镜像文件，在正式生产环境加载镜像并运行即可，应用安装配置在镜像里已经内置好，大大节省部署配置和测试验证时间。

更便捷的升级和扩容和缩容

随着微服务架构和Docker的发展，大量的应用会通过微服务方式架构，应用的开发构建将变成搭乐高积木一样，每个Docker容器将变成一块“积木”，应用的升级将变得非常容易。当现有的容器不足以支撑业务处理时，可通过镜像运行新的容器进行快速扩容，使应用系统的扩容从原先的天级变成分钟级甚至秒级。

更简单的系统运维

应用容器化运行后，生产环境运行的应用可与开发、测试环境的应用高度一致，容器会将应用程序相关的环境和状态完全封装起来，不会因为底层基础架构和操作系统的不同给应用带来影响，产生新的BUG。当出现程序异常时，也可以通过测试环境的相同容器进行快速定位和修复。

更高效的计算资源利用

Docker是内核级虚拟化，其不像传统的虚拟化技术一样需要额外的Hypervisor支持，所以在一台物理机上可以运行很多个容器实例，可大大提升物理服务器的CPU和内存的利用率。

docker下载

docker官网: <http://www.docker.com>

设置镜像:

yum-config-manager --add-repo <http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo>

docker的组成

镜像

Docker 镜像（Image）就是一个**只读**的模板。镜像可以用来创建 Docker 容器，一个镜像可以创建很多容器。

它也相当于是一个root文件系统。比如官方镜像 centos:7 就包含了完整的一套 centos:7 最小系统的 root 文件系统。

相当于容器的“源代码”，docker镜像文件类似于Java的类模板，而docker容器实例类似于java中新出来的实例对象。

容器

1. 从面向对象角度

Docker 利用容器（Container）独立运行的一个或一组应用，应用程序或服务运行在容器里面，容器就类似于一个虚拟化的运行环境，容器是用镜像创建的运行实例。就像是Java中的类和实例对象一样，镜像静态的定义，容器是镜像运行时的实体。容器为镜像提供了一个标准的和隔离的运行环境，它可以被启动、开始、停止、删除。每个容器都是相互隔离的、保证安全的平台

2. 从镜像容器角度

可以把容器看做是一个简易版的 Linux 环境（包括root用户权限、进程空间、用户空间和网络空间等）和运行在其中的应用程序。

仓库

仓库（Repository）是集中存放镜像文件的场所。

类似于

Maven仓库，存放各种jar包的地方；

github仓库，存放各种git项目的地方；

Docker公司提供的官方registry被称为Docker Hub，存放各种镜像模板的地方。

仓库分为公开仓库（Public）和私有仓库（Private）两种形式。

最大的公开仓库是 Docker Hub(<https://hub.docker.com/>),

存放了数量庞大的镜像供用户下载。国内的公开仓库包括阿里云、网易云等

docker常用命令

镜像命令

docker images

列出本地主机上的镜像

```
PS C:\Users\mao\Desktop> docker images
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
hello-world     latest      feb5d9fea6a5   8 months ago   13.3kB
PS C:\Users\mao\Desktop>
```

- REPOSITORY: 表示镜像的仓库源
- TAG: 镜像的标签版本号
- IMAGE ID: 镜像ID
- CREATED: 镜像创建时间
- SIZE: 镜像大小

同一仓库源可以有多个 TAG 版本, 代表这个仓库源的不同个版本, 我们使用 REPOSITORY:TAG 来定义不同的镜像。如果你不指定一个镜像的版本标签, 例如你只使用 ubuntu, docker 将默认使用 ubuntu:latest 镜像

参数:

-a: 列出本地所有的镜像 (含历史映像层)

-q: 只显示镜像ID。

```
PS C:\Users\mao\Desktop> docker images -q
feb5d9fea6a5
PS C:\Users\mao\Desktop>
```

docker search 镜像名

```
PS C:\Users\mao\Desktop> docker search redis
NAME                                STARS      OFFICIAL    AUTOMATED    DESCRIPTION
redis                               10986      [OK]
bitnami/redis                       220        [OK]         Bitnami Redis Docker Image
bitnami/redis-sentinel              37        [OK]         Bitnami Docker Image for Redis Sentinel
bitnami/redis-cluster               32
rapidfort/redis-cluster             13         RapidFort optimized, hardened image for Redis
circleci/redis                     13        [OK]         CircleCI images for Redis
rapidfort/redis                     13         RapidFort optimized, hardened image for Redis
ubuntu/redis                        10         Redis, an open source key-value store. Long-term support
```

```

bitnami/redis-exporter
7
clearlinux/redis
Redis key-value data structure
server with t... 4
google/guestbook-python-redis
A simple guestbook example
written in Python... 4
ibmcom/ibm-cloud-databases-redis-operator-bundle
Bundle image for the IBM
Operator for Redis 1
bitnami/redis-sentinel-exporter
1
ibmcom/ibm-cloud-databases-redis-operator
Container image for the IBM
Operator for Red... 1
ibmcom/redis-ha
1
ibmcom/ibm-cloud-databases-redis-catalog
Catalog image for the IBM
Operator for Redis 1
cimg/redis
0
blackflysolutions/redis
Redis container for Drupal
and CiviCRM 0
ibmcom/redisearch-ppc64le
0
drud/redis
redis
0 [OK]
ibmcom/redis-ppc64le
0
rancher/redislabs-ssl
0
vmware/redis-photon
0
newrelic/k8s-nri-redis
New Relic Infrastructure Redis
Integration (... 0
astronomerinc/ap-redis
Redis service for Airflow's
celery executor ... 0
PS C:\Users\mao\Desktop>

```

- NAME: 镜像名字
- DESCRIPTION: 镜像说明
- STARS: 点赞数量
- OFFICIAL: 是否为官方
- AUTOMATED: 是否是自动构建的

参数:

--limit: 只列出N个镜像, 默认25个

```

PS C:\Users\mao\Desktop> docker search --limit 5 redis
NAME                                DESCRIPTION                                STARS
OFFICIAL    AUTOMATED
redis                                Redis is an open source key-value store that... 10986
[OK]
bitnami/redis                        Bitnami Redis Docker Image                220
[OK]
bitnami/redis-sentinel              Bitnami Docker Image for Redis Sentinel    37
[OK]
circleci/redis                     CircleCI images for Redis                  13
[OK]
bitnami/redis-exporter              7
PS C:\Users\mao\Desktop> docker search --limit 2 redis
NAME                                DESCRIPTION                                STARS
OFFICIAL    AUTOMATED
redis                                Redis is an open source key-value store that... 10986 [OK]
bitnami/redis                        Bitnami Redis Docker Image                220
[OK]
PS C:\Users\mao\Desktop>

```

docker pull 镜像名

下载某个镜像

或者 docker pull 镜像名:TAG

如果没有指定TAG，则下载最新版，相当于docker pull 镜像名:latest

```

PS C:\Users\mao\Desktop> docker pull redis
Using default tag: latest
latest: Pulling from library/redis
42c077c10790: Pull complete
a300d83d65f9: Pull complete
ebdc3afaab5c: Pull complete
31eec7f8651c: Pull complete
9c6a6b89d274: Pull complete
5c8099a4b45c: Pull complete
Digest: sha256:1b90dbfe6943c72a7469c134cad3f02eb810f016049a0e19ad78be07040cdb0c
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Users\mao\Desktop>

```

```

PS C:\Users\mao\Desktop> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
redis         latest    53aa81e8adfa   3 days ago    117MB
hello-world   latest    feb5d9fea6a5   8 months ago  13.3kB
PS C:\Users\mao\Desktop>

```

docker system df

查看镜像/容器/数据卷所占的空间

```
PS C:\Users\mao\Desktop> docker system df
TYPE                TOTAL        ACTIVE        SIZE        RECLAIMABLE
Images              2            1            116.8MB     116.8MB (99%)
Containers          4            0             0B           0B
Local volumes       0            0             0B           0B
Build cache         0            0             0B           0B
PS C:\Users\mao\Desktop>
```

docker rmi 镜像ID

删除镜像

- 删除一个: `docker rmi -f 镜像ID`
- 删除多个: `docker rmi -f 镜像名1:TAG 镜像名2:TAG ...`
- 删除全部: `docker rmi -f $(docker images -qa)`

```
PS C:\Users\mao\Desktop> docker images
REPOSITORY    TAG        IMAGE ID      CREATED        SIZE
redis         latest    53aa81e8adfa  3 days ago    117MB
hello-world   latest    feb5d9fea6a5  8 months ago  13.3kB
PS C:\Users\mao\Desktop> docker rmi -f feb5d9fea6a5
Untagged: hello-world:latest
Untagged: hello-
world@sha256:10d7d58d5ebd2a652f4d93fdd86da8f265f5318c6a73cc5b6a9798ff6d2b2e67
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
PS C:\Users\mao\Desktop> docker images
REPOSITORY    TAG        IMAGE ID      CREATED        SIZE
redis         latest    53aa81e8adfa  3 days ago    117MB
PS C:\Users\mao\Desktop>
```

容器命令

新建和启动容器

命令: `docker run [options] image [COMMAND] [ARG...]`

OPTIONS说明:

- name="容器新名字" 为容器指定一个名称
- d: 后台运行容器并返回容器ID, 也即启动守护式容器(后台运行);
- i: 以交互模式运行容器, 通常与 -t 同时使用;
- t: 为容器重新分配一个伪输入终端, 通常与 -i 同时使用;

也即启动交互式容器(前台有伪终端, 等待交互);

-P: 随机端口映射, 大写P

-p: 指定端口映射, 小写p

```
PS C:\Users\mao\Desktop> docker run -it ubuntu
root@0f625a3b6f69:/#
root@0f625a3b6f69:/#
root@0f625a3b6f69:/#
root@0f625a3b6f69:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root
run sbin srv sys tmp usr var
root@0f625a3b6f69:/#
```

后台运行:

```
PS C:\Users\mao\Desktop> docker run -d ubuntu
5952c102876398890fd1e4cf5130b13922229a170f603cef4fd9b9b619576874
PS C:\Users\mao\Desktop>
```

列出当前所有正在运行的容器

命令: docker ps

```
PS C:\Users\mao\Desktop> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS
NAMES
057d044acdc   ubuntu   "bash"    15 seconds ago   Up 15 seconds
musing_ride
PS C:\Users\mao\Desktop>
```

参数:

-a :列出当前所有正在运行的容器+历史上运行过的


```
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------------|---------------------|-------------------|------------------------------|
| 057d044acdcb | ubuntu | "bash" | 2 minutes ago | Up 2 minutes |
| | | musings_ride | | |
| 5952c1028763 | ubuntu | "bash" | 3 minutes ago | Exited (0) 3 minutes ago |
| | | silly_noyce | | |
| 0f625a3b6f69 | ubuntu | "bash" | 5 minutes ago | Exited (130) 3 minutes ago |
| | | silly_ellis | | |
| f62123ed8d68 | feb5d9fea6a5 | "/hello" | About an hour ago | Exited (0) About an hour ago |
| | | hungry_hamilton | | |
| 09603a7cc2bb | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |
| | | keen_rosalind | | |
| 58f57bb9913f | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |
| | | determined_ishizaka | | |
| 057165d214ed | feb5d9fea6a5 | "/hello" | 4 weeks ago | Exited (0) 4 weeks ago |
| | | goofy_hertz | | |

```
PS C:\Users\mao\Desktop>
```

-l:显示最近创建的容器。

```
PS C:\Users\mao\Desktop> docker ps -l
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS |
|--------------|--------|--------------|---------------|--------------|-------|
| 057d044acdcb | ubuntu | "bash" | 2 minutes ago | Up 2 minutes | |
| | | musings_ride | | | |

```
PS C:\Users\mao\Desktop>
```

-n: 显示最近n个创建的容器。

```
PS C:\Users\mao\Desktop> docker ps -n 2
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------|--------------|---------------|--------------------------|
| 057d044acdcb | ubuntu | "bash" | 3 minutes ago | Up 3 minutes |
| | | musings_ride | | |
| 5952c1028763 | ubuntu | "bash" | 4 minutes ago | Exited (0) 4 minutes ago |
| | | silly_noyce | | |

```
PS C:\Users\mao\Desktop> docker ps -n 4
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------------|-----------------|-------------------|------------------------------|
| 057d044acdcb | ubuntu | "bash" | 3 minutes ago | Up 3 minutes |
| | | musings_ride | | |
| 5952c1028763 | ubuntu | "bash" | 4 minutes ago | Exited (0) 4 minutes ago |
| | | silly_noyce | | |
| 0f625a3b6f69 | ubuntu | "bash" | 6 minutes ago | Exited (130) 5 minutes ago |
| | | silly_ellis | | |
| f62123ed8d68 | feb5d9fea6a5 | "/hello" | About an hour ago | Exited (0) About an hour ago |
| | | hungry_hamilton | | |

```
PS C:\Users\mao\Desktop>
```

-q:静默模式，只显示容器编号。

```
PS C:\Users\mao\Desktop> docker ps -q
057d044acdcb
PS C:\Users\mao\Desktop>
```

退出容器

方式一：

run进去容器，exit退出，容器停止

```
PS C:\Users\mao\Desktop> docker run -it ubuntu
root@cb7c2e3d6b9f:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root
run sbin srv sys tmp usr var
root@cb7c2e3d6b9f:/# exit
exit
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------------|---------------------|--------------------|--------------------|
| | PORTS | NAMES | | |
| cb7c2e3d6b9f | ubuntu | "bash" | 38 seconds ago | Exited (0) 13 |
| seconds ago | | zealous_kirch | | |
| a6b5661d143d | ubuntu | "bash" | About a minute ago | Exited (0) About a |
| minute ago | | admiring_colden | | |
| 057d044acdcb | ubuntu | "bash" | 8 minutes ago | Exited (127) 2 |
| minutes ago | | musings_ride | | |
| 5952c1028763 | ubuntu | "bash" | 9 minutes ago | Exited (0) 9 |
| minutes ago | | silly_noyce | | |
| 0f625a3b6f69 | ubuntu | "bash" | 11 minutes ago | Exited (130) 10 |
| minutes ago | | silly_ellis | | |
| f62123ed8d68 | feb5d9fea6a5 | "/hello" | About an hour ago | Exited (0) About |
| an hour ago | | hungry_hamilton | | |
| 09603a7cc2bb | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours |
| ago | | keen_rosalind | | |
| 58f57bb9913f | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours |
| ago | | determined_ishizaka | | |
| 057165d214ed | feb5d9fea6a5 | "/hello" | 4 weeks ago | Exited (0) 4 weeks |
| ago | | goofy_hertz | | |

```
PS C:\Users\mao\Desktop>
```

方式二：

进去容器，ctrl+p+q退出，容器不停止

```
PS C:\Users\mao\Desktop> docker run -it ubuntu
root@f8d05f9740a4:/#
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|-------|---------|---------|--------|
| | PORTS | NAMES | | |

```
f8d05f9740a4  ubuntu  "bash"  22 seconds ago  Up 22 seconds
sad_elion
cb7c2e3d6b9f  ubuntu  "bash"  About a minute ago  Exited (0) About a
minute ago  zealous_kirch
a6b5661d143d  ubuntu  "bash"  2 minutes ago  Exited (0) 2
minutes ago  admiring_colden
057d044acdc  ubuntu  "bash"  9 minutes ago  Exited (127) 4
minutes ago  musing_ride
5952c1028763  ubuntu  "bash"  10 minutes ago  Exited (0) 10
minutes ago  silly_noyce
0f625a3b6f69  ubuntu  "bash"  12 minutes ago  Exited (130) 11
minutes ago  silly_ellis
f62123ed8d68  feb5d9fea6a5  "/hello"  About an hour ago  Exited (0) About
an hour ago  hungry_hamilton
09603a7cc2bb  feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours
ago  keen_rosalind
58f57bb9913f  feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours
ago  determined_ishizaka
057165d214ed  feb5d9fea6a5  "/hello"  4 weeks ago  Exited (0) 4 weeks
ago  goofy_hertz
PS C:\Users\mao\Desktop>
```

启动已停止运行的容器

命令：docker start 容器id或者容器名

```
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
43c96b66638b   ubuntu    "bash"                  2 minutes ago  Up 2 minutes
stupefied_sinoussi
5065bd9f0811   ubuntu    "bash"                  2 minutes ago  Up 2 minutes
sleepy_mahavira
2e6e0a4e3a1d   ubuntu    "bash"                  2 minutes ago  Up 2 minutes
modest_visvesvaraya
f8d05f9740a4   ubuntu    "bash"                  3 minutes ago  Up 3 minutes
sad_elion
cb7c2e3d6b9f   ubuntu    "bash"                  5 minutes ago  Exited (0) 4
minutes ago  zealous_kirch
a6b5661d143d   ubuntu    "bash"                  5 minutes ago  Exited (0) 4
seconds ago  admiring_colden
057d044acdc    ubuntu    "bash"                  12 minutes ago  Exited (127) 7
minutes ago  musing_ride
5952c1028763   ubuntu    "bash"                  14 minutes ago  Exited (0) 14
minutes ago  silly_noyce
0f625a3b6f69   ubuntu    "bash"                  15 minutes ago  Exited (130) 14
minutes ago  silly_ellis
f62123ed8d68   feb5d9fea6a5  "/hello"  About an hour ago  Exited (0) About an
hour ago  hungry_hamilton
09603a7cc2bb   feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours
ago  keen_rosalind
58f57bb9913f   feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours
ago  determined_ishizaka
057165d214ed   feb5d9fea6a5  "/hello"  4 weeks ago  Exited (0) 4 weeks
ago  goofy_hertz
```

```

PS C:\Users\mao\Desktop> docker start 057d044acdcb
057d044acdcb
PS C:\Users\mao\Desktop> docker ps -a

```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|---------------------|----------|-------------------|---------------------|
| PORTS | NAMES | | | |
| 43c96b66638b | ubuntu | "bash" | 3 minutes ago | Up 3 minutes |
| | stupefied_sinoussi | | | |
| 5065bd9f0811 | ubuntu | "bash" | 3 minutes ago | Up 3 minutes |
| | sleepy_mahavira | | | |
| 2e6e0a4e3a1d | ubuntu | "bash" | 3 minutes ago | Up 3 minutes |
| | modest_visvesvaraya | | | |
| f8d05f9740a4 | ubuntu | "bash" | 4 minutes ago | Up 4 minutes |
| | sad_elion | | | |
| cb7c2e3d6b9f | ubuntu | "bash" | 5 minutes ago | Exited (0) 5 |
| minutes ago | zealous_kirch | | | |
| a6b5661d143d | ubuntu | "bash" | 6 minutes ago | Exited (0) 45 |
| seconds ago | admiring_colden | | | |
| 057d044acdcb | ubuntu | "bash" | 13 minutes ago | Up 3 seconds |
| | musing_ride | | | |
| 5952c1028763 | ubuntu | "bash" | 14 minutes ago | Exited (0) 14 |
| minutes ago | silly_noyce | | | |
| 0f625a3b6f69 | ubuntu | "bash" | 16 minutes ago | Exited (130) 15 |
| minutes ago | silly_ellis | | | |
| f62123ed8d68 | feb5d9fea6a5 | "/hello" | About an hour ago | Exited (0) About an |
| hour ago | hungry_hamilton | | | |
| 09603a7cc2bb | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours |
| ago | keen_rosalind | | | |
| 58f57bb9913f | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours |
| ago | determined_ishizaka | | | |
| 057165d214ed | feb5d9fea6a5 | "/hello" | 4 weeks ago | Exited (0) 4 weeks |
| ago | goofy_hertz | | | |

```

PS C:\Users\mao\Desktop>

```

```

PS C:\Users\mao\Desktop> docker ps -a

```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|---------------------|---------|----------------|--------------------|
| PORTS | NAMES | | | |
| 43c96b66638b | ubuntu | "bash" | 3 minutes ago | Up 3 minutes |
| | stupefied_sinoussi | | | |
| 5065bd9f0811 | ubuntu | "bash" | 4 minutes ago | Up 4 minutes |
| | sleepy_mahavira | | | |
| 2e6e0a4e3a1d | ubuntu | "bash" | 4 minutes ago | Up 4 minutes |
| | modest_visvesvaraya | | | |
| f8d05f9740a4 | ubuntu | "bash" | 5 minutes ago | Up 5 minutes |
| | sad_elion | | | |
| cb7c2e3d6b9f | ubuntu | "bash" | 6 minutes ago | Exited (0) 6 |
| minutes ago | zealous_kirch | | | |
| a6b5661d143d | ubuntu | "bash" | 7 minutes ago | Exited (0) About a |
| minute ago | admiring_colden | | | |
| 057d044acdcb | ubuntu | "bash" | 14 minutes ago | Up 53 seconds |
| | musing_ride | | | |
| 5952c1028763 | ubuntu | "bash" | 15 minutes ago | Exited (0) 15 |
| minutes ago | silly_noyce | | | |
| 0f625a3b6f69 | ubuntu | "bash" | 17 minutes ago | Exited (130) 16 |
| minutes ago | silly_ellis | | | |

```

f62123ed8d68 feb5d9fea6a5 "/hello" About an hour ago Exited (0) About an
hour ago hungry_hamilton
09603a7cc2bb feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours
ago keen_rosalind
58f57bb9913f feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours
ago determined_ishizaka
057165d214ed feb5d9fea6a5 "/hello" 4 weeks ago Exited (0) 4 weeks
ago goofy_hertz
PS C:\Users\mao\Desktop> docker start silly_ellis
silly_ellis
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
43c96b66638b ubuntu "bash" 4 minutes ago Up 4 minutes
stupefied_sinoussi
5065bd9f0811 ubuntu "bash" 4 minutes ago Up 4 minutes
sleepy_mahavira
2e6e0a4e3a1d ubuntu "bash" 4 minutes ago Up 4 minutes
modest_visvesvaraya
f8d05f9740a4 ubuntu "bash" 5 minutes ago Up 5 minutes
sad_elion
cb7c2e3d6b9f ubuntu "bash" 7 minutes ago Exited (0) 6 minutes
ago zealous_kirch
a6b5661d143d ubuntu "bash" 7 minutes ago Exited (0) 2 minutes
ago admiring_colden
057d044acdcb ubuntu "bash" 14 minutes ago Up About a minute
musing_ride
5952c1028763 ubuntu "bash" 16 minutes ago Exited (0) 16 minutes
ago silly_noyce
0f625a3b6f69 ubuntu "bash" 18 minutes ago Up 4 seconds
silly_ellis
f62123ed8d68 feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
hungry_hamilton
09603a7cc2bb feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
keen_rosalind
58f57bb9913f feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
determined_ishizaka
057165d214ed feb5d9fea6a5 "/hello" 4 weeks ago Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop>

```

重启容器

命令：docker restart 容器id或者容器名

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
43c96b66638b ubuntu "bash" 5 minutes ago Up 5 minutes
stupefied_sinoussi
5065bd9f0811 ubuntu "bash" 5 minutes ago Up 5 minutes
sleepy_mahavira

```

```

2e6e0a4e3a1d ubuntu "bash" 6 minutes ago Up 6 minutes
modest_visvesvaraya
f8d05f9740a4 ubuntu "bash" 7 minutes ago Up 7 minutes
sad_elion
cb7c2e3d6b9f ubuntu "bash" 8 minutes ago Exited (0) 8 minutes
ago zealous_kirch
a6b5661d143d ubuntu "bash" 8 minutes ago Exited (0) 3 minutes
ago admiring_colden
057d044acdcb ubuntu "bash" 16 minutes ago Up 2 minutes
musing_ride
5952c1028763 ubuntu "bash" 17 minutes ago Exited (0) 17 minutes
ago silly_noyce
0f625a3b6f69 ubuntu "bash" 19 minutes ago Up About a minute
silly_ellis
f62123ed8d68 feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
hungry_hamilton
09603a7cc2bb feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
keen_rosalind
58f57bb9913f feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
determined_ishizaka
057165d214ed feb5d9fea6a5 "/hello" 4 weeks ago Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop> docker restart 43c96b66638b
43c96b66638b
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
43c96b66638b ubuntu "bash" 6 minutes ago Up 5 seconds
stupefied_sinoussi
5065bd9f0811 ubuntu "bash" 6 minutes ago Up 6 minutes
sleepy_mahavira
2e6e0a4e3a1d ubuntu "bash" 6 minutes ago Up 6 minutes
modest_visvesvaraya
f8d05f9740a4 ubuntu "bash" 7 minutes ago Up 7 minutes
sad_elion
cb7c2e3d6b9f ubuntu "bash" 9 minutes ago Exited (0) 8 minutes
ago zealous_kirch
a6b5661d143d ubuntu "bash" 9 minutes ago Exited (0) 4 minutes
ago admiring_colden
057d044acdcb ubuntu "bash" 17 minutes ago Up 3 minutes
musing_ride
5952c1028763 ubuntu "bash" 18 minutes ago Exited (0) 18 minutes
ago silly_noyce
0f625a3b6f69 ubuntu "bash" 20 minutes ago Up 2 minutes
silly_ellis
f62123ed8d68 feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
hungry_hamilton
09603a7cc2bb feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
keen_rosalind
58f57bb9913f feb5d9fea6a5 "/hello" 2 hours ago Exited (0) 2 hours ago
determined_ishizaka
057165d214ed feb5d9fea6a5 "/hello" 4 weeks ago Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop>

```

```
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------------|----------|----------------|---------------------------|
| PORTS | NAMES | | | |
| 43c96b66638b | ubuntu | "bash" | 6 minutes ago | Up 5 seconds |
| 5065bd9f0811 | ubuntu | "bash" | 6 minutes ago | Up 6 minutes |
| 2e6e0a4e3a1d | ubuntu | "bash" | 6 minutes ago | Up 6 minutes |
| f8d05f9740a4 | ubuntu | "bash" | 7 minutes ago | Up 7 minutes |
| cb7c2e3d6b9f | ubuntu | "bash" | 9 minutes ago | Exited (0) 8 minutes ago |
| a6b5661d143d | ubuntu | "bash" | 9 minutes ago | Exited (0) 4 minutes ago |
| 057d044acdcb | ubuntu | "bash" | 17 minutes ago | Up 3 minutes |
| 5952c1028763 | ubuntu | "bash" | 18 minutes ago | Exited (0) 18 minutes ago |
| 0f625a3b6f69 | ubuntu | "bash" | 20 minutes ago | Up 2 minutes |
| f62123ed8d68 | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |
| 09603a7cc2bb | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |
| 58f57bb9913f | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |
| 057165d214ed | feb5d9fea6a5 | "/hello" | 4 weeks ago | Exited (0) 4 weeks ago |

```
PS C:\Users\mao\Desktop> docker restart sleepy_mahavira  
sleepy_mahavira
```

```
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------------|----------|----------------|---------------------------|
| PORTS | NAMES | | | |
| 43c96b66638b | ubuntu | "bash" | 7 minutes ago | Up About a minute |
| 5065bd9f0811 | ubuntu | "bash" | 7 minutes ago | Up 3 seconds |
| 2e6e0a4e3a1d | ubuntu | "bash" | 8 minutes ago | Up 8 minutes |
| f8d05f9740a4 | ubuntu | "bash" | 8 minutes ago | Up 8 minutes |
| cb7c2e3d6b9f | ubuntu | "bash" | 10 minutes ago | Exited (0) 9 minutes ago |
| a6b5661d143d | ubuntu | "bash" | 10 minutes ago | Exited (0) 5 minutes ago |
| 057d044acdcb | ubuntu | "bash" | 18 minutes ago | Up 4 minutes |
| 5952c1028763 | ubuntu | "bash" | 19 minutes ago | Exited (0) 19 minutes ago |
| 0f625a3b6f69 | ubuntu | "bash" | 21 minutes ago | Up 3 minutes |
| f62123ed8d68 | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |
| 09603a7cc2bb | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |
| 58f57bb9913f | feb5d9fea6a5 | "/hello" | 2 hours ago | Exited (0) 2 hours ago |

```
057165d214ed    feb5d9fea6a5    "/hello"    4 weeks ago    Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop>
```

停止容器

命令: docker stop 容器id或者容器名

```
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE          COMMAND          CREATED          STATUS
PORTS         NAMES
43c96b66638b   ubuntu        "bash"         8 minutes ago   Up 2 minutes
stupefied_sinoussi
5065bd9f0811   ubuntu        "bash"         8 minutes ago   Up About a minute
sleepy_mahavira
2e6e0a4e3a1d   ubuntu        "bash"         9 minutes ago   Up 9 minutes
modest_visvesvaraya
f8d05f9740a4   ubuntu        "bash"         9 minutes ago   Up 9 minutes
sad_elion
cb7c2e3d6b9f   ubuntu        "bash"         11 minutes ago  Exited (0) 11 minutes
ago
zealous_kirch
a6b5661d143d   ubuntu        "bash"         11 minutes ago  Exited (0) 6 minutes
ago
admiring_colden
057d044acdcb   ubuntu        "bash"         19 minutes ago  Up 5 minutes
musing_ride
5952c1028763   ubuntu        "bash"         20 minutes ago  Exited (0) 20 minutes
ago
silly_noyce
0f625a3b6f69   ubuntu        "bash"         22 minutes ago  Up 4 minutes
silly_ellis
f62123ed8d68   feb5d9fea6a5 "/hello"       2 hours ago     Exited (0) 2 hours ago
hungry_hamilton
09603a7cc2bb   feb5d9fea6a5 "/hello"       2 hours ago     Exited (0) 2 hours ago
keen_rosalind
58f57bb9913f   feb5d9fea6a5 "/hello"       2 hours ago     Exited (0) 2 hours ago
determined_ishizaka
057165d214ed   feb5d9fea6a5 "/hello"       4 weeks ago     Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop> docker stop 43c96b66638b
43c96b66638b
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE          COMMAND          CREATED          STATUS
PORTS         NAMES
43c96b66638b   ubuntu        "bash"         9 minutes ago   Exited (137) 3 seconds
ago
stupefied_sinoussi
5065bd9f0811   ubuntu        "bash"         9 minutes ago   Up About a minute
sleepy_mahavira
2e6e0a4e3a1d   ubuntu        "bash"         9 minutes ago   Up 9 minutes
modest_visvesvaraya
f8d05f9740a4   ubuntu        "bash"         10 minutes ago  Up 10 minutes
sad_elion
cb7c2e3d6b9f   ubuntu        "bash"         11 minutes ago  Exited (0) 11 minutes
ago
zealous_kirch
```



```

a6b5661d143d  ubuntu      "bash"      12 minutes ago  Exited (0) 6 minutes
ago          admiring_colden
057d044acdcb  ubuntu      "bash"      19 minutes ago  Up 6 minutes
              musing_ride
5952c1028763  ubuntu      "bash"      20 minutes ago  Exited (0) 20 minutes
ago          silly_noyce
0f625a3b6f69  ubuntu      "bash"      22 minutes ago  Up 4 minutes
              silly_ellis
f62123ed8d68  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
              hungry_hamilton
09603a7cc2bb  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
              keen_rosalind
58f57bb9913f  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
              determined_ishizaka
057165d214ed  feb5d9fea6a5  "/hello"    4 weeks ago     Exited (0) 4 weeks ago
              goofy_hertz
PS C:\Users\mao\Desktop>

```

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS         NAMES
43c96b66638b  ubuntu    "bash"                  9 minutes ago   Exited (137) 26
seconds ago    stupefied_sinoussi
5065bd9f0811   ubuntu    "bash"                  9 minutes ago   Up About a minute
              sleepy_mahavira
2e6e0a4e3a1d   ubuntu    "bash"                  9 minutes ago   Up 9 minutes
              modest_visvesvaraya
f8d05f9740a4   ubuntu    "bash"                  10 minutes ago  Up 10 minutes
              sad_elion
cb7c2e3d6b9f   ubuntu    "bash"                  12 minutes ago  Exited (0) 11 minutes
ago           zealous_kirch
a6b5661d143d   ubuntu    "bash"                  12 minutes ago  Exited (0) 7 minutes
ago           admiring_colden
057d044acdcb   ubuntu    "bash"                  20 minutes ago  Up 6 minutes
              musing_ride
5952c1028763   ubuntu    "bash"                  21 minutes ago  Exited (0) 21 minutes
ago           silly_noyce
0f625a3b6f69   ubuntu    "bash"                  23 minutes ago  Up 5 minutes
              silly_ellis
f62123ed8d68   feb5d9fea6a5  "/hello"              2 hours ago     Exited (0) 2 hours ago
              hungry_hamilton
09603a7cc2bb   feb5d9fea6a5  "/hello"              2 hours ago     Exited (0) 2 hours ago
              keen_rosalind
58f57bb9913f   feb5d9fea6a5  "/hello"              2 hours ago     Exited (0) 2 hours ago
              determined_ishizaka
057165d214ed   feb5d9fea6a5  "/hello"              4 weeks ago     Exited (0) 4 weeks ago
              goofy_hertz
PS C:\Users\mao\Desktop> docker stop sad_elion
sad_elion
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS         NAMES
43c96b66638b  ubuntu    "bash"                  10 minutes ago  Exited (137) 55
seconds ago    stupefied_sinoussi

```

```

5065bd9f0811  ubuntu      "bash"      10 minutes ago  Up 2 minutes
                sleepy_mahavira
2e6e0a4e3a1d  ubuntu      "bash"      10 minutes ago  Up 10 minutes
                modest_visvesvaraya
f8d05f9740a4  ubuntu      "bash"      11 minutes ago  Exited (137) 2 seconds
ago          sad_elion
cb7c2e3d6b9f  ubuntu      "bash"      12 minutes ago  Exited (0) 12 minutes
ago          zealous_kirch
a6b5661d143d  ubuntu      "bash"      13 minutes ago  Exited (0) 7 minutes
ago          admiring_colden
057d044acdcb  ubuntu      "bash"      20 minutes ago  Up 7 minutes
                musing_ride
5952c1028763  ubuntu      "bash"      21 minutes ago  Exited (0) 21 minutes
ago          silly_noyce
0f625a3b6f69  ubuntu      "bash"      23 minutes ago  Up 5 minutes
                silly_ellis
f62123ed8d68  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
                hungry_hamilton
09603a7cc2bb  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
                keen_rosalind
58f57bb9913f  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
                determined_ishizaka
057165d214ed  feb5d9fea6a5  "/hello"    4 weeks ago     Exited (0) 4 weeks ago
                goofy_hertz
PS C:\Users\mao\Desktop>

```

强制停止容器

命令：docker kill 容器id或者容器名

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
43c96b66638b   ubuntu    "bash"                  11 minutes ago Exited (137) 2 minutes
ago          stupefied_sinoussi
5065bd9f0811   ubuntu    "bash"                  11 minutes ago Up 3 minutes
                sleepy_mahavira
2e6e0a4e3a1d   ubuntu    "bash"                  11 minutes ago Up 11 minutes
                modest_visvesvaraya
f8d05f9740a4   ubuntu    "bash"                  12 minutes ago Exited (137) About a
minute ago   sad_elion
cb7c2e3d6b9f   ubuntu    "bash"                  14 minutes ago Exited (0) 13 minutes
ago          zealous_kirch
a6b5661d143d   ubuntu    "bash"                  14 minutes ago Exited (0) 9 minutes
ago          admiring_colden
057d044acdcb   ubuntu    "bash"                  22 minutes ago Up 8 minutes
                musing_ride
5952c1028763   ubuntu    "bash"                  23 minutes ago Exited (0) 23 minutes
ago          silly_noyce
0f625a3b6f69   ubuntu    "bash"                  25 minutes ago Up 7 minutes
                silly_ellis
f62123ed8d68   feb5d9fea6a5  "/hello"                2 hours ago     Exited (0) 2 hours ago
                hungry_hamilton
09603a7cc2bb   feb5d9fea6a5  "/hello"                2 hours ago     Exited (0) 2 hours ago
                keen_rosalind

```

```

58f57bb9913f  feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours ago
determined_ishizaka
057165d214ed  feb5d9fea6a5  "/hello"  4 weeks ago  Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop> docker kill 5065bd9f0811
5065bd9f0811
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
43c96b66638b  ubuntu    "bash"                  11 minutes ago Exited (137) 2 minutes
ago           stupefied_sinoussi
5065bd9f0811  ubuntu    "bash"                  12 minutes ago Exited (137) 4 seconds
ago           sleepy_mahavira
2e6e0a4e3a1d  ubuntu    "bash"                  12 minutes ago Up 12 minutes
modest_visvesvaraya
f8d05f9740a4  ubuntu    "bash"                  13 minutes ago Exited (137) About a
minute ago    sad_elion
cb7c2e3d6b9f  ubuntu    "bash"                  14 minutes ago Exited (0) 14 minutes
ago           zealous_kirch
a6b5661d143d  ubuntu    "bash"                  15 minutes ago Exited (0) 9 minutes
ago           admiring_colden
057d044acdcb  ubuntu    "bash"                  22 minutes ago Up 9 minutes
musing_ride
5952c1028763  ubuntu    "bash"                  23 minutes ago Exited (0) 23 minutes
ago           silly_noyce
0f625a3b6f69  ubuntu    "bash"                  25 minutes ago Up 7 minutes
silly_ellis
f62123ed8d68  feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours ago
hungry_hamilton
09603a7cc2bb  feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours ago
keen_rosalind
58f57bb9913f  feb5d9fea6a5  "/hello"  2 hours ago  Exited (0) 2 hours ago
determined_ishizaka
057165d214ed  feb5d9fea6a5  "/hello"  4 weeks ago  Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop>

```

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
43c96b66638b  ubuntu    "bash"                  12 minutes ago Exited (137) 3 minutes
ago           stupefied_sinoussi
5065bd9f0811  ubuntu    "bash"                  12 minutes ago Exited (137) 23
seconds ago    sleepy_mahavira
2e6e0a4e3a1d  ubuntu    "bash"                  12 minutes ago Up 12 minutes
modest_visvesvaraya
f8d05f9740a4  ubuntu    "bash"                  13 minutes ago Exited (137) 2 minutes
ago           sad_elion
cb7c2e3d6b9f  ubuntu    "bash"                  14 minutes ago Exited (0) 14 minutes
ago           zealous_kirch
a6b5661d143d  ubuntu    "bash"                  15 minutes ago Exited (0) 10 minutes
ago           admiring_colden
057d044acdcb  ubuntu    "bash"                  22 minutes ago Up 9 minutes
musing_ride

```

```

5952c1028763  ubuntu      "bash"      24 minutes ago  Exited (0) 24 minutes ago
0f625a3b6f69  ubuntu      "bash"      25 minutes ago  Up 7 minutes
f62123ed8d68  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
09603a7cc2bb  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
58f57bb9913f  feb5d9fea6a5  "/hello"    2 hours ago     Exited (0) 2 hours ago
057165d214ed  feb5d9fea6a5  "/hello"    4 weeks ago     Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop> docker kill modest_visvesvaraya
modest_visvesvaraya
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS         NAMES
43c96b66638b  ubuntu    "bash"                  12 minutes ago  Exited (137) 3 minutes ago
5065bd9f0811  ubuntu    "bash"                  13 minutes ago  Exited (137) About a minute ago
2e6e0a4e3a1d  ubuntu    "bash"                  13 minutes ago  Exited (137) 2 seconds ago
f8d05f9740a4  ubuntu    "bash"                  14 minutes ago  Exited (137) 2 minutes ago
cb7c2e3d6b9f  ubuntu    "bash"                  15 minutes ago  Exited (0) 15 minutes ago
a6b5661d143d  ubuntu    "bash"                  16 minutes ago  Exited (0) 10 minutes ago
057d044acdcb  ubuntu    "bash"                  23 minutes ago  Up 9 minutes
5952c1028763  ubuntu    "bash"                  24 minutes ago  Exited (0) 24 minutes ago
0f625a3b6f69  ubuntu    "bash"                  26 minutes ago  Up 8 minutes
f62123ed8d68  feb5d9fea6a5  "/hello"              2 hours ago     Exited (0) 2 hours ago
09603a7cc2bb  feb5d9fea6a5  "/hello"              2 hours ago     Exited (0) 2 hours ago
58f57bb9913f  feb5d9fea6a5  "/hello"              2 hours ago     Exited (0) 2 hours ago
057165d214ed  feb5d9fea6a5  "/hello"              4 weeks ago     Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop>

```

删除已停止的容器

命令: docker rm 容器id

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS
PORTS         NAMES

```

```

5065bd9f0811  ubuntu      "bash"      4 days ago  Exited (137) 4 days ago
sleepy_mahavira
2e6e0a4e3a1d  ubuntu      "bash"      4 days ago  Exited (137) 4 days ago
modest_visvesvaraya
f8d05f9740a4  ubuntu      "bash"      4 days ago  Exited (137) 4 days ago
sad_elion
cb7c2e3d6b9f  ubuntu      "bash"      4 days ago  Exited (0) 4 days ago
zealous_kirch
a6b5661d143d  ubuntu      "bash"      4 days ago  Exited (0) 4 days ago
admiring_colden
057d044acdc b  ubuntu      "bash"      4 days ago  Exited (255) 4 days ago
musing_ride
5952c1028763  ubuntu      "bash"      4 days ago  Exited (0) 4 days ago
silly_noyce
0f625a3b6f69  ubuntu      "bash"      4 days ago  Exited (255) 4 days ago
silly_ellis
f62123ed8d68  feb5d9fea6a5  "/hello"    4 days ago  Exited (0) 4 days ago
hungry_hamilton
09603a7cc2bb  feb5d9fea6a5  "/hello"    4 days ago  Exited (0) 4 days ago
keen_rosalind
58f57bb9913f  feb5d9fea6a5  "/hello"    4 days ago  Exited (0) 4 days ago
determined_ishizaka
057165d214ed  feb5d9fea6a5  "/hello"    4 weeks ago Exited (0) 4 weeks ago
goofy_hertz

```

```

PS C:\Users\mao\Desktop> docker rm cb7c2e3d6b9f
cb7c2e3d6b9f

```

```

PS C:\Users\mao\Desktop> docker ps -a

```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|---------------|--------------|----------|-------------|-------------------------|
| 5065bd9f0811 | ubuntu | "bash" | 4 days ago | Exited (137) 4 days ago |
| 2e6e0a4e3a1d | ubuntu | "bash" | 4 days ago | Exited (137) 4 days ago |
| f8d05f9740a4 | ubuntu | "bash" | 4 days ago | Exited (137) 4 days ago |
| a6b5661d143d | ubuntu | "bash" | 4 days ago | Exited (0) 4 days ago |
| 057d044acdc b | ubuntu | "bash" | 4 days ago | Exited (255) 4 days ago |
| 5952c1028763 | ubuntu | "bash" | 4 days ago | Exited (0) 4 days ago |
| 0f625a3b6f69 | ubuntu | "bash" | 4 days ago | Exited (255) 4 days ago |
| f62123ed8d68 | feb5d9fea6a5 | "/hello" | 4 days ago | Exited (0) 4 days ago |
| 09603a7cc2bb | feb5d9fea6a5 | "/hello" | 4 days ago | Exited (0) 4 days ago |
| 58f57bb9913f | feb5d9fea6a5 | "/hello" | 4 days ago | Exited (0) 4 days ago |
| 057165d214ed | feb5d9fea6a5 | "/hello" | 4 weeks ago | Exited (0) 4 weeks ago |

```

PS C:\Users\mao\Desktop> docker rm f62123ed8d68
f62123ed8d68

```

```

PS C:\Users\mao\Desktop> docker ps -a

```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------|---------|------------|-------------------------|
| 5065bd9f0811 | ubuntu | "bash" | 4 days ago | Exited (137) 4 days ago |

```

2e6e0a4e3a1d  ubuntu      "bash"      4 days ago  Exited (137) 4 days ago
modest_visvesvaraya
f8d05f9740a4  ubuntu      "bash"      4 days ago  Exited (137) 4 days ago
sad_elion
a6b5661d143d  ubuntu      "bash"      4 days ago  Exited (0) 4 days ago
admiring_colden
057d044acdcb  ubuntu      "bash"      4 days ago  Exited (255) 4 days ago
musing_ride
5952c1028763  ubuntu      "bash"      4 days ago  Exited (0) 4 days ago
silly_noyce
0f625a3b6f69  ubuntu      "bash"      4 days ago  Exited (255) 4 days ago
silly_ellis
09603a7cc2bb  feb5d9fea6a5  "/hello"    4 days ago  Exited (0) 4 days ago
keen_rosalind
58f57bb9913f  feb5d9fea6a5  "/hello"    4 days ago  Exited (0) 4 days ago
determined_ishizaka
057165d214ed  feb5d9fea6a5  "/hello"    4 weeks ago Exited (0) 4 weeks ago
goofy_hertz
PS C:\Users\mao\Desktop>

```

启动守护式容器

命令：docker run -d 容器名

#使用镜像centos:latest以后台模式启动一个容器

```
docker run -d centos
```

问题：然后docker ps -a 进行查看, 会发现容器已经退出

很重要的要说明的一点: Docker容器后台运行,就必须有一个前台进程.

容器运行的命令如果不是那些一直挂起的命令（比如运行top，tail），就是会自动退出的。

这个是docker的机制问题,比如你的web容器,我们以nginx为例，正常情况下,

我们配置启动服务只需要启动响应的service即可。例如service nginx start

但是,这样做,nginx为后台进程模式运行,就导致docker前台没有运行的应用,

这样的容器后台启动后,会立即自杀因为他觉得他没事可做了。

所以，最佳的解决方案是,将你要运行的程序以前台进程的形式运行，

常见就是命令行模式，表示我还有交互操作，别中断

```

PS C:\Users\mao\Desktop> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
redis         latest    53aa81e8adfa   7 days ago    117MB
ubuntu        latest    d2e4e1f51132   5 weeks ago   77.8MB
PS C:\Users\mao\Desktop> docker run -p6389:6379 -d redis
a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11
PS C:\Users\mao\Desktop> docker ps -a

```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|------------------------|--------------------------|----------------|-----------------------|
| | PORTS | NAMES | | |
| a7275991a5f8 | redis | "docker-entrypoint.s..." | 10 seconds ago | Up 9 seconds |
| | 0.0.0.0:6389->6379/tcp | nostalgic_zhukovsky | | |
| 5952c1028763 | ubuntu | "bash" | 4 days ago | Exited (0) 4 days ago |
| | | silly_noyce | | |

```

PS C:\Users\mao\Desktop> redis-cli -p 6389
127.0.0.1:6389> ping
PONG
127.0.0.1:6389> exit
PS C:\Users\mao\Desktop>

```

查看容器日志

命令: docker logs 容器ID

| | |
|---|--|
| PS C:\Users\mao\Desktop> docker ps -a | |
| | CONTAINER ID IMAGE COMMAND CREATED STATUS |
| | PORTS NAMES |
| a7275991a5f8 | redis "docker-entrypoint.s..." 6 minutes ago Exited (0) |
| About a minute ago | nostalgic_zhukovsky |
| 5952c1028763 | ubuntu "bash" 4 days ago Exited (0) 4 days ago |
| | silly_noyce |
| PS C:\Users\mao\Desktop> docker logs a7275991a5f8 | |
| 1:C 05 Jun 2022 06:14:56.388 # | oO0oO0OoO0Oo Redis is starting oO0oO0OoO0Oo |
| 1:C 05 Jun 2022 06:14:56.389 # | Redis version=7.0.0, bits=64, commit=00000000, modified=0, pid=1, just started |
| 1:C 05 Jun 2022 06:14:56.389 # | Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf |
| 1:M 05 Jun 2022 06:14:56.389 * | monotonic clock: POSIX clock_gettime |
| 1:M 05 Jun 2022 06:14:56.390 * | Running mode=standalone, port=6379. |
| 1:M 05 Jun 2022 06:14:56.390 # | Server initialized |
| 1:M 05 Jun 2022 06:14:56.390 # | WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect. |
| 1:M 05 Jun 2022 06:14:56.391 * | The AOF directory appendonlydir doesn't exist |
| 1:M 05 Jun 2022 06:14:56.391 * | Ready to accept connections |
| 1:signal-handler (1654409781) | Received SIGTERM scheduling shutdown... |
| 1:M 05 Jun 2022 06:16:21.632 # | User requested shutdown... |
| 1:M 05 Jun 2022 06:16:21.632 * | Saving the final RDB snapshot before exiting. |
| 1:M 05 Jun 2022 06:16:21.634 * | DB saved on disk |
| 1:M 05 Jun 2022 06:16:21.634 # | Redis is now ready to exit, bye bye... |
| 1:C 05 Jun 2022 06:16:46.592 # | oO0oO0OoO0Oo Redis is starting oO0oO0OoO0Oo |
| 1:C 05 Jun 2022 06:16:46.593 # | Redis version=7.0.0, bits=64, commit=00000000, modified=0, pid=1, just started |
| 1:C 05 Jun 2022 06:16:46.593 # | Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf |
| 1:M 05 Jun 2022 06:16:46.593 * | monotonic clock: POSIX clock_gettime |
| 1:M 05 Jun 2022 06:16:46.593 * | Running mode=standalone, port=6379. |
| 1:M 05 Jun 2022 06:16:46.593 # | Server initialized |


```
1:M 05 Jun 2022 06:16:46.593 # WARNING overcommit_memory is set to 0! Background
save may fail under low memory condition. To fix this issue add
'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command
'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 05 Jun 2022 06:16:46.594 * The AOF directory appendonlydir doesn't exist
1:M 05 Jun 2022 06:16:46.594 * Loading RDB produced by version 7.0.0
1:M 05 Jun 2022 06:16:46.594 * RDB age 25 seconds
1:M 05 Jun 2022 06:16:46.594 * RDB memory usage when created 0.87 Mb
1:M 05 Jun 2022 06:16:46.594 * Done loading RDB, keys loaded: 0, keys expired:
0.
1:M 05 Jun 2022 06:16:46.594 * DB loaded from disk: 0.000 seconds
1:M 05 Jun 2022 06:16:46.594 * Ready to accept connections
1:signal-handler (1654409818) Received SIGTERM scheduling shutdown...
1:M 05 Jun 2022 06:16:58.633 # User requested shutdown...
1:M 05 Jun 2022 06:16:58.633 * Saving the final RDB snapshot before exiting.
1:M 05 Jun 2022 06:16:58.635 * DB saved on disk
1:M 05 Jun 2022 06:16:58.635 # Redis is now ready to exit, bye bye...
1:C 05 Jun 2022 06:17:16.302 # 0000o000o000o Redis is starting o000o000o000o
1:C 05 Jun 2022 06:17:16.302 # Redis version=7.0.0, bits=64, commit=00000000,
modified=0, pid=1, just started
1:C 05 Jun 2022 06:17:16.302 # warning: no config file specified, using the
default config. In order to specify a config file use redis-server
/path/to/redis.conf
1:M 05 Jun 2022 06:17:16.303 * monotonic clock: POSIX clock_gettime
1:M 05 Jun 2022 06:17:16.303 * Running mode=standalone, port=6379.
1:M 05 Jun 2022 06:17:16.303 # Server initialized
1:M 05 Jun 2022 06:17:16.303 # WARNING overcommit_memory is set to 0! Background
save may fail under low memory condition. To fix this issue add
'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command
'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 05 Jun 2022 06:17:16.304 * The AOF directory appendonlydir doesn't exist
1:M 05 Jun 2022 06:17:16.304 * Loading RDB produced by version 7.0.0
1:M 05 Jun 2022 06:17:16.304 * RDB age 18 seconds
1:M 05 Jun 2022 06:17:16.304 * RDB memory usage when created 0.82 Mb
1:M 05 Jun 2022 06:17:16.304 * Done loading RDB, keys loaded: 0, keys expired:
0.
1:M 05 Jun 2022 06:17:16.304 * DB loaded from disk: 0.000 seconds
1:M 05 Jun 2022 06:17:16.304 * Ready to accept connections
1:signal-handler (1654409962) Received SIGTERM scheduling shutdown...
1:M 05 Jun 2022 06:19:22.919 # User requested shutdown...
1:M 05 Jun 2022 06:19:22.920 * Saving the final RDB snapshot before exiting.
1:M 05 Jun 2022 06:19:22.922 * DB saved on disk
1:M 05 Jun 2022 06:19:22.922 # Redis is now ready to exit, bye bye...
1:C 05 Jun 2022 06:20:03.271 # 0000o000o000o Redis is starting o000o000o000o
1:C 05 Jun 2022 06:20:03.271 # Redis version=7.0.0, bits=64, commit=00000000,
modified=0, pid=1, just started
1:C 05 Jun 2022 06:20:03.271 # warning: no config file specified, using the
default config. In order to specify a config file use redis-server
/path/to/redis.conf
1:M 05 Jun 2022 06:20:03.271 * monotonic clock: POSIX clock_gettime
1:M 05 Jun 2022 06:20:03.272 * Running mode=standalone, port=6379.
1:M 05 Jun 2022 06:20:03.272 # Server initialized
1:M 05 Jun 2022 06:20:03.272 # WARNING overcommit_memory is set to 0! Background
save may fail under low memory condition. To fix this issue add
'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command
'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 05 Jun 2022 06:20:03.272 * The AOF directory appendonlydir doesn't exist
1:M 05 Jun 2022 06:20:03.272 * Loading RDB produced by version 7.0.0
```



```

1:M 05 Jun 2022 06:20:03.272 * RDB age 41 seconds
1:M 05 Jun 2022 06:20:03.272 * RDB memory usage when created 0.87 Mb
1:M 05 Jun 2022 06:20:03.272 * Done loading RDB, keys loaded: 0, keys expired: 0.
1:M 05 Jun 2022 06:20:03.272 * DB loaded from disk: 0.000 seconds
1:M 05 Jun 2022 06:20:03.272 * Ready to accept connections
1:signal-handler (1654410018) Received SIGTERM scheduling shutdown...
1:M 05 Jun 2022 06:20:18.916 # User requested shutdown...
1:M 05 Jun 2022 06:20:18.916 * Saving the final RDB snapshot before exiting.
1:M 05 Jun 2022 06:20:18.918 * DB saved on disk
1:M 05 Jun 2022 06:20:18.918 # Redis is now ready to exit, bye bye...
PS C:\Users\mao\Desktop>

```

查看容器内运行的进程

命令: docker top 容器ID

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
a7275991a5f8   redis     "docker-entrypoint.s..." 8 minutes ago  Exited (0) 3
minutes ago   nostalgic_zhukovsky
5952c1028763   ubuntu    "bash"                   4 days ago    Exited (0) 4
days ago     silly_noyce
PS C:\Users\mao\Desktop> docker top a7275991a5f8
Error response from daemon: Container
a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11 is not running
PS C:\Users\mao\Desktop> docker start a7275991a5f8
a7275991a5f8
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
a7275991a5f8   redis     "docker-entrypoint.s..." 9 minutes ago  Up 4 seconds
0.0.0.0:6389->6379/tcp nostalgic_zhukovsky
5952c1028763   ubuntu    "bash"                   4 days ago    Exited (0) 4
days ago     silly_noyce
PS C:\Users\mao\Desktop> docker top a7275991a5f8
UID          PID    PPID    C
STIME        TTY    TIME    CMD
999          1664   1644    0
06:23        ?      00:00:00 redis-server *:6379
PS C:\Users\mao\Desktop>

```

查看容器内部细节

命令: docker inspect 容器ID

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES

```

```

a7275991a5f8    redis    "docker-entrypoint.s..." 10 minutes ago    Up About a
minute        0.0.0.0:6389->6379/tcp    nostalgic_zhukovsky
5952c1028763    ubuntu    "bash"                4 days ago        Exited (0) 4
days ago                                           silly_noyce
PS C:\Users\mao\Desktop> docker inspect a7275991a5f8
[
  {
    "Id":
"a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11",
    "Created": "2022-06-05T06:14:55.905008518Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "redis-server"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 1664,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2022-06-05T06:23:57.563259974Z",
      "FinishedAt": "2022-06-05T06:20:18.920939498Z"
    },
    "Image":
"sha256:53aa81e8adfa939348cd4c846c0ab682b16dc7641714e36bfc57b764f0b947dc",
    "ResolveConfPath":
"/var/lib/docker/containers/a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11/resolve.conf",
    "HostnamePath":
"/var/lib/docker/containers/a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11/hostname",
    "HostsPath":
"/var/lib/docker/containers/a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11/hosts",
    "LogPath":
"/var/lib/docker/containers/a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11/a7275991a5f8ca607a27bb46c920e9cd0efbda2eeb5e1f08323a621e1a005d11-
json.log",
    "Name": "/nostalgic_zhukovsky",
    "RestartCount": 0,
    "Driver": "overlay2",
    "Platform": "linux",
    "MountLabel": "",
    "ProcessLabel": "",
    "AppArmorProfile": "",
    "ExecIDs": null,
    "HostConfig": {
      "Binds": null,
      "ContainerIDFile": "",
      "LogConfig": {
        "Type": "json-file",
        "Config": {}
      },
      "NetworkMode": "default",

```

```
"PortBindings": {
  "6379/tcp": [
    {
      "HostIp": "",
      "HostPort": "6389"
    }
  ]
},
"RestartPolicy": {
  "Name": "no",
  "MaximumRetryCount": 0
},
"AutoRemove": false,
"VolumeDriver": "",
"VolumesFrom": null,
"CapAdd": null,
"CapDrop": null,
"CgroupnsMode": "host",
"Dns": [],
"DnsOptions": [],
"DnsSearch": [],
"ExtraHosts": null,
"GroupAdd": null,
"IpcMode": "private",
"Cgroup": "",
"Links": null,
"OomScoreAdj": 0,
"PidMode": "",
"Privileged": false,
"PublishAllPorts": false,
"ReadonlyRootfs": false,
"SecurityOpt": null,
"UTSMode": "",
"UsernsMode": "",
"ShmSize": 67108864,
"Runtime": "runc",
"ConsoleSize": [
  50,
  160
],
"Isolation": "",
"CpuShares": 0,
"Memory": 0,
"NanoCpus": 0,
"CgroupParent": "",
"Blkioweight": 0,
"BlkioweightDevice": [],
"BlkiodeviceReadBps": null,
"BlkiodeviceWriteBps": null,
"BlkiodeviceReadIops": null,
"BlkiodeviceWriteIops": null,
"CpuPeriod": 0,
"CpuQuota": 0,
"CpuRealtimePeriod": 0,
"CpuRealtimeRuntime": 0,
"CpusetCpus": "",
"CpusetMems": "",
"Devices": [],
```

```

        "DeviceCgroupRules": null,
        "DeviceRequests": null,
        "KernelMemory": 0,
        "KernelMemoryTCP": 0,
        "MemoryReservation": 0,
        "MemorySwap": 0,
        "MemorySwappiness": null,
        "OomKillDisable": false,
        "PidsLimit": null,
        "Ulimits": null,
        "CpuCount": 0,
        "CpuPercent": 0,
        "IOMaximumIops": 0,
        "IOMaximumBandwidth": 0,
        "MaskedPaths": [
            "/proc/asound",
            "/proc/acpi",
            "/proc/kcore",
            "/proc/keys",
            "/proc/latency_stats",
            "/proc/timer_list",
            "/proc/timer_stats",
            "/proc/sched_debug",
            "/proc/scsi",
            "/sys/firmware"
        ],
        "ReadOnlyPaths": [
            "/proc/bus",
            "/proc/fs",
            "/proc/irq",
            "/proc/sys",
            "/proc/sysrq-trigger"
        ]
    },
    "GraphDriver": {
        "Data": {
            "LowerDir":
"/var/lib/docker/overlay2/13ef8274b81a6c8f26ac06895ac304c739946c2fec8e1afb91b9c9cc3ab65f0f-init/diff:/var/lib/docker/overlay2/554d073f62c69fe5ddf8714516024f5ebbd351d52b678ba42d8d23033fa78c0b/diff:/var/lib/docker/overlay2/6eae66b49bdbb91a82f0c570a68e46060541bb449cb23fb280c5e8ed33160c6c/diff:/var/lib/docker/overlay2/c12ba5c77a448628b25da9a453c4b0a6b7f7495e54695e8c9691deaaf57a3772/diff:/var/lib/docker/overlay2/4fc5482dfe8a9747a641e624dc32314b7721176f8b32bfc231d864fd3b2f0729/diff:/var/lib/docker/overlay2/91211f126b80c5a64623113c4b5fba230fbdf27a90780597225b255d72f589f6/diff:/var/lib/docker/overlay2/aa45b050ad99ffd7c424550d337ecfa68105d5c946443933f257ab55ec5e86d0/diff",
            "MergedDir":
"/var/lib/docker/overlay2/13ef8274b81a6c8f26ac06895ac304c739946c2fec8e1afb91b9c9cc3ab65f0f/merged",
            "UpperDir":
"/var/lib/docker/overlay2/13ef8274b81a6c8f26ac06895ac304c739946c2fec8e1afb91b9c9cc3ab65f0f/diff",
            "WorkDir":
"/var/lib/docker/overlay2/13ef8274b81a6c8f26ac06895ac304c739946c2fec8e1afb91b9c9cc3ab65f0f/work"
        },
        "Name": "overlay2"
    }

```

```

    },
    "Mounts": [
      {
        "Type": "volume",
        "Name":
"d3f51821d9173e29f4d17fbb2c34f080819ba7af04926d45735efab95258d7c3",
        "Source":
"/var/lib/docker/volumes/d3f51821d9173e29f4d17fbb2c34f080819ba7af04926d45735efab95258d7c3/_data",
        "Destination": "/data",
        "Driver": "local",
        "Mode": "",
        "RW": true,
        "Propagation": ""
      }
    ],
    "Config": {
      "Hostname": "a7275991a5f8",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "ExposedPorts": {
        "6379/tcp": {}
      },
      "Tty": false,
      "OpenStdin": false,
      "StdinOnce": false,
      "Env": [

"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
        "GOSU_VERSION=1.14",
        "REDIS_VERSION=7.0.0",
        "REDIS_DOWNLOAD_URL=http://download.redis.io/releases/redis-7.0.0.tar.gz",

"REDIS_DOWNLOAD_SHA=284d8bd1fd85d6a55a05ee4e7c31c31977ad56cbf344ed83790beeb148baa720"
      ],
      "Cmd": [
        "redis-server"
      ],
      "Image": "redis",
      "Volumes": {
        "/data": {}
      },
      "WorkingDir": "/data",
      "Entrypoint": [
        "docker-entrypoint.sh"
      ],
      "OnBuild": null,
      "Labels": {}
    },
    "NetworkSettings": {
      "Bridge": "",
      "SandboxID":
"83e118569bb5bfc82b4373c053814d3c6d980be3e7489805c9da06b5a23e1fda",

```

```

    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {
      "6379/tcp": [
        {
          "HostIp": "0.0.0.0",
          "HostPort": "6389"
        }
      ]
    },
    "SandboxKey": "/var/run/docker/netns/83e118569bb5",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID":
"07cd994deaa2d32bf5d28c81cb5b157fe081da4648e914534a9c1626b15363a3",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:02",
    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID":
"f98ca8a13ff8439242393c0d1a517feb40fa821c712b68669df607373e2a5f7e",
        "EndpointID":
"07cd994deaa2d32bf5d28c81cb5b157fe081da4648e914534a9c1626b15363a3",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:11:00:02",
        "DriverOpts": null
      }
    }
  }
}
]
PS C:\Users\mao\Desktop>

```

进入正在运行的容器并以命令行交互

命令: docker exec -it 容器ID bashShell

```
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------|--------------------------|----------------|-----------------------|
| a7275991a5f8 | redis | "docker-entrypoint.s..." | 13 minutes ago | Up 4 minutes |
| 5952c1028763 | ubuntu | "bash" | 4 days ago | Exited (0) 4 days ago |

```

PS C:\Users\mao\Desktop> docker exec -it a7275991a5f8 /bin/bash
root@a7275991a5f8:/data# ls
dump.rdb
root@a7275991a5f8:/data# cd ../../
root@a7275991a5f8:/# ls
bin boot data dev etc home lib lib64 media mnt opt proc root run
sbin srv sys tmp usr var
root@a7275991a5f8:/# exit
exit
PS C:\Users\mao\Desktop>

```

重新进入：

docker attach 容器ID

区别：

attach 直接进入容器启动命令的终端，不会启动新的进程，exit退出，会导致容器的停止。

exec 是在容器中打开新的终端，并且可以启动新的进程，exit退出，不会导致容器的停止。

从容器内拷贝文件到主机上

docker cp 容器ID:容器内路径 目的主机路径

export ： 导出容器的内容留作为一个tar归档文件[对应import命令]

import ： 从tar包中的内容创建一个新的文件系统再导入为镜像[对应export]

```

PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS          NAMES
a7275991a5f8   redis    "docker-entrypoint.s..." 26 minutes ago Up 7 minutes
0.0.0.0:6389->6379/tcp nostalgic_zhukovsky
5952c1028763   ubuntu   "bash"                   4 days ago    Exited (0) 4
days ago      silly_noyce
PS C:\Users\mao\Desktop> docker export a7275991a5f8 a.tar
"docker export" requires exactly 1 argument.
See 'docker export --help'.

Usage:  docker export [OPTIONS] CONTAINER

```

```
Export a container's filesystem as a tar archive
PS C:\Users\mao\Desktop> docker export a7275991a5f8 > a.tar
PS C:\Users\mao\Desktop> ls
```

目录: C:\Users\mao\Desktop

| Mode | LastWriteTime | Length | Name |
|--------|----------------|-----------|-------|
| -a---- | 2022/6/5 14:43 | 204765792 | a.tar |

Docker 镜像

是什么?

是一种轻量级、可执行的独立软件包，它包含运行某个软件所需的所有内容，我们把应用程序和配置依赖打包好形成一个可交付的运行环境(包括代码、运行时需要的库、环境变量和配置文件等)，这个打包好的运行环境就是image镜像文件。

只有通过这个镜像文件才能生成Docker容器实例(类似Java中新出来一个对象)。

联合文件系统

UnionFS (联合文件系统)：Union文件系统 (UnionFS) 是一种分层、轻量级并且高性能的文件系统，它支持对文件系统的修改作为一次提交来一层层的叠加，同时可以将不同目录挂载到同一个虚拟文件系统下(unite several directories into a single virtual filesystem)。Union 文件系统是 Docker 镜像的基础。镜像可以通过分层来进行继承，基于基础镜像（没有父镜像），可以制作各种具体的应用镜像。

特性：一次同时加载多个文件系统，但从外面看起来，只能看到一个文件系统，联合加载会把各层文件系统叠加起来，这样最终的文件系统会包含所有底层的文件和目录

Docker镜像加载原理

docker的镜像实际上由一层一层的文件系统组成，这种层级的文件系统UnionFS。

bootfs(boot file system)主要包含bootloader和kernel, bootloader主要是引导加载kernel, Linux刚启动时会加载bootfs文件系统，在Docker镜像的最底层是引导文件系统bootfs。这一层与我们典型的Linux/Unix系统是一样的，包含boot加载器和内核。当boot加载完成之后整个内核就都在内存中了，此时内存的使用权已由bootfs转交给内核，此时系统也会卸载bootfs。

rootfs (root file system)，在bootfs之上。包含的就是典型Linux系统中的/dev, /proc, /bin, /etc等标准目录和文件。rootfs就是各种不同的操作系统发行版，比如Ubuntu, Centos等等。

对于一个精简的OS, rootfs可以很小，只需要包括最基本的命令、工具和程序库就可以了，因为底层直接用Host的kernel，自己只需要提供rootfs就行了。由此可见对于不同的linux发行版, bootfs基本是一致的, rootfs会有差别, 因此不同的发行版可以公用bootfs。

镜像分层最大的一个好处就是共享资源，方便复制迁移，就是为了复用。

比如说有多个镜像都从相同的 base 镜像构建而来，那么 Docker Host 只需在磁盘上保存一份 base 镜像；

同时内存中也只需加载一份 base 镜像，就可以为所有容器服务了。而且镜像的每一层都可以被共享。

Docker 镜像层都是只读的，容器层是可写的

当容器启动时，一个新的可写层被加载到镜像的顶部。

这一层通常被称作“容器层”，“容器层”之下的都叫“镜像层”。

当容器启动时，一个新的可写层被加载到镜像的顶部。这一层通常被称作“容器层”，“容器层”之下的都叫“镜像层”。

所有对容器的改动 - 无论添加、删除、还是修改文件都只会发生在容器层中。只有容器层是可写的，容器层下面的所有镜像层都是只读的。

docker commit

docker commit 提交容器副本使之成为一个新的镜像

命令：docker commit -m="提交的描述信息" -a="作者" 容器ID 要创建的目标镜像名:[标签名]

```
PS C:\Users\mao\Desktop> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS          NAMES
a7275991a5f8   redis     "docker-entrypoint.s..." 39 minutes ago Up 19 minutes
0.0.0.0:6379->tcp nostalgic_zhukovsky
5952c1028763   ubuntu    "bash"                  4 days ago    Exited (0) 4
days ago      silly_noyce
PS C:\Users\mao\Desktop> docker stop nostalgic_zhukovsky
nostalgic_zhukovsky
PS C:\Users\mao\Desktop> docker commit -m="abc" -a="mao" a7275991a5f8
myredis:1.0
sha256:be89dd503a5f6160a49385b8c78ce72cb354403fa5e6e45e23f2270018092fb7
PS C:\Users\mao\Desktop> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
myredis       1.0       be89dd503a5f   9 seconds ago  117MB
redis         latest    53aa81e8adfa   7 days ago     117MB
ubuntu        latest    d2e4e1f51132   5 weeks ago    77.8MB
PS C:\Users\mao\Desktop> docker run -d myredis
Unable to find image 'myredis:latest' locally
docker: Error response from daemon: pull access denied for myredis, repository
does not exist or may require 'docker login': denied: requested access to the
resource is denied.
```

```
See 'docker run --help'.
PS C:\Users\mao\Desktop> docker run -d myredis:1.0
88cb42a3ce4452e42735008a5460d0c0ead2499c7ed86569d84c9f6effd86870
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|-------------|--------------------------|----------------|--------------------------|
| | PORTS | NAMES | | |
| 88cb42a3ce44 | myredis:1.0 | "docker-entrypoint.s..." | 11 seconds ago | Up 10 seconds |
| a7275991a5f8 | redis | "docker-entrypoint.s..." | 41 minutes ago | Exited (0) 2 minutes ago |
| 5952c1028763 | ubuntu | "bash" | 4 days ago | Exited (0) 4 days ago |

```
PS C:\Users\mao\Desktop>
```

Docker 容器数据卷

命令：docker run -it --privileged=true -v 宿主机绝对路径目录:/容器内目录 镜像名

--privileged=true:

Docker挂载主机目录访问如果出现cannot open directory .: Permission denied

使用--privileged=true可以解决此问题

```
docker run -it --privileged=true -p6380:6379 --name redis1 -v
H:/Docker/redis:/redis redis
```

读写:

命令：docker run -it --privileged=true -v /宿主机绝对路径目录:/容器内目录:rw 镜像名

只读:

命令：docker run -it --privileged=true -v /宿主机绝对路径目录:/容器内目录:ro 镜像名

常用安装

安装Tomcat

1. 搜索镜像

命令: docker search tomcat

```
PS C:\Users\mao\Desktop> docker search Tomcat
NAME                                STARS      OFFICIAL    AUTOMATED    DESCRIPTION
tomcat                              3341       [OK]        Apache Tomcat is an open source
implementati...
tomEE                               97         [OK]        Apache TomEE is an all-Apache Java
bitnami/tomcat                      45         [OK]        Bitnami Tomcat Docker Image
arm32v7/tomcat                      11         Apache Tomcat is an open source
implementati...
rightctrl/tomcat                    7          CentOS , Oracle Java, tomcat
application ssl...                  7          [OK]
arm64v8/tomcat                      7          Apache Tomcat is an open source
implementati...
jelastic/tomcat                     4          An image of the Tomcat Java
application serv...
amd64/tomcat                        4          Apache Tomcat is an open source
implementati...
tomcat2111/pisignage-server          3          [OK]        PiSignage Server
oobsri/tomcat8                      2          Testing CI Jobs with different
names.
cfje/tomcat-resource                2          Tomcat Concourse Resource
appsvc/tomcat                       1
chenyufeng/tomcat-centos             1          [OK]        tomcat基于centos6的镜像
ppc64le/tomcat                      1          Apache Tomcat is an open source
implementati...
tomcatling/jupyterhub_aws            1
tomcat2111/papercut-mf              0          PaperCut MF Application Server
softwareplant/tomcat                0          [OK]        Tomcat images for jira-cloud
testing
tomcatengineering/pg_backup_rotated  0          Clone of
martianrock/pg_backup_rotated but w...
misolims/miso-base                  0          MySQL 5.7 Database and Tomcat 8
Server neede...
s390x/tomcat                        0          Apache Tomcat is an open source
implementati...
tomcat2111/bitbucket-pipelines-elasticsearch 0          Elasticsearch for Bitbucket's
Pipelines
semoss/docker-tomcat                0          Tomcat, Java, Maven, and Git on
top of debian
tomcat2111/phpredisadmin             0          [OK]        This is a Docker image for
phpredisadmin
secoresearch/tomcat-varnish          0          [OK]        Tomcat and Varnish 5.0
tomcat0823/auto1                    0
```

```
PS C:\Users\mao\Desktop>
```

2. 下载

命令: docker pull tomcat

```
PS C:\Users\mao\Desktop> docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
e756f3fdd6a3: Pull complete
bf168a674899: Pull complete
e604223835cc: Pull complete
6d5c91c4cd86: Pull complete
5e20d165240e: Pull complete
1334d60df9a8: Pull complete
16c2728dcd90: Pull complete
05288798d23d: Pull complete
c022dc2b2581: Pull complete
d86ac2f896ee: Pull complete
Digest: sha256:b4e84cff017ff5202cb760ccb1373dd950158f926d6afb04bd5e9f7337291501
Status: Downloaded newer image for tomcat:latest
docker.io/library/tomcat:latest
PS C:\Users\mao\Desktop>
```

3. 查看是否拉取成功

docker images

```
PS C:\Users\mao\Desktop> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
tomcat        latest    c795915cb678   2 weeks ago   680MB
redis         latest    53aa81e8adfa   2 weeks ago   117MB
mysql         latest    65b636d5542b   2 weeks ago   524MB
ubuntu        latest    d2e4e1f51132   6 weeks ago   77.8MB
PS C:\Users\mao\Desktop>
```

4. 运行

命令: docker run -d --name tomcat1 -p8080:8080 tomcat

命令: docker run -d --name tomcat1 -p8080:8080 --privileged=true -v
H:/Docker/tomcat/:/usr/local/tomcat/webapps tomcat

```
PS C:\Users\mao\Desktop> docker run -d --name tomcat1 -p8080:8080 --privileged=true -v H:/Docker/tomcat:/usr/local/tomcat/webapps tomcat 3ca156e4541d4828906f34700fd690ea9bce0e543117ed27724cea70cf9f5780
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------|--------------------------|----------------|---------------------------|
| 3ca156e4541d | tomcat | "catalina.sh run" | 15 seconds ago | Up 14 seconds |
| 3948921a6099 | redis | "docker-entrypoint.s..." | 52 minutes ago | Exited (0) 11 minutes ago |

```
PS C:\Users\mao\Desktop>
```

5. 测试

访问8080端口

安装mysql

1. 搜索和下载

命令: docker pull mysql

2. 查看是否拉取成功

docker images

```
PS C:\Users\mao\Desktop> docker images
```

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|------------|--------|--------------|-------------|--------|
| tomcat | latest | c795915cb678 | 2 weeks ago | 680MB |
| redis | latest | 53aa81e8adfa | 2 weeks ago | 117MB |
| mysql | latest | 65b636d5542b | 2 weeks ago | 524MB |
| ubuntu | latest | d2e4e1f51132 | 6 weeks ago | 77.8MB |

```
PS C:\Users\mao\Desktop>
```

3. 运行

命令: docker run -p 3307:3306 -e MYSQL_ROOT_PASSWORD=123456 -d mysql

命令:

```
docker run -d -p 3307:3306 --privileged=true -v
H:/Docker/mysql/log:/var/log/mysql -v H:/Docker/mysql/data:/var/lib/mysql -v
H:/Docker/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=123456 --name
mysql1 mysql
```

```
PS C:\Users\mao\Desktop> docker run -d -p 3307:3306 --privileged=true -v
H:/Docker/mysql/log/:/var/log/mysql -v H:/Docker/mysql/data:/var/lib/mysql -v
H:/Docker/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=123456 --name
mysql1 mysql
2d379d342bb6879bef7d9e359cb532f6146c108b26a72fcc2e7085301331d200
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------|--------------------------|-------------------|----------------------------|
| 2d379d342bb6 | mysql | "docker-entrypoint.s..." | 48 seconds ago | Up 47 seconds |
| 3ca156e4541d | tomcat | "catalina.sh run" | 22 minutes ago | Exited (143) 8 minutes ago |
| 3948921a6099 | redis | "docker-entrypoint.s..." | About an hour ago | Exited (0) 34 minutes ago |

```
PS C:\Users\mao\Desktop>
```

4. 在H:\Docker\mysql\conf下创建my.conf

写入一下内容

```
default_character_set=utf8
collation_server = utf8_general_ci
character_set_server = utf8
```

5. 测试

```
PS C:\Users\mao\Desktop> mysql -P 3307 -u root -p
Enter password: *****
welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.29 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.00 sec)
```

```
mysql>
```

6. 验证编码

```
mysql> SHOW VARIABLES LIKE 'character%'
```

```
-> ;
```

| variable_name | value |
|--------------------------|--------------------------------|
| character_set_client | gbk |
| character_set_connection | gbk |
| character_set_database | utf8mb4 |
| character_set_filesystem | binary |
| character_set_results | gbk |
| character_set_server | utf8mb4 |
| character_set_system | utf8mb3 |
| character_sets_dir | /usr/share/mysql-8.0/charsets/ |

8 rows in set (0.01 sec)

7. 数据测试

```
mysql> create database db1
```

```
-> ;
```

Query OK, 1 row affected (0.12 sec)

```
mysql> use db1
```

Database changed

```
mysql> create table test(id int,name char(6));
```

Query OK, 0 rows affected (0.37 sec)

```
mysql> show tables;
```

```
+-----+
```

| Tables_in_db1 |
|---------------|
|---------------|

```
+-----+
```

| |
|------|
| test |
|------|

```
+-----+
```

1 row in set (0.00 sec)

```
mysql> insert into test values(1,"1234");
```

Query OK, 1 row affected (0.09 sec)

```
mysql> insert into test values(1,"你好");
```

Query OK, 1 row affected (0.09 sec)

```
mysql> select * from test;
```

```
+-----+
```

| id | name |
|----|------|
|----|------|

```
+-----+
```

| | |
|---|------|
| 1 | 1234 |
|---|------|

| | |
|---|----|
| 1 | 你好 |
|---|----|

```
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

安装redis

1. 搜索和下载

命令: `docker pull redis`

2. 查看是否拉取成功

`docker images`

```
PS C:\Users\mao\Desktop> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
tomcat        latest    c795915cb678   2 weeks ago    680MB
redis         latest    53aa81e8adfa   2 weeks ago    117MB
mysql         latest    65b636d5542b   2 weeks ago    524MB
ubuntu        latest    d2e4e1f51132   6 weeks ago    77.8MB
PS C:\Users\mao\Desktop>
```

3. 运行

命令:

```
docker run -p 6380:6379 --name redis1 --privileged=true -v
H:/Docker/redis/conf:/etc/redis/ -v H:/Docker/redis/data:/data -d redis redis-
server /etc/redis/redis.conf
```



```
PS C:\Users\mao\Desktop> docker run -p 6380:6379 --name redis1 --privileged=true
-v H:/Docker/redis/conf:/etc/redis/ -v H:/Docker/redis/data:/data -d redis
redis-server /etc/redis/redis.conf
8a80769441281ea5f5872c3656eea04c7728b026baae1ea7a13d540fc4daecc3
PS C:\Users\mao\Desktop> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|--------|--------------------------|----------------|---------------------------|
| 8a8076944128 | redis | "docker-entrypoint.s..." | 31 seconds ago | Exited (1) 28 seconds ago |
| 81902b3cfdc4 | mysql | "docker-entrypoint.s..." | 11 hours ago | Exited (0) 11 hours ago |
| 24ee3e986397 | mysql | "docker-entrypoint.s..." | 11 hours ago | Exited (0) 11 hours ago |
| 2d379d342bb6 | mysql | "docker-entrypoint.s..." | 36 hours ago | Exited (0) 12 hours ago |
| 3ca156e4541d | tomcat | "catalina.sh run" | 37 hours ago | Up About an hour |

```
PS C:\Users\mao\Desktop>
```

```
PS C:\Users\mao\Desktop> docker logs 8a8076944128
1:C 15 Jun 2022 03:06:54.166 # Fatal error, can't open config file
'/etc/redis/redis.conf': No such file or directory
PS C:\Users\mao\Desktop>
```

4. 创建配置文件

在H:\Docker\redis\conf目录下创建redis.conf文件

文件内容可以从其他redis服务器里复制并粘贴

```
PS H:\Docker\redis\conf> ls
```

目录: H:\Docker\redis\conf

| Mode | LastWriteTime | Length | Name |
|--------|-----------------|--------|------------|
| -a---- | 2022/6/15 11:04 | 66028 | redis.conf |

```
PS H:\Docker\redis\conf>
```

```
# pwd
/data
# cd ..
# cd etc
# cd redis
# ls
redis.conf
#
```

5. 重启redis服务器

命令:

```
docker restart redis1
```

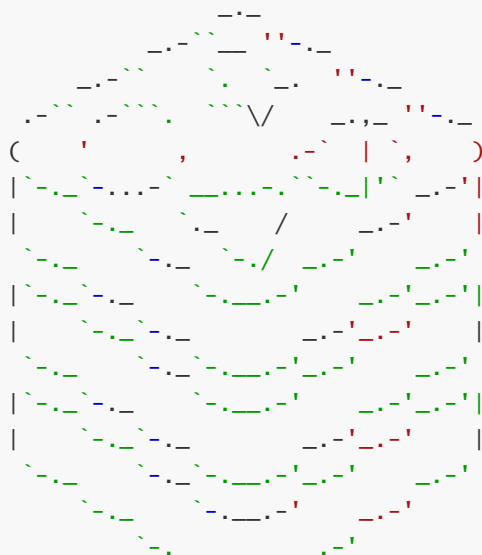
```
PS C:\Users\mao\Desktop> docker restart redis1
redis1
PS C:\Users\mao\Desktop>
```

6. 查看日志

命令:

```
docker logs redis1
```

```
1:C 15 Jun 2022 03:14:13.190 # oOoOoOoOoOoOo Redis is starting oOoOoOoOoOoOo
1:C 15 Jun 2022 03:14:13.190 # Redis version=7.0.0, bits=64, commit=00000000,
modified=0, pid=1, just started
1:C 15 Jun 2022 03:14:13.190 # Configuration loaded
1:M 15 Jun 2022 03:14:13.191 * monotonic clock: POSIX clock_gettime
```



Redis 7.0.0 (00000000/0) 64 bit

Running in standalone mode

Port: 6379

PID: 1

<https://redis.io>

```
1:M 15 Jun 2022 03:14:13.191 # Server initialized
1:M 15 Jun 2022 03:14:13.191 # WARNING overcommit_memory is set to 0! Background
save may fail under low memory condition. To fix this issue add
'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command
'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 15 Jun 2022 03:14:13.192 * The AOF directory appendonlydir doesn't exist
1:M 15 Jun 2022 03:14:13.193 * Loading RDB produced by version 7.0.0
1:M 15 Jun 2022 03:14:13.193 * RDB age 1 seconds
1:M 15 Jun 2022 03:14:13.193 * RDB memory usage when created 0.82 Mb
1:M 15 Jun 2022 03:14:13.193 * Done loading RDB, keys loaded: 0, keys expired:
0.
1:M 15 Jun 2022 03:14:13.193 * DB loaded from disk: 0.002 seconds
1:M 15 Jun 2022 03:14:13.193 * Ready to accept connections
PS C:\Users\mao\Desktop>
```

配置文件成功加载

7. 连接服务器

外部使用命令

```
redis-cli -p 6380
```

```
C:\Users\mao>redis-cli -p 6380
127.0.0.1:6380> ping
PONG
127.0.0.1:6380>
```

或者

```
docker exec -it redis1 /bin/bash
redis-cli
```

```
PS C:\Users\mao\Desktop> docker exec -it redis1 /bin/bash
root@8a8076944128:/data# redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379>
```

8. 测试

数据测试

```
127.0.0.1:6379> set a 1
OK
127.0.0.1:6379> keys *
1) "a"
127.0.0.1:6379> get a
"1"
127.0.0.1:6379>
```

配置文件测试

当前为16个库

```
127.0.0.1:6379> select 15
OK
127.0.0.1:6379[15]> select 16
(error) ERR DB index is out of range
127.0.0.1:6379[15]>
```

更改库的大小，把库的数量更改为2

重启服务器

再次访问

```
127.0.0.1:6380> select 1
OK
127.0.0.1:6380[1]> select 2
(error) ERR DB index is out of range
127.0.0.1:6380[1]>
```

配置文件有效

DockerFile

是什么？

Dockerfile是用来构建Docker镜像的文本文件，是由一条条构建镜像所需的指令和参数构成的脚本。

Dockerfile，需要定义一个Dockerfile，Dockerfile定义了进程需要的一切东西。Dockerfile涉及的内容包括执行代码或者是文件、环境变量、依赖包、运行时环境、动态链接库、操作系统的发行版、服务进程和内核进程(当应用进程需要和系统服务和内核进程打交道，这时需要考虑如何设计namespace的权限控制)等等；

构建过程

- 每条保留字指令都必须为大写字母且后面要跟随至少一个参数
- 指令按照从上到下，顺序执行
- #表示注释
- 每条指令都会创建一个新的镜像层并对镜像进行提交

1. docker从基础镜像运行一个容器
2. 执行一条指令并对容器作出修改
3. 执行类似docker commit的操作提交一个新的镜像层
4. docker再基于刚提交的镜像运行一个新容器
5. 执行dockerfile中的下一条指令直到所有指令都执行完成

保留字

FROM

基础镜像，当前新镜像是基于哪个镜像的，指定一个已经存在的镜像作为模板，第一条必须是from

MAINTAINER

maintainer，镜像维护者的姓名和邮箱地址

RUN

容器构建时需要运行的命令

一共有两种模式：

- shell模式：RUN 命令
- exec模式：RUN["可执行文件","参数1","参数2".....]

RUN在docker build时运行

EXPOSE

当前容器对外暴露出的端口

WORKDIR

指定在创建容器后，终端默认登陆的进来工作目录，一个落脚点

USER

指定该镜像以什么样的用户去执行，如果都不指定，默认是root

ENV

用来在构建镜像过程中设置环境变量

```
ENV 变量名 变量值
```

比如：

```
ENV JAVA_HOME /usr/test
```

ADD

将宿主机目录下的文件拷贝进镜像且会自动处理URL和解压tar压缩包

COPY

类似ADD，拷贝文件和目录到镜像中，但是不解压

```
COPY src dest
```

VOLUME

容器数据卷，用于数据保存和持久化工作

CMD

指定容器启动后的要干的事情

一共有两种模式：

- shell模式：RUN 命令

- exec模式: RUN["可执行文件","参数1","参数2".....]

Dockerfile 中可以有多条 CMD 指令, 但只有最后一条生效, CMD 会被 docker run 之后的参数替换
CMD是在docker run 时运行。

ENTRYPOINT

类似于 CMD 指令, 但是ENTRYPOINT不会被docker run后面的命令覆盖, 而且这些命令行参数会被当作参数送给 ENTRYPOINT 指令指定的程序

ENTRYPOINT可以和CMD一起用, 一般是变参才会使用 CMD, 这里的 CMD 等于是在给 ENTRYPOINT 传参。

当指定了ENTRYPOINT后, CMD的含义就发生了变化, 不再是直接运行其命令而是将CMD的内容作为参数传递给ENTRYPOINT指令, 他两个组合会变成 ENTRYPOINT CMD

如果 Dockerfile 中如果存在多个 ENTRYPOINT 指令, 仅最后一条生效。

示例

```
#
# NOTE: THIS DOCKERFILE IS GENERATED VIA "apply-templates.sh"
#
# PLEASE DO NOT EDIT IT DIRECTLY.
#

FROM amazoncorretto:17

ENV CATALINA_HOME /usr/local/tomcat
ENV PATH $CATALINA_HOME/bin:$PATH
RUN mkdir -p "$CATALINA_HOME"
WORKDIR $CATALINA_HOME

# let "Tomcat Native" live somewhere isolated
ENV TOMCAT_NATIVE_LIBDIR $CATALINA_HOME/native-jni-lib
ENV LD_LIBRARY_PATH ${LD_LIBRARY_PATH:+$LD_LIBRARY_PATH:}$TOMCAT_NATIVE_LIBDIR

# see https://www.apache.org/dist/tomcat/tomcat-10/KEYS
# see also "versions.sh" (https://github.com/docker-library/tomcat/blob/master/versions.sh)
ENV GPG_KEYS A9C5DF4D22E99998D9875A5110C01C5A2F6059E7

ENV TOMCAT_MAJOR 10
ENV TOMCAT_VERSION 10.0.22
ENV TOMCAT_SHA512
fe46db8794f066882b30e7a94bd8d3dbcf29e8e8ffaf67c1355846755745a7c9eafd124819283f21
8bcf410921a485b44b57b56fd6251fb99d67d95f3dd36826

RUN set -eux; \
\
# http://yum.baseurl.org/wiki/YumDB.html
if ! command -v yumdb > /dev/null; then \
```

```

yum install -y yum-utils; \
yumdb set reason dep yum-utils; \
fi; \
# a helper function to "yum install" things, but only if they aren't installed
(and to set their "reason" to "dep" so "yum autoremove" can purge them for us)
_yum_install_temporary() { ( set -eu +x; \
    local pkg todo=''; \
    for pkg; do \
        if ! rpm --query "$pkg" > /dev/null 2>&1; then \
            todo="$todo $pkg"; \
        fi; \
    done; \
    if [ -n "$todo" ]; then \
        set -x; \
        yum install -y $todo; \
        yumdb set reason dep $todo; \
    fi; \
) }; \
_yum_install_temporary gzip tar; \
\
ddist() { \
    local f="$1"; shift; \
    local distFile="$1"; shift; \
    local mvnFile="{1:-}"; \
    local success=; \
    local distUrl=; \
    for distUrl in \
# https://issues.apache.org/jira/browse/INFRA-8753?
focusedCommentId=14735394#comment-14735394
        "https://www.apache.org/dyn/closer.cgi?
action=download&filename=$distFile" \
# if the version is outdated (or we're grabbing the .asc file), we might have to
pull from the dist/archive :/
        "https://downloads.apache.org/$distFile" \
        "https://www-us.apache.org/dist/$distFile" \
        "https://www.apache.org/dist/$distFile" \
        "https://archive.apache.org/dist/$distFile" \
# if all else fails, let's try Maven (https://www.mail-
archive.com/users@tomcat.apache.org/msg134940.html;
https://mvnrepository.com/artifact/org.apache.tomcat/tomcat;
https://repo1.maven.org/maven2/org/apache/tomcat/tomcat/)
    ${mvnFile:+ "https://repo1.maven.org/maven2/org/apache/tomcat/tomcat/$mvnFile"} \
    ; do \
        if curl -fL -o "$f" "$distUrl" && [ -s "$f" ]; then \
            success=1; \
            break; \
        fi; \
    done; \
    [ -n "$success" ]; \
}; \
\
ddist 'tomcat.tar.gz'
"tomcat/tomcat-$TOMCAT_MAJOR/v$TOMCAT_VERSION/bin/apache-
tomcat-$TOMCAT_VERSION.tar.gz" "$TOMCAT_VERSION/tomcat-$TOMCAT_VERSION.tar.gz";
\
echo "$TOMCAT_SHA512 *tomcat.tar.gz" | sha512sum --strict --check -; \

```



```

ddist 'tomcat.tar.gz.asc'
"tomcat/tomcat-$TOMCAT_MAJOR/v$TOMCAT_VERSION/bin/apache-
tomcat-$TOMCAT_VERSION.tar.gz.asc"
"$TOMCAT_VERSION/tomcat-$TOMCAT_VERSION.tar.gz.asc"; \
    export GNUPGHOME="$(mktemp -d)"; \
    for key in $GPG_KEYS; do \
        gpg --batch --keyserver keyserver.ubuntu.com --recv-keys "$key"; \
    done; \
    gpg --batch --verify tomcat.tar.gz.asc tomcat.tar.gz; \
    tar -xf tomcat.tar.gz --strip-components=1; \
    rm bin/*.bat; \
    rm tomcat.tar.gz*; \
    command -v gpgconf && gpgconf --kill all || :; \
    rm -rf "$GNUPGHOME"; \
\
# https://tomcat.apache.org/tomcat-9.0-doc/security-
howto.html#Default_web_applications
mv webapps webapps.dist; \
mkdir webapps; \
# we don't delete them completely because they're frankly a pain to get back for
users who do want them, and they're generally tiny (~7MB)
\
nativeBuildDir="$(mktemp -d)"; \
tar -xf bin/tomcat-native.tar.gz -C "$nativeBuildDir" --strip-components=1;
\
_yum_install_temporary \
    apr-devel \
    gcc \
    make \
    openssl-devel \
; \
( \
    export CATALINA_HOME="$PWD"; \
    cd "$nativeBuildDir/native"; \
    aprConfig="$(command -v apr-1-config)"; \
    ./configure \
        --libdir="$TOMCAT_NATIVE_LIBDIR" \
        --prefix="$CATALINA_HOME" \
        --with-apr="$aprConfig" \
        --with-java-home="$JAVA_HOME" \
        --with-ssl=yes \
    ; \
    nproc="$(nproc)"; \
    make -j "$nproc"; \
    make install; \
); \
rm -rf "$nativeBuildDir"; \
rm bin/tomcat-native.tar.gz; \
\
# mark any explicit dependencies as manually installed
find "$TOMCAT_NATIVE_LIBDIR" -type f -executable -exec ldd '{}' ';' \
| awk '/=>/ && $(NF-1) != "=>" { print $(NF-1) }' \
| xargs -rt readlink -e \
| sort -u \
| xargs -rt rpm --query --whatprovides \
| sort -u \
| tee "$TOMCAT_NATIVE_LIBDIR/.dependencies.txt" \
| xargs -r yumdb set reason user \

```

```

; \
\
# clean up anything added temporarily and not later marked as necessary
yum autoremove -y; \
yum clean all; \
rm -rf /var/cache/yum; \
\
# sh removes env vars it doesn't support (ones with periods)
# https://github.com/docker-library/tomcat/issues/77
find ./bin/ -name '*.sh' -exec sed -ri 's|^#!/bin/sh$|#!/usr/bin/env bash|'
'{}' +; \
\
# fix permissions (especially for running as non-root)
# https://github.com/docker-library/tomcat/issues/35
chmod -R +rX .; \
chmod 777 logs temp work; \
\
# smoke test
catalina.sh version

# verify Tomcat Native is working properly
RUN set -eux; \
nativeLines="$(catalina.sh configtest 2>&1)"; \
nativeLines="$(echo "$nativeLines" | grep 'Apache Tomcat Native')"; \
nativeLines="$(echo "$nativeLines" | sort -u)"; \
if ! echo "$nativeLines" | grep -E 'INFO: Loaded( APR based)? Apache Tomcat
Native library' >&2; then \
    echo >&2 "$nativeLines"; \
    exit 1; \
fi

EXPOSE 8080
CMD ["catalina.sh", "run"]

```

自定义镜像

java17

说明：使用Ubuntu镜像，生成包含java17, vim、ifconfig的镜像

下载java17

下载页面

[Java Downloads | Oracle](#)

直接下载

https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz

放入指定目录

我放入的目录是C:\Users\mao\Desktop\test

新建Dockerfile文件

在刚才的目录下建立一个Dockerfile文件，要和jdk的存放目录一致

```
PS C:\Users\mao\Desktop\test> ls
```

目录: C:\Users\mao\Desktop\test

| Mode | LastWriteTime | Length | Name |
|--------|-----------------|-----------|-----------------------------|
| -a---- | 2022/6/17 21:47 | 0 | Dockerfile |
| -a---- | 2022/6/17 21:42 | 181203151 | jdk-17_linux-x64_bin.tar.gz |

```
PS C:\Users\mao\Desktop\test>
```

编写Dockerfile文件

使用记事本打开，输入以下命令

```
#基础镜像，当前新镜像是基于哪个镜像的，指定一个已经存在的镜像作为模板，第一条必须是from
FROM ubuntu
# 镜像维护者的姓名和邮箱地址
MAINTAINER mao<1296193245@qq.com>
#用来在构建镜像过程中设置环境变量
ENV MYPATH /usr/local
# 指定在创建容器后，终端默认登陆的进来工作目录
WORKDIR $MYPATH

#更新依赖
RUN apt-get update
#安装vim编辑器
RUN apt-get -y install vim
#安装ifconfig命令查看网络IP
RUN apt-get install net-tools
#安装java17及lib库
# RUN apt-get -y install glibc
# 创建一个文件夹
RUN mkdir /usr/local/java
#ADD 是相对路径jar,把jdk-17_linux-x64_bin.tar.gz添加到容器中,安装包必须要和Dockerfile文件在同一位置
ADD jdk-17_linux-x64_bin.tar.gz /usr/local/java/
#配置java环境变量
```

```

ENV JAVA_HOME /usr/local/java/jdk-17.0.3.1
ENV JRE_HOME $JAVA_HOME/jre
ENV CLASSPATH
$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib:$CLASSPATH
ENV PATH $JAVA_HOME/bin:$PATH

#暴露端口
EXPOSE 80

#打印
CMD echo $MYPATH
CMD echo "success-----ok"
# 如果不传入参数，默认运行/bin/bash
CMD /bin/bash

```

构建

构建命令：

```
docker build -t 新镜像名字:TAG .
```

运行命令：

```
docker build -t java17:1.0 .
```

```

PS C:\Users\mao\Desktop\test> docker build -t java17:1.0 .
[+] Building 5.1s (12/12) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 1.23kB                                           0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest                0.0s
=> [1/7] FROM docker.io/library/ubuntu                                         0.0s
=> [internal] load build context                                                0.0s
=> => transferring context: 51B                                                0.0s
=> CACHED [2/7] WORKDIR /usr/local                                             0.0s
=> CACHED [3/7] RUN apt-get update                                             0.0s

```

```

=> CACHED [4/7] RUN apt-get -y install vim                                0.0s
=> CACHED [5/7] RUN apt-get install net-tools                            0.0s
=> [6/7] RUN mkdir /usr/local/java                                       0.4s
=> [7/7] ADD jdk-17_linux-x64_bin.tar.gz /usr/local/java/                3.5s
=> exporting to image                                                    1.2s
=> => exporting layers                                                    1.2s
=> => writing image
sha256:282982c69086e51636a387c7156a1e246a6f7109c36a663c09e0302ec2c95a03 0.0s
=> => naming to docker.io/library/java17:1.0                             0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and
learn how to fix them
PS C:\Users\mao\Desktop\test>

```

查看镜像

```
docker images
```

```

PS C:\Users\mao\Desktop\test> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
java17        1.0       282982c69086  12 seconds ago  489MB
tomcat        latest    c795915cb678  2 weeks ago   680MB
redis         latest    53aa81e8adfa  2 weeks ago   117MB
mysql         latest    65b636d5542b  2 weeks ago   524MB
ubuntu        latest    d2e4e1f51132  6 weeks ago   77.8MB
PS C:\Users\mao\Desktop\test>

```

运行容器

```
docker run -it --name java17 java17:1.0 /bin/bash
```

```

PS C:\Users\mao\Desktop\test> docker run -it --name java17 java17:1.0 /bin/bash
root@20c9959d6873:/usr/local# pwd
/usr/local
root@20c9959d6873:/usr/local# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 172.17.0.2 netmask 255.255.0.0 broadcast 172.17.255.255

```

```

ether 02:42:ac:11:00:02 txqueuelen 0 (Ethernet)
RX packets 10 bytes 876 (876.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@20c9959d6873:/usr/local# vim
root@20c9959d6873:/usr/local# ls
bin etc games include java lib man sbin share src test.txt
root@20c9959d6873:/usr/local# java -version
java version "17.0.3.1" 2022-04-22 LTS
Java(TM) SE Runtime Environment (build 17.0.3.1+2-LTS-6)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.3.1+2-LTS-6, mixed mode, sharing)
root@20c9959d6873:/usr/local#

```

Docker运行后端服务

创建springboot项目

项目名称为Docker_boot

maven

maven的pom.xml文件:

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.0</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>mao</groupId>
  <artifactId>Docker_boot</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Docker_boot</name>
  <description>Docker_boot</description>
  <properties>

```

```

        <java.version>11</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>

```

配置

application.yml文件:

```

#多环境开发---单文件
spring:
  profiles:
    # 当前环境
    active: dev

#-----通用环境-----

server:
  #配置端口号
  port: 8082

---

#-----开发环境-----

spring:
  config:
    activate:

```

```

on-profile: dev

# 开启debug模式，输出调试信息，常用于检查系统运行状况
#debug: true

# 设置日志级别，root表示根节点，即整体应用日志级别
logging:
# 日志输出到文件的文件名
file:
    name: server.log
# 字符集
charset:
    file: UTF-8
# 分文件
logback:
    rollingpolicy:
        #最大文件大小
        max-file-size: 16KB
        # 文件格式
        file-name-pattern: server_log-%d{yyyy/MM月/dd日/}%i.log
# 设置日志组
group:
# 自定义组名，设置当前组中所包含的包
mao_pro: mao
level:
    root: info
    # 为对应组设置日志级别
    mao_pro: debug
    # 日志输出格式
# pattern:
# console: "%d %clr(%p) --- [%16t] %clr(%-40.40c){cyan} : %m %n"

---
#----生产环境-----

spring:
config:
    activate:
        on-profile: pro

# 开启debug模式，输出调试信息，常用于检查系统运行状况
#debug: true

# 设置日志级别，root表示根节点，即整体应用日志级别
logging:
# 日志输出到文件的文件名

```



```

file:
  name: server.log
# 字符集
charset:
  file: UTF-8
# 分文件
logback:
  rollingpolicy:
    #最大文件大小
    max-file-size: 4MB
    # 文件格式
    file-name-pattern: server_log-%d{yyyy/MM月/dd日/}%i.log
# 设置日志组
group:
  # 自定义组名，设置当前组中所包含的包
  mao_pro: mao
level:
  root: info
  # 为对应组设置日志级别
  mao_pro: warn
  # 日志输出格式
# pattern:
# console: "%d %clr(%p) --- [%16t] %clr(%-40.40c){cyan} : %m %n"

---
#----测试环境-----

spring:
  config:
    activate:
      on-profile: test

```

controller

MyController:

```

package mao.docker_boot.controller;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

import java.util.UUID;

/**

```

```

* Project name(项目名称): Docker_boot
* Package(包名): mao.docker_boot.controller
* Class(类名): MyController
* Author(作者): mao
* Author QQ: 1296193245
* GitHub: https://github.com/maomao124/
* Date(创建日期): 2022/6/17
* Time(创建时间): 23:01
* Version(版本): 1.0
* Description(描述): 业务类
*/

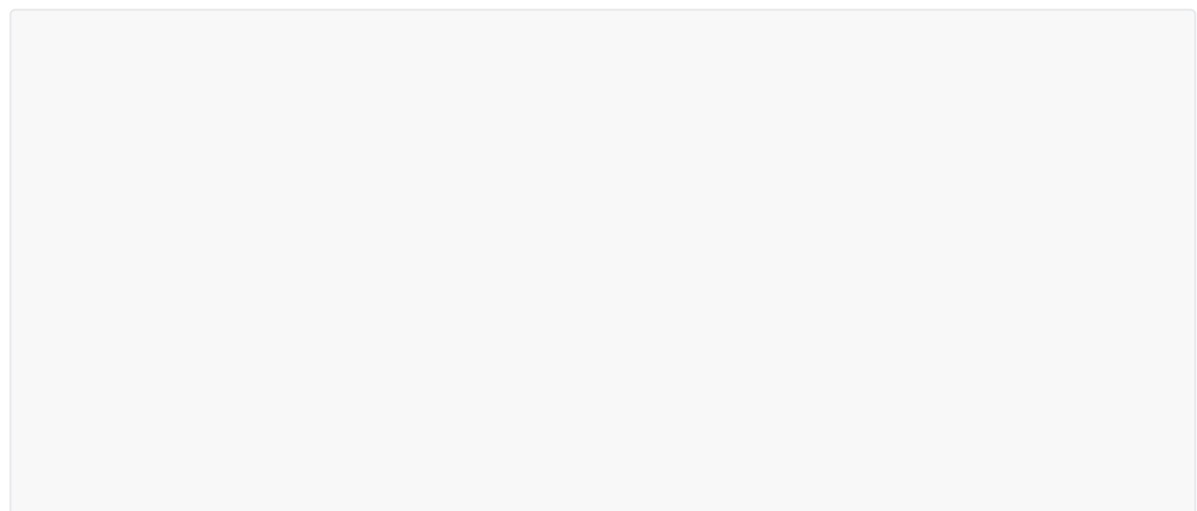
@RestController
public class MyController
{
    /**
     * 端口号，从配置文件里读
     */
    @Value("${server.port}")
    private String port;

    @RequestMapping("/docker")
    public String helloDocker()
    {
        return "hello docker and spring boot"+"<br>端口号: "+port+"<br>"+
        UUID.randomUUID();
    }

    @RequestMapping(value = "/index",method = RequestMethod.GET)
    public String index()
    {
        return "服务端口号: "+" "+port+"<br>"+UUID.randomUUID();
    }
}

```

启动测试



```
C:\Users\mao\.jdk\openjdk-16.0.2\bin\java.exe -XX:TieredStopAtLevel=1 -noverify
-Dspring.output.ansi.enabled=always "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2021.2.2\lib\idea_rt.jar=54055:C:\Program
Files\JetBrains\IntelliJ IDEA 2021.2.2\bin" -Dcom.sun.management.jmxremote -
Dspring.jmx.enabled=true -Dspring.liveBeansView.mbeanDomain -
Dspring.application.admin.enabled=true -Dfile.encoding=UTF-8 -classpath H:\程序\大
三下期
```

```
\Docker_boot\target\classes;C:\Users\mao\.m2\repository\org\springframework\boot
\spring-boot-starter-web\2.7.0\spring-boot-starter-web-
2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter\2.7.0\spring-boot-starter-
2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-
boot\2.7.0\spring-boot-
2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
autoconfigure\2.7.0\spring-boot-autoconfigure-
2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter-logging\2.7.0\spring-boot-starter-logging-
2.7.0.jar;C:\Users\mao\.m2\repository\ch\qos\logback\logback-
classic\1.2.11\logback-classic-
1.2.11.jar;C:\Users\mao\.m2\repository\ch\qos\logback\logback-
core\1.2.11\logback-core-
1.2.11.jar;C:\Users\mao\.m2\repository\org\apache\logging\log4j\log4j-to-
slf4j\2.17.2\log4j-to-slf4j-
2.17.2.jar;C:\Users\mao\.m2\repository\org\apache\logging\log4j\log4j-
api\2.17.2\log4j-api-2.17.2.jar;C:\Users\mao\.m2\repository\org\slf4j\jul-to-
slf4j\1.7.36\jul-to-slf4j-
1.7.36.jar;C:\Users\mao\.m2\repository\jakarta\annotation\jakarta.annotation-
api\1.3.5\jakarta.annotation-api-
1.3.5.jar;C:\Users\mao\.m2\repository\org\yaml\snakeyaml\1.30\snakeyaml-
1.30.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter-json\2.7.0\spring-boot-starter-json-
2.7.0.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\core\jackson-
databind\2.13.3\jackson-databind-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\core\jackson-
annotations\2.13.3\jackson-annotations-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\core\jackson-
core\2.13.3\jackson-core-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\datatype\jackson-
datatype-jdk8\2.13.3\jackson-datatype-jdk8-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\datatype\jackson-
datatype-jsr310\2.13.3\jackson-datatype-jsr310-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\module\jackson-
module-parameter-names\2.13.3\jackson-module-parameter-names-
2.13.3.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter-tomcat\2.7.0\spring-boot-starter-tomcat-
2.7.0.jar;C:\Users\mao\.m2\repository\org\apache\tomcat\embed\tomcat-embed-
core\9.0.63\tomcat-embed-core-
9.0.63.jar;C:\Users\mao\.m2\repository\org\apache\tomcat\embed\tomcat-embed-
el\9.0.63\tomcat-embed-el-
9.0.63.jar;C:\Users\mao\.m2\repository\org\apache\tomcat\embed\tomcat-embed-
websocket\9.0.63\tomcat-embed-websocket-
9.0.63.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
web\5.3.20\spring-web-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
beans\5.3.20\spring-beans-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
webmvc\5.3.20\spring-webmvc-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
```

```
aop\5.3.20\spring-aop-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
context\5.3.20\spring-context-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
expression\5.3.20\spring-expression-
5.3.20.jar;C:\Users\mao\.m2\repository\org\slf4j\slf4j-api\1.7.36\slf4j-api-
1.7.36.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
core\5.3.20\spring-core-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
jcl\5.3.20\spring-jcl-5.3.20.jar mao.docker_boot.DockerBootApplication
OpenJDK 64-Bit Server VM warning: Options -Xverify:none and -noverify were
deprecated in JDK 13 and will likely be removed in a future release.
```

```

      .  _ _ _ _ _
     /\ /  _ _ ' _ _ _ _ _ ( ) _ _ _ _ _ \ \ \ \
    ( ( ) \ _ _ | ' _ | ' _ | ' _ \ / _ ' | \ \ \ \
   \ \ / _ _ | | | | | | | | | | | | | | | | | | | |
    ' | _ _ | . _ | | | | | | | | | | | | | | | | | |
   =====|_|=====|_|_/=/_/=/_/=/
:: Spring Boot ::                (v2.7.0)

```

```

2022-06-17 23:07:20.341 INFO 19932 --- [main]
mao.docker_boot.DockerBootApplication : Starting DockerBootApplication using
Java 16.0.2 on mao with PID 19932 (H:\程序\大三下期\docker_boot\target\classes
started by mao in H:\程序\大三下期\docker_boot)
2022-06-17 23:07:20.343 DEBUG 19932 --- [main]
mao.docker_boot.DockerBootApplication : Running with Spring Boot v2.7.0,
Spring v5.3.20
2022-06-17 23:07:20.344 INFO 19932 --- [main]
mao.docker_boot.DockerBootApplication : The following 1 profile is active:
"dev"
2022-06-17 23:07:20.922 INFO 19932 --- [main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8082
(http)
2022-06-17 23:07:20.929 INFO 19932 --- [main]
o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-06-17 23:07:20.929 INFO 19932 --- [main]
org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
Tomcat/9.0.63]
2022-06-17 23:07:20.998 INFO 19932 --- [main] o.a.c.c.C.[Tomcat].
[localhost].[/] : Initializing Spring embedded webApplicationContext
2022-06-17 23:07:20.999 INFO 19932 --- [main]
w.s.c.ServletWebServerApplicationContext : Root webApplicationContext:
initialization completed in 618 ms
2022-06-17 23:07:21.227 INFO 19932 --- [main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8082
(http) with context path ''
2022-06-17 23:07:21.236 INFO 19932 --- [main]
mao.docker_boot.DockerBootApplication : Started DockerBootApplication in
1.213 seconds (JVM running for 1.648)

```

访问

localhost:8082/index

无问题

生成jar包

```
mvn package
```

或者

```
mvn package -DskipTests
```

```
PS H:\程序\大三下期\Docke_boot> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< mao:Docke_boot >-----
[INFO] Building Docke_boot 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @
Docke_boot ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:compile (default-compile) @ Docke_boot
---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 2 source files to H:\程序\大三下期\Docke_boot\target\classes
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:testResources (default-testResources) @
Docke_boot ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] skip non existing resourceDirectory H:\程序\大三下期
\Docke_boot\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.10.1:testCompile (default-testCompile) @
Docke_boot ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to H:\程序\大三下期\Docke_boot\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ Docke_boot ---
[INFO]
[INFO] -----
[INFO] T E S T S
```

```
[INFO] -----
[INFO] Running mao.docker_boot.DockerBootApplicationTests
23:10:09.954 [main] DEBUG org.springframework.test.context.BootstrapUtils -
Instantiating CacheAwareContextLoaderDelegate from class
[org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate]
23:10:09.962 [main] DEBUG org.springframework.test.context.BootstrapUtils -
Instantiating BootstrapContext using constructor [public
org.springframework.test.context.support.DefaultBootstrapContext(java.lang.Class
,org.springframework.test.context.CacheAwareContextLoaderDelegate)]
23:10:09.992 [main] DEBUG org.springframework.test.context.BootstrapUtils -
Instantiating TestContextBootstrapper for test class
[mao.docker_boot.DockerBootApplicationTests] from class
[org.springframework.boot.test.context.SpringBootTestContextBootstrapper]
23:10:10.003 [main] INFO
org.springframework.boot.test.context.SpringBootTestContextBootstrapper -
Neither @ContextConfiguration nor @ContextHierarchy found for test class
[mao.docker_boot.DockerBootApplicationTests], using SpringBootTestContextLoader
23:10:10.006 [main] DEBUG
org.springframework.test.context.support.AbstractContextLoader - Did not detect
default resource location for test class
[mao.docker_boot.DockerBootApplicationTests]: class path resource
[mao/docker_boot/DockerBootApplicationTests-context.xml] does not exist
23:10:10.007 [main] DEBUG
org.springframework.test.context.support.AbstractContextLoader - Did not detect
default resource location for test class
[mao.docker_boot.DockerBootApplicationTests]: class path resource
[mao/docker_boot/DockerBootApplicationTestsContext.groovy] does not exist
23:10:10.007 [main] INFO
org.springframework.test.context.support.AbstractContextLoader - Could not
detect default resource locations for test class
[mao.docker_boot.DockerBootApplicationTests]: no resource found for suffixes {-
context.xml, Context.groovy}.
23:10:10.007 [main] INFO
org.springframework.test.context.support.AnnotationConfigContextLoaderUtils -
Could not detect default configuration classes for test class
[mao.docker_boot.DockerBootApplicationTests]: DockerBootApplicationTests does not
declare any static, non-private, non-final, nested classes annotated with
@Configuration.
23:10:10.045 [main] DEBUG
org.springframework.test.context.support.ActiveProfilesUtils - Could not find an
'annotation declaring class' for annotation type
[org.springframework.test.context.ActiveProfiles] and class
[mao.docker_boot.DockerBootApplicationTests]
23:10:10.092 [main] DEBUG
org.springframework.context.annotation.ClassPathScanningCandidateComponentProvid
er - Identified candidate component class: file [H:\程序\大三下期
\docker_boot\target\classes\mao\docker_boot\DockerBootApplication.class]
23:10:10.093 [main] INFO
org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found
@SpringBootConfiguration mao.docker_boot.DockerBootApplication for test class
mao.docker_boot.DockerBootApplicationTests
23:10:10.173 [main] DEBUG
org.springframework.boot.test.context.SpringBootTestContextBootstrapper -
@TestExecutionListeners is not present for class
[mao.docker_boot.DockerBootApplicationTests]: using defaults.
```

```
23:10:10.174 [main] INFO
org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Loaded
default TestExecutionListener class names from location [META-
INF/spring.factories]:
[org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener,
org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener,
org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListen
er,
org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerRese
tTestExecutionListener,
org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailur
eTestExecutionListener,
org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionLi
stener,
org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceSer
verTestExecutionListener,
org.springframework.test.context.web.ServletTestExecutionListener,
org.springframework.test.context.support DirtiesContextBeforeModesTestExecutionL
istener,
org.springframework.test.context.event.ApplicationEventsTestExecutionListener,
org.springframework.test.context.support.DependencyInjectionTestExecutionListene
r, org.springframework.test.context.support DirtiesContextTestExecutionListener,
org.springframework.test.context.transaction.TransactionalTestExecutionListener,
org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener,
org.springframework.test.context.event.EventPublishingTestExecutionListener]
23:10:10.185 [main] DEBUG
org.springframework.boot.test.context.SpringBootTestContextBootstrapper -
Skipping candidate TestExecutionListener
[org.springframework.test.context.transaction.TransactionalTestExecutionListener]
due to a missing dependency. Specify custom listener classes or make the default
listener classes and their required dependencies available. Offending class:
[org.springframework.transaction.interceptor.TransactionAttributesSource]
23:10:10.185 [main] DEBUG
org.springframework.boot.test.context.SpringBootTestContextBootstrapper -
Skipping candidate TestExecutionListener
[org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener] due to a
missing dependency. Specify custom listener classes or make the default listener
classes and their required dependencies available. Offending class:
[org.springframework.transaction.interceptor.TransactionAttribute]
```

```
23:10:10.185 [main] INFO
org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Using
TestExecutionListeners:
[org.springframework.test.context.web.ServletTestExecutionListener@20f5281c,
org.springframework.test.context.support DirtiesContextBeforeModesTestExecutionL
istener@56c4278e,
org.springframework.test.context.event.ApplicationEventsTestExecutionListener@30
1eda63,
org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener@3d246ea3,
org.springframework.boot.test.autoconfigure.SpringBootDependencyInjectionTestExe
cutionListener@341814d3,
org.springframework.test.context.support DirtiesContextTestExecutionListener@439
7ad89,
org.springframework.test.context.event.EventPublishingTestExecutionListener@59cb
a5a,
org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener@1bd39
d3c,
org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListen
er@6f19ac19,
org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerRese
tTestExecutionListener@119cbf96,
org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailur
eTestExecutionListener@71329995,
org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionLi
stener@768fc0f2,
org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceSer
verTestExecutionListener@5454d35e]
```



```
23:10:10.189 [main] DEBUG
org.springframework.test.context.support.AbstractDirtyContextTestExecutionList
ener - Before test class: context [DefaultTestContext@7d9e8ef7 testClass =
DockerBootApplicationTests, testInstance = [null], testMethod = [null],
testException = [null], mergedContextConfiguration =
[WebMergedContextConfiguration@f107c50 testClass = DockerBootApplicationTests,
locations = '{}', classes = '{class mao.docker_boot.DockerBootApplication}',
contextInitializerClasses = '[]', activeProfiles = '{}', propertySourceLocations
= '{}', propertySourceProperties =
'{org.springframework.boot.test.context.SpringBootTestContextBootstrapper=true}'
, contextCustomizers =
set[org.springframework.boot.test.context.filter.ExcludeFilterContextCustomizer@
4944252c,
org.springframework.boot.test.json.DuplicateJsonObjectContextCustomizerFactory$D
uplicateJsonObjectContextCustomizer@272ed83b,
org.springframework.boot.test.mock.mockito.MockitoContextCustomizer@0,
org.springframework.boot.test.web.client.TestRestTemplateContextCustomizer@15bbf
42f,
org.springframework.boot.test.autoconfigure.actuate.metrics.MetricsExportContext
CustomizerFactory$DisableMetricExportContextCustomizer@20f5239f,
org.springframework.boot.test.autoconfigure.properties.PropertyMappingContextCus
tomizer@0,
org.springframework.boot.test.autoconfigure.web.servlet.WebDriverContextCustomiz
erFactory$Customizer@3e84448c,
org.springframework.boot.test.context.SpringBootTestArgs@1,
org.springframework.boot.test.context.SpringBootTestWebEnvironment@4b1c1ea0],
resourceBasePath = 'src/main/webapp', contextLoader =
'org.springframework.boot.test.context.SpringBootTestContextLoader', parent =
[null]], attributes =
map['org.springframework.test.context.web.ServletTestExecutionListener.activateL
istener' -> true]], class annotated with @DirtyContext [false] with mode
[null].
```

```
2022-06-17 23:10:10.592 INFO 296 --- [main]
m.d.DockerBootApplicationTests : Starting DockerBootApplicationTests
using Java 16.0.2 on mao with PID 296 (started by mao in H:\程序\大三下期
\docker_boot)
2022-06-17 23:10:10.593 DEBUG 296 --- [main]
m.d.DockerBootApplicationTests : Running with Spring Boot v2.7.0,
Spring v5.3.20
2022-06-17 23:10:10.594 INFO 296 --- [main]
m.d.DockerBootApplicationTests : The following 1 profile is active:
"dev"
2022-06-17 23:10:11.923 INFO 296 --- [main]
m.d.DockerBootApplicationTests : Started DockerBootApplicationTests in
1.696 seconds (JVM running for 2.504)
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.53 s -
in mao.docker_boot.DockerBootApplicationTests
[INFO]
[INFO] Results:
```

```
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.2.2:jar (default-jar) @ Docker_boot ---
[INFO] Building jar: H:\程序\大三下期\docker_boot\target\docker_boot-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.0:repackage (repackage) @ Docker_boot ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 12.120 s
[INFO] Finished at: 2022-06-17T23:10:15+08:00
[INFO] -----
PS H:\程序\大三下期\docker_boot>
```

项目地址:

https://github.com/maomao124/Docker_boot.git/

将jar包放入指定目录

我放入的目录是C:\Users\mao\Desktop\test

```
PS C:\Users\mao\Desktop\test> ls

    目录: C:\Users\mao\Desktop\test

Mode                LastWriteTime         Length Name
----                -
-a-----         2022/6/17      23:10      17610406 Docker_boot-0.0.1-SNAPSHOT.jar

PS C:\Users\mao\Desktop\test>
```

新建Dockerfile文件

在刚才的目录下建立一个Dockerfile文件，要和jdk的存放目录一致

```
PS C:\Users\mao\Desktop\test> ls
```

目录: C:\Users\mao\Desktop\test

| Mode | LastWriteTime | Length | Name |
|--------|-----------------|----------|--------------------------------|
| -a---- | 2022/6/17 23:17 | 0 | Dockerfile |
| -a---- | 2022/6/17 23:10 | 17610406 | Docker_boot-0.0.1-SNAPSHOT.jar |

```
PS C:\Users\mao\Desktop\test>
```

编写Dockerfile文件

使用记事本打开，输入以下命令

```
# 基础镜像使用java
FROM java17:1.0
# 作者
MAINTAINER mao
# VOLUME 指定临时文件目录为/tmp，在主机/var/lib/docker目录下创建了一个临时文件并链接到容器的/tmp
VOLUME /tmp
# 将jar包添加到容器中并更名为Docker_boot.jar
ADD Docker_boot-0.0.1-SNAPSHOT.jar Docker_boot.jar
# 运行jar包
RUN bash -c 'touch Docker_boot.jar'
ENTRYPOINT ["java","-jar","Docker_boot.jar"]
#暴露端口作为微服务
EXPOSE 8082
```

构建

运行命令：

```
docker build -t docker_boot:1.0 .
```

```
PS C:\Users\mao\Desktop\test> docker build -t docker_boot:1.0 .
[+] Building 1.0s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
```

0.0s

```

=> => transferring dockerfile: 501B
0.0s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [internal] load metadata for docker.io/library/java17:1.0
0.0s
=> [internal] load build context
0.2s
=> => transferring context: 17.61MB
0.1s
=> [1/3] FROM docker.io/library/java17:1.0
0.2s
=> [2/3] ADD Docker_boot-0.0.1-SNAPSHOT.jar Docker_boot.jar
0.1s
=> [3/3] RUN bash -c 'touch Docker_boot.jar'
0.5s
=> exporting to image
0.1s
=> => exporting layers
0.1s
=> => writing image
sha256:d25f3f4955aec0b9ad19ab1e5290a1cc4a50eb283917df4ef8f520df6f37a08f
0.0s
=> => naming to docker.io/library/docker_boot:1.0
0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and
learn how to fix them
PS C:\Users\mao\Desktop\test>

```

查看镜像

docker images

```

PS C:\Users\mao\Desktop\test> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
docker_boot   1.0       d25f3f4955ae   55 seconds ago 525MB
java17        1.0       282982c69086   14 hours ago  489MB
tomcat        latest    c795915cb678   2 weeks ago   680MB
redis         latest    53aa81e8adfa   2 weeks ago   117MB
mysql         latest    65b636d5542b   2 weeks ago   524MB
ubuntu        latest    d2e4e1f51132   7 weeks ago   77.8MB
PS C:\Users\mao\Desktop\test>

```

运行容器

```
docker run -d -p8082:8082 --name docker_boot1 docker_boot:1.0
```

```
PS C:\Users\mao\Desktop\test> docker run -d -p8082:8082 --name docker_boot1
docker_boot:1.0
251d1d54e1e699d3afdc6b1ab08f2cd4ada9ddd9bfcd1bde0c4b72b2d563e31
PS C:\Users\mao\Desktop\test> docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS |
|--------------|-----------------|--------------------------|---------------|--------------|
| 251d1d54e1e6 | docker_boot:1.0 | "java -jar Docker_bo..." | 4 seconds ago | Up 3 seconds |
| 20c9959d6873 | java17:1.0 | "/bin/bash" | 14 hours ago | Exited |
| 9cf1e361139c | java17:1.0 | "--name java17 /bin/..." | 14 hours ago | Exited |
| 8a8076944128 | redis | "docker-entrypoint.s..." | 3 days ago | Exited |
| 81902b3cfdc4 | mysql | "docker-entrypoint.s..." | 3 days ago | Exited |
| 24ee3e986397 | mysql | "docker-entrypoint.s..." | 3 days ago | Exited |
| 2d379d342bb6 | mysql | "docker-entrypoint.s..." | 4 days ago | Exited |
| 3ca156e4541d | tomcat | "catalina.sh run" | 4 days ago | Exited |

访问测试

localhost:8082/index

localhost:8082/docker

```
PS C:\Users\mao\Desktop\test> docker exec -it 251d1d54e1e6 /bin/bash
root@251d1d54e1e6:/usr/local# ls
Docker_boot.jar bin etc games include java lib man sbin server.log
share src
root@251d1d54e1e6:/usr/local# vi server.log
```

```
2023-06-18 04:53:55.630 INFO 1 --- [main] mao.docker_boot.DockerBootApplication
: Starting DockerBootApplication v0.0.1-SNAPSHOT using Java 17.0.3.1 on
251d1d54e1e6 with PID 1 (/usr/local/Docker_boot.jar started by root in
/usr/local)
2023-06-18 04:53:55.632 DEBUG 1 --- [main] mao.docker_boot.DockerBootApplication
: Running with Spring Boot v2.7.0, Spring v5.3.20
2023-06-18 04:53:55.633 INFO 1 --- [main] mao.docker_boot.DockerBootApplication
: The following 1 profile is active: "dev"
2023-06-18 04:53:56.548 INFO 1 --- [main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8082
(http)
2023-06-18 04:53:56.557 INFO 1 --- [main]
o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-18 04:53:56.557 INFO 1 --- [main]
org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
Tomcat/9.0.63]
2023-06-18 04:53:56.619 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/]
: Initializing Spring embedded WebApplicationContext
2023-06-18 04:53:56.619 INFO 1 --- [main]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
initialization completed in 931 ms
2023-06-18 04:53:56.968 INFO 1 --- [main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8082
(http) with context path ''
2023-06-18 04:53:56.981 INFO 1 --- [main] mao.docker_boot.DockerBootApplication
: Started DockerBootApplication in 1.888 seconds (JVM running for 2.266)
2023-06-18 04:55:23.282 INFO 1 --- [http-nio-8082-exec-1] o.a.c.c.C.[Tomcat].
[localhost].[/] : Initializing Spring DispatcherServlet
'dispatcherServlet'
2023-06-18 04:55:23.282 INFO 1 --- [http-nio-8082-exec-1]
o.s.web.servlet.DispatcherServlet : Initializing Servlet
'dispatcherServlet'
2023-06-18 04:55:23.283 INFO 1 --- [http-nio-8082-exec-1]
o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
```

Docker网络

命令

所有命令:

```
PS C:\Users\mao\Desktop\test> docker network --help
```

```
Usage:  docker network COMMAND
```

```
Manage networks
```

Commands:

| | |
|------------|--|
| connect | Connect a container to a network |
| create | Create a network |
| disconnect | Disconnect a container from a network |
| inspect | Display detailed information on one or more networks |
| ls | List networks |
| prune | Remove all unused networks |
| rm | Remove one or more networks |

Run '`docker network COMMAND --help`' for more information on a command.

PS C:\Users\mao\Desktop\test>

查看网络

```
docker network ls
```

```
PS C:\Users\mao\Desktop\test> docker network ls
NETWORK ID        NAME          DRIVER  SCOPE
e26ce6e77409     bridge       bridge   local
58028068163e     host         host     local
c0e15a3d2248     none         null     local
PS C:\Users\mao\Desktop\test>
```

查看网络源数据

```
docker network inspect 网络名字
```

```
PS C:\Users\mao\Desktop\test> docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "e26ce6e774096478217bfed7fab0eecd79286ac7d640037e3c3874440cd4a5",
    "Created": "2022-06-18T04:22:16.059751393Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    }
  }
],
```

```
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {
      "com.docker.network.bridge.default_bridge": "true",
      "com.docker.network.bridge.enable_icc": "true",
      "com.docker.network.bridge.enable_ip_masquerade": "true",
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  }
]
PS C:\Users\mao\Desktop\test>
```

新建网络

```
docker network create 网络名字
```

```
PS C:\Users\mao\Desktop\test> docker network create abc
619d44f4afea6b7aca982a0ecd36bf69d097d66fdf5f41004c1d16dbeb8839c
```

```
PS C:\Users\mao\Desktop\test> docker network ls
NETWORK ID        NAME        DRIVER        SCOPE
619d44f4afea      abc         bridge        local
e26ce6e77409      bridge     bridge        local
58028068163e      host       host          local
c0e15a3d2248      none       null          local
PS C:\Users\mao\Desktop\test>
```

删除网络

```
docker network rm 网络名字
```



```
PS C:\Users\mao\Desktop\test> docker network rm abc
abc
PS C:\Users\mao\Desktop\test> docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
e26ce6e77409        bridge             bridge             local
58028068163e        host               host               local
c0e15a3d2248        none               null               local
PS C:\Users\mao\Desktop\test>
```

prune

删除所有未使用的网络

```
docker network prune
```

网络模式

bridge

Docker 服务默认会创建一个 docker0 网桥，其上有一个 docker0 内部接口，该桥接网络的名称为 docker0，它在内核层连通了其他的物理或虚拟网卡，这就将所有容器和本地主机都放到同一个物理网络。Docker 默认指定了 docker0 接口的 IP 地址和子网掩码，让主机和容器之间可以通过网桥相互通信。

1. Docker使用Linux桥接，在宿主机虚拟一个Docker容器网桥(docker0)，Docker启动一个容器时会根据Docker网桥的网段分配给容器一个IP地址，称为Container-IP，同时Docker网桥是每个容器的默认网关。因为在同一宿主机内的容器都接入同一个网桥，这样容器之间就能够通过容器的Container-IP直接通信。
2. docker run 的时候，没有指定network的话默认使用的网桥模式就是bridge，使用的就是docker0。在宿主机ifconfig,就可以看到docker0和自己create的网络(后面讲)eth0，eth1，eth2.....代表网卡一，网卡二，网卡三.....，lo代表127.0.0.1，即localhost，inet addr用来表示网卡的IP地址
3. 网桥docker0创建一对对等虚拟设备接口一个叫veth，另一个叫eth0，成对匹配。
 - 整个宿主机的网桥模式都是docker0，类似一个交换机有一堆接口，每个接口叫veth，在本地主机和容器内分别创建一个虚拟接口，并让他们彼此联通。这样一对接口叫veth pair
 - 每个容器实例内部也有一块网卡，每个接口叫eth0
 - docker0上面的每个veth匹配某个容器实例内部的eth0，两两配对，——匹配

host

host, 直接使用宿主机的 IP 地址与外界进行通信, 不再需要额外进行NAT 转换。

```
PS C:\Users\mao\Desktop> docker network inspect host
[
  {
    "Name": "host",
    "Id": "58028068163e97bb12cd7f24921cccf49ca06286f756e82b34d7a73494a562b",
    "Created": "2022-05-02T11:06:10.406162704Z",
    "Scope": "local",
    "Driver": "host",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": []
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
PS C:\Users\mao\Desktop>
```

none

在none模式下, 并不为Docker容器进行任何网络配置。

也就是说, 这个Docker容器没有网卡、IP、路由等信息, 只有一个lo

需要我们自己为Docker容器添加网卡、配置IP等。

```
PS C:\Users\mao\Desktop> docker network inspect none
[
  {
    "Name": "none",
    "Id": "c0e15a3d224806a9b9df4a1558048e744ccd81dfb2b82d93ef82a0426997ae69",
    "Created": "2022-05-02T11:06:10.394932679Z",
    "Scope": "local",
    "Driver": "null",
  }
]
```

```
    "EnableIPv6": false,
    "IPAM": {
        "Driver": "default",
        "Options": null,
        "Config": []
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
        "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
}
]
```

PS C:\Users\mao\Desktop>

container

新建的容器和已经存在的一个容器共享一个网络ip配置而不是和宿主机共享。新创建的容器不会创建自己的网卡，配置自己的IP，而是和一个指定的容器共享IP、端口范围等。同样，两个容器除了网络方面，其他的如文件系统、进程列表等还是隔离的。

使用：

```
docker run -it --network container:被共享的容器名 ...
```

自定义网络

创建网络：

```
docker network create 网络名字
```

自定义网络本身就维护好了主机名和ip的对应关系（ip和域名都能通）

平台架构

Docker 是一个 C/S 模式的架构，后端是一个松耦合架构，众多模块各司其职。

Docker 运行的基本流程：

1. 用户是使用 Docker Client 与 Docker Daemon 建立通信，并发送请求给后者。
2. Docker Daemon 作为 Docker 架构中的主体部分，首先提供 Docker Server 的功能使其可以接受 Docker Client 的请求。
3. Docker Engine 执行 Docker 内部的一系列工作，每一项工作都是以一个 Job 的形式存在。
4. Job 的运行过程中，当需要容器镜像时，则从 Docker Registry 中下载镜像，并通过镜像管理驱动 Graph driver 将下载镜像以 Graph 的形式存储。
5. 当需要为 Docker 创建网络环境时，通过网络管理驱动 Network driver 创建并配置 Docker 容器网络环境。
6. 当需要限制 Docker 容器运行资源或执行用户指令等操作时，则通过 Execdriver 来完成。
7. Libcontainer 是一项独立的容器管理包，Network driver 以及 Exec driver 都是通过 Libcontainer 来实现具体对容器进行的操作。

Docker-compose 容器编排

是什么？

Compose 是 Docker 公司推出的一个工具软件，可以管理多个 Docker 容器组成一个应用。你需要定义一个 YAML 格式的配置文件 `docker-compose.yml`，写好多个容器之间的调用关系。然后，只要一个命令，就能同时启动/关闭这些容器

docker 建议我们每一个容器中只运行一个服务，因为 docker 容器本身占用资源极少，所以最好是将每个服务单独的分割开来。但是这样我们又面临了一个问题？

如果我需要同时部署好多个服务，难道要每个服务单独写 Dockerfile 然后在构建镜像，构建容器，这样累都累死了，所以 docker 官方给我们提供了 docker-compose 多服务部署的工具

例如要实现一个 Web 微服务项目，除了 Web 服务容器本身，往往还需要再加上后端的数据库 mysql 服务容器，redis 服务器，注册中心 eureka，甚至还包括负载均衡容器等等

Compose 允许用户通过一个单独的 `docker-compose.yml` 模板文件（YAML 格式）来定义一组相关联的应用容器为一个项目（project）

可以很容易地用一个配置文件定义一个多容器的应用，然后使用一条指令安装这个应用的所有依赖，完成构建。Docker-Compose 解决了容器与容器之间如何管理编排的问题。

下载

<https://docs.docker.com/compose/install/>

[安装 Docker Compose CLI 插件](#) | [Docker 文档](#)

| Compose file format | Docker Engine release |
|-----------------------|-----------------------|
| Compose specification | 19.03.0+ |
| 3.8 | 19.03.0+ |
| 3.7 | 18.06.0+ |
| 3.6 | 18.02.0+ |
| 3.5 | 17.12.0+ |
| 3.4 | 17.09.0+ |
| 3.3 | 17.06.0+ |
| 3.2 | 17.04.0+ |
| 3.1 | 1.13.1+ |
| 3.0 | 1.13.0+ |
| 2.4 | 17.12.0+ |
| 2.3 | 17.06.0+ |
| 2.2 | 1.13.0+ |
| 2.1 | 1.12.0+ |
| 2.0 | 1.10.0+ |

GitHub 现在需要 TLS1.2。在 PowerShell 中，运行以下命令

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

运行以下命令，下载最新版本的 Compose

```
Invoke-WebRequest
"https://github.com/docker/compose/releases/download/v2.6.0/docker-compose-
windows-x86_64.exe" -UseBasicParsing -OutFile $Env:ProgramFiles\Docker\docker-
compose.exe
```

查看是否安装成功：

```
docker-compose --version
```

```
PS C:\WINDOWS\system32> docker-compose --version
docker-compose version 1.29.2, build 5becea4c
PS C:\WINDOWS\system32>
```

概念

- 一个文件: docker-compose.yml
- 服务: 一个个应用容器实例, 比如订单微服务、库存微服务、mysql容器、nginx容器或者redis容器
- 工程: 由一组关联的应用容器组成的一个完整业务单元, 在docker-compose.yml 文件中定义。

步骤

1. 编写Dockerfile定义各个微服务应用并构建出对应的镜像文件
2. 使用 docker-compose.yml定义一个完整业务单元, 安排好整体应用中的各个容器服务。
3. 最后, 执行docker-compose up命令来启动并运行整个应用程序, 完成一键部署上线

常用命令

帮助

```
docker-compose -h
```

```
PS C:\Users\mao\Desktop> docker-compose -h
Define and run multi-container applications with Docker.

Usage:
  docker-compose [-f <arg>...] [--profile <name>...] [options] [--] [COMMAND]
[ARGS...]
  docker-compose -h|--help

Options:
  -f, --file FILE            specify an alternate compose file
                              (default: docker-compose.yml)
  -p, --project-name NAME    specify an alternate project name
                              (default: directory name)
  --profile NAME             specify a profile to enable
  -c, --context NAME         specify a context name
  --verbose                  show more output
  --log-level LEVEL          set log level (DEBUG, INFO, WARNING, ERROR,
                              CRITICAL)
  --ansi (never|always|auto) control when to print ANSI control characters
```

| | |
|---|---|
| <code>--no-ansi</code> | Do not print ANSI control characters (DEPRECATED) |
| <code>-v, --version</code> | Print version and <code>exit</code> |
| <code>-H, --host HOST</code> | Daemon socket to connect to |
| | |
| <code>--tls</code> | Use TLS; implied by <code>--tlsverify</code> |
| <code>--tlscacert CA_PATH</code> | Trust certs signed only by this CA |
| <code>--tlscert CLIENT_CERT_PATH</code> | Path to TLS certificate file |
| <code>--tlskey TLS_KEY_PATH</code> | Path to TLS key file |
| <code>--tlsverify</code> | Use TLS and verify the remote |
| <code>--skip-hostname-check</code> | Don't <code>check the daemon's</code> hostname against the name specified <code>in</code> the client certificate |
| <code>--project-directory PATH</code> | Specify an alternate working directory (default: the path of the Compose file) |
| <code>--compatibility</code> | If <code>set</code> , Compose will attempt to convert keys <code>in</code> v3 files to their non-Swarm equivalent |
| (DEPRECATED) | |
| <code>--env-file PATH</code> | Specify an alternate environment file |

Commands:

| | |
|----------------------|--|
| <code>build</code> | Build or rebuild services |
| <code>config</code> | Validate and view the Compose file |
| <code>create</code> | Create services |
| <code>down</code> | Stop and remove resources |
| <code>events</code> | Receive real time events from containers |
| <code>exec</code> | Execute a command <code>in</code> a running container |
| <code>help</code> | Get help on a command |
| <code>images</code> | List images |
| <code>kill</code> | Kill containers |
| <code>logs</code> | View output from containers |
| <code>pause</code> | Pause services |
| <code>port</code> | Print the public port <code>for</code> a port binding |
| <code>ps</code> | List containers |
| <code>pull</code> | Pull <code>service</code> images |
| <code>push</code> | Push <code>service</code> images |
| <code>restart</code> | Restart services |
| <code>rm</code> | Remove stopped containers |
| <code>run</code> | Run a one-off command |
| <code>scale</code> | Set number of containers <code>for</code> a <code>service</code> |
| <code>start</code> | Start services |
| <code>stop</code> | Stop services |
| <code>top</code> | Display the running processes |
| <code>unpause</code> | Unpause services |
| <code>up</code> | Create and <code>start</code> containers |
| <code>version</code> | Show version information and quit |

Docker Compose is now `in` the Docker CLI, try ``docker compose``
 PS C:\Users\mao\Desktop>

启动服务

启动所有docker-compose服务

```
docker-compose up
```

启动所有docker-compose服务并后台运行

```
docker-compose up -d
```

停止并删除

停止并删除容器、网络、卷、镜像

```
docker-compose down
```

进入容器实例内部

```
docker-compose exec yml文件里面的服务id
```

相当于

```
docker-compose exec docker-compose.yml文件中写的服务id /bin/bash
```

展示编排过的容器

展示当前docker-compose编排过的运行的所有容器

```
docker-compose ps
```

展示编排过的容器进程

```
docker-compose top
```

查看容器输出日志

```
docker-compose logs yml文件里面的服务id
```


检查配置

```
docker-compose config
```

静默检查配置

```
docker-compose config -q
```

重启服务

```
docker-compose restart
```

启动服务

```
docker-compose start
```

停止服务

```
docker-compose stop
```

示例

创建springboot项目

项目名称为Docker_compose_boot

maven

maven的pom.xml文件:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
```

```
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.7.0</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>mao</groupId>
<artifactId>Docker_compose_boot</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>Docker_compose_boot</name>
<description>Docker_compose_boot</description>
<properties>
    <java.version>11</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!--mysql 依赖 spring-boot-->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
    <!--spring-boot druid连接池依赖-->
    <dependency>
        <groupId>com.alibaba</groupId>
        <artifactId>druid-spring-boot-starter</artifactId>
        <version>1.2.8</version>
    </dependency>
    <!--spring boot redis 依赖-->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-redis</artifactId>
    </dependency>
    <!--spring-boot 开发者工具 主要用于热部署-->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <optional>true</optional>
    </dependency>
    <!--spring-boot mybatis-plus依赖-->
    <dependency>
        <groupId>com.baomidou</groupId>
        <artifactId>mybatis-plus-boot-starter</artifactId>
        <version>3.5.1</version>
    </dependency>
    <!--mybatis-plus代码生成器-->
    <dependency>
        <groupId>com.baomidou</groupId>
        <artifactId>mybatis-plus-generator</artifactId>
        <version>3.5.1</version>
    </dependency>
    <!-- 模板引擎 默认 -->
    <dependency>
        <groupId>org.apache.velocity</groupId>
        <artifactId>velocity-engine-core</artifactId>
```

```

        <version>2.3</version>
    </dependency>
    <!--swagger-->
    <dependency>
        <groupId>io.springfox</groupId>
        <artifactId>springfox-swagger2</artifactId>
        <version>2.7.0</version>
    </dependency>
    <dependency>
        <groupId>cn.hutool</groupId>
        <artifactId>hutool-all</artifactId>
        <version>5.8.0</version>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>

```

配置

application.yml文件:

```

#多环境开发---单文件
spring:
  profiles:
    # 当前环境
    active: dev

#-----通用环境-----

# 导入jdbc.properties文件
# 导入redis配置文件
config:
  import: classpath:jdbc.properties,classpath:redis.properties

#spring-mvc配置

```

```
mvc:
  # 视图解析器
view:
  prefix:
  suffix: .html
```

```
server:
  #配置端口号
  port: 8080
```

#-----开发环境-----

```
spring:
  config:
    activate:
      on-profile: dev
```

配置数据源

```
datasource:
  # driver-class-name: ${jdbc.driver}
  # url: ${jdbc.url}
  # username: ${jdbc.username}
  # password: ${jdbc.password}
  # 配置数据源-druid
  druid:
    driver-class-name: ${jdbc.driver}
    url: ${jdbc.url}
    username: ${jdbc.username}
    password: ${jdbc.password}
```

关于redis

```
redis:
  host: ${redis.host} # "127.0.0.1"
  port: ${redis.port} # 6379
  password: ${redis.password} # "123456"
  jedis:
    pool:
      # 连接池开关
      enabled: true
      # 在给定时间可以分配的最大连接数
      max-active: ${max-active} # 8
      #池中“空闲”连接的最大数量。使用负值表示无限数量的空闲连接。
      max-idle: ${max-idle} # 8
      # 当池耗尽时，连接分配在抛出异常之前应该阻塞的最长时间
      max-wait: ${max-wait} # 2000
```

```

# 关于热部署
devtools:
  restart:
    # 热部署开关
    enabled: true
    # 自定义不参与重启排除项
    exclude: public/**,static/**

# 开启debug模式，输出调试信息，常用于检查系统运行状况
#debug: true

# 设置日志级别，root表示根节点，即整体应用日志级别
logging:
  # 日志输出到文件的文件名
  file:
    name: server.log
  # 字符集
  charset:
    file: UTF-8
  # 分文件
  logback:
    rollingpolicy:
      #最大文件大小
      max-file-size: 16KB
      # 文件格式
      file-name-pattern: server_log-%d{yyyy/MM月/dd日/}%i.log
  # 设置日志组
  group:
    # 自定义组名，设置当前组中所包含的包
    mao_pro: mao
  level:
    root: info
    # 为对应组设置日志级别
    mao_pro: debug
    # 日志输出格式
  # pattern:
    # console: "%d %clr(%p) --- [%16t] %clr(%-40.40c){cyan} : %m %n"

---
#----生产环境-----

spring:
  config:
    activate:
      on-profile: pro

# 开启debug模式，输出调试信息，常用于检查系统运行状况
#debug: true

```

```

# 设置日志级别，root表示根节点，即整体应用日志级别
logging:
  # 日志输出到文件的文件名
  file:
    name: server.log
  # 字符集
  charset:
    file: UTF-8
  # 分文件
  logback:
    rollingpolicy:
      #最大文件大小
      max-file-size: 4MB
      # 文件格式
      file-name-pattern: server_log-%d{yyyy/MM月/dd日/}%i.log
  # 设置日志组
  group:
    # 自定义组名，设置当前组中所包含的包
    mao_pro: mao
  level:
    root: info
    # 为对应组设置日志级别
    mao_pro: warn
    # 日志输出格式
  # pattern:
  # console: "%d %clr(%p) --- [%16t] %clr(%-40.40c){cyan} : %m %n"

---

#----测试环境-----

spring:
  config:
    activate:
      on-profile: test

```

业务类

```

package mao.docker_compose_boot.controller;

import mao.docker_compose_boot.entity.Student;
import mao.docker_compose_boot.service.IStudentService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

```

```

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

/**
 * <p>
 * 前端控制器
 * </p>
 *
 * @author mao
 * @since 2022-06-19
 */

@RequestMapping("/student")
@RestController()
public class StudentController
{

    @Autowired
    private IStudentService studentService;

    @GetMapping("")
    public List<Student> getAll()
    {
        return studentService.list();
    }

    @GetMapping("/{id}")
    public Student getStudent(@PathVariable Long id)
    {
        return studentService.getStudent(id);
    }

}

```

```

package mao.docker_compose_boot.entity;

import com.baomidou.mybatisplus.annotation.IdType;
import com.baomidou.mybatisplus.annotation.TableId;

import java.io.Serializable;

import io.swagger.annotations.ApiModel;
import io.swagger.annotations.ApiModelProperty;

/**
 * <p>
 *
 * </p>
 *

```

```

* @author mao
* @since 2022 -06-19
*/
@ApiModel(value = "student对象", description = "")
public class Student implements Serializable
{

    private static final long serialVersionUID = 1L;

    @ApiModelProperty("学生学号")
    @TableId(value = "student_no", type = IdType.AUTO)
    private Long studentNo;

    @ApiModelProperty("学生姓名")
    private String name;

    @ApiModelProperty("学生性别")
    private String sex;

    @ApiModelProperty("电话号码")
    private String telephoneNumber;

    @ApiModelProperty("家庭电话号码")
    private String familyTelephoneNumber;

    @ApiModelProperty("生日信息，由程序填充，这里用varchar型")
    private String birthday;

    @ApiModelProperty("家庭地址")
    private String address;

    @ApiModelProperty("身份证号码")
    private String idCard;

    @ApiModelProperty("电子邮箱")
    private String email;

    @ApiModelProperty("寝室号")
    private String dormitoryNumber;

    @ApiModelProperty("所属班级编号")
    private Long classNo;

    @ApiModelProperty("状态，在读、退学或者毕业")
    private String state;

    @ApiModelProperty("备注")
    private String remarks;

    /**
     * Gets student no.
     *
     * @return the student no
     */
    public Long getStudentNo()
    {
        return studentNo;
    }

```



```
}

/**
 * Sets student no.
 *
 * @param studentNo the student no
 */
public void setStudentNo(Long studentNo)
{
    this.studentNo = studentNo;
}

/**
 * Gets name.
 *
 * @return the name
 */
public String getName()
{
    return name;
}

/**
 * Sets name.
 *
 * @param name the name
 */
public void setName(String name)
{
    this.name = name;
}

/**
 * Gets sex.
 *
 * @return the sex
 */
public String getSex()
{
    return sex;
}

/**
 * Sets sex.
 *
 * @param sex the sex
 */
public void setSex(String sex)
{
    this.sex = sex;
}

/**
 * Gets telephone number.
 *
 * @return the telephone number
 */
public String getTelephoneNumber()
```

```

{
    return telephoneNumber;
}

/**
 * Sets telephone number.
 *
 * @param telephoneNumber the telephone number
 */
public void setTelephoneNumber(String telephoneNumber)
{
    this.telephoneNumber = telephoneNumber;
}

/**
 * Gets family telephone number.
 *
 * @return the family telephone number
 */
public String getFamilyTelephoneNumber()
{
    return familyTelephoneNumber;
}

/**
 * Sets family telephone number.
 *
 * @param familyTelephoneNumber the family telephone number
 */
public void setFamilyTelephoneNumber(String familyTelephoneNumber)
{
    this.familyTelephoneNumber = familyTelephoneNumber;
}

/**
 * Gets birthday.
 *
 * @return the birthday
 */
public String getBirthday()
{
    return birthday;
}

/**
 * Sets birthday.
 *
 * @param birthday the birthday
 */
public void setBirthday(String birthday)
{
    this.birthday = birthday;
}

/**
 * Gets address.
 *
 * @return the address

```

```

    */
    public String getAddress()
    {
        return address;
    }

    /**
     * Sets address.
     *
     * @param address the address
     */
    public void setAddress(String address)
    {
        this.address = address;
    }

    /**
     * Gets id card.
     *
     * @return the id card
     */
    public String getIdCard()
    {
        return idCard;
    }

    /**
     * Sets id card.
     *
     * @param idCard the id card
     */
    public void setIdCard(String idCard)
    {
        this.idCard = idCard;
    }

    /**
     * Gets email.
     *
     * @return the email
     */
    public String getEmail()
    {
        return email;
    }

    /**
     * Sets email.
     *
     * @param email the email
     */
    public void setEmail(String email)
    {
        this.email = email;
    }

    /**
     * Gets dormitory number.

```

```

    *
    * @return the dormitory number
    */
    public String getDormitoryNumber()
    {
        return dormitoryNumber;
    }

    /**
     * Sets dormitory number.
     *
     * @param dormitoryNumber the dormitory number
     */
    public void setDormitoryNumber(String dormitoryNumber)
    {
        this.dormitoryNumber = dormitoryNumber;
    }

    /**
     * Gets class no.
     *
     * @return the class no
     */
    public Long getClassNo()
    {
        return classNo;
    }

    /**
     * Sets class no.
     *
     * @param classNo the class no
     */
    public void setClassNo(Long classNo)
    {
        this.classNo = classNo;
    }

    /**
     * Gets state.
     *
     * @return the state
     */
    public String getState()
    {
        return state;
    }

    /**
     * Sets state.
     *
     * @param state the state
     */
    public void setState(String state)
    {
        this.state = state;
    }

```

```

/**
 * Gets remarks.
 *
 * @return the remarks
 */
public String getRemarks()
{
    return remarks;
}

/**
 * Sets remarks.
 *
 * @param remarks the remarks
 */
public void setRemarks(String remarks)
{
    this.remarks = remarks;
}

@Override
@SuppressWarnings("all")
public String toString()
{
    final StringBuilder stringbuilder = new StringBuilder();
    stringbuilder.append("studentNo: ").append(studentNo).append('\n');
    stringbuilder.append("name: ").append(name).append('\n');
    stringbuilder.append("sex: ").append(sex).append('\n');

    stringbuilder.append("telephoneNumber: ").append(telephoneNumber).append('\n');

    stringbuilder.append("familyTelephoneNumber: ").append(familyTelephoneNumber).append('\n');
    stringbuilder.append("birthday: ").append(birthday).append('\n');
    stringbuilder.append("address: ").append(address).append('\n');
    stringbuilder.append("idCard: ").append(idCard).append('\n');
    stringbuilder.append("email: ").append(email).append('\n');

    stringbuilder.append("dormitoryNumber: ").append(dormitoryNumber).append('\n');
    stringbuilder.append("classNo: ").append(classNo).append('\n');
    stringbuilder.append("state: ").append(state).append('\n');
    stringbuilder.append("remarks: ").append(remarks).append('\n');
    return stringbuilder.toString();
}
}

```

```

package mao.docker_compose_boot.mapper;

import mao.docker_compose_boot.entity.Student;
import com.baomidou.mybatisplus.core.mapper.BaseMapper;
import org.apache.ibatis.annotations.Mapper;

```

```

/**
 * <p>
 * Mapper 接口
 * </p>
 *
 * @author mao
 * @since 2022-06-19
 */

@Mapper
public interface StudentMapper extends BaseMapper<Student>
{

}

```

```

package mao.docker_compose_boot.service;

import mao.docker_compose_boot.entity.Student;
import com.baomidou.mybatisplus.extension.service.IService;

/**
 * <p>
 * 服务类
 * </p>
 *
 * @author mao
 * @since 2022-06-19
 */
public interface IStudentService extends IService<Student>
{

    /**
     * 获得学生信息
     *
     * @param id 学生的学号, (no)
     * @return Student
     */
    Student getStudent(Long id);
}

```

```

package mao.docker_compose_boot.service.impl;

import mao.docker_compose_boot.entity.Student;
import mao.docker_compose_boot.mapper.StudentMapper;
import mao.docker_compose_boot.service.IStudentService;
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;
import mao.docker_compose_boot.utils.RedisUtils;

```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.concurrent.TimeUnit;

/**
 * <p>
 * 服务实现类
 * </p>
 *
 * @author mao
 * @since 2022-06-19
 */
@Service
public class StudentServiceImpl extends ServiceImpl<StudentMapper, Student>
implements IStudentService
{

    @Autowired
    private RedisUtils redisUtils;

    @Override
    public Student getStudent(Long id)
    {
        return redisUtils.query("docker_compose_boot:student:",
            "docker_compose_boot:lock:",
                id, Student.class, this::getById,
                300L, TimeUnit.SECONDS, 50);
    }
}

```

```

package mao.docker_compose_boot.utils;

import cn.hutool.core.util.BooleanUtil;
import cn.hutool.core.util.StrUtil;
import cn.hutool.json.JSONObject;
import cn.hutool.json.JSONUtil;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.data.redis.core.StringRedisTemplate;
import org.springframework.data.redis.core.convert.RedisData;
import org.springframework.stereotype.Component;

import javax.annotation.Resource;
import java.time.LocalDateTime;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;
import java.util.function.Function;

/**
 * Project name(项目名称): Docker_compose_boot

```

```
* Package(包名): mao.docker_compose_boot.utils
* Class(类名): RedisUtils
* Author(作者): mao
* Author QQ: 1296193245
* GitHub: https://github.com/maomao124/
* Date(创建日期): 2022/6/19
* Time(创建时间): 13:25
* Version(版本): 1.0
* Description(描述): 无
*/
```

@Component

public class RedisUtils

{

 @Resource

 StringRedisTemplate stringRedisTemplate;

 private static final Logger log = LoggerFactory.getLogger(RedisUtils.class);

 //线程池

 private static final ExecutorService CACHE_REBUILD_EXECUTOR =
Executors.newFixedThreadPool(10);

 /**

 * 向redis里添加数据

 *

 * @param redisKey redis的key

 * @param value 数据

 * @param expireTime 过期时间

 * @param timeUnit 时间单位

 */

 public void set(String redisKey, Object value, Long expireTime, TimeUnit
timeUnit)

 {

 stringRedisTemplate.opsForValue().set(redisKey,
JSONUtil.toJsonStr(value), expireTime, timeUnit);

 }

// /**

// * 向redis里添加数据 设置逻辑过期

// *

// * @param redisKey redis的key

// * @param value 数据

// * @param expireTime 过期时间

// * @param timeUnit 时间单位

// */

// public void setwithLogicalExpire(String redisKey, Object value, Long
expireTime, TimeUnit timeUnit)

// {

// RedisData redisData = new RedisData();

// //添加数据

// redisData.setData(value);

// //设置过期时间


```

//
redisData.setExpireTime(LocalDateTime.now().plusSeconds(timeUnit.toSeconds(expireTime)));
//          //放入redis
//          stringRedisTemplate.opsForValue().set(redisKey,
JSONUtil.toJsonStr(redisData));
//      }

/**
 * 查询数据，有缓存，解决缓存穿透问题，未解决缓存雪崩问题
 *
 * @param keyPrefix redisKey的前缀
 * @param id        id
 * @param type       返回值的类型
 * @param dbFallback 查询数据库的函数
 * @param expireTime 过期时间
 * @param timeUnit   时间单位
 * @param <R>         返回值的类型
 * @param <ID>        id的类型
 * @return 泛型R
 */
public <R, ID> R queryWithPassThrough(String keyPrefix, ID id, Class<R>
type,
                                     Function<ID, R> dbFallback, Long
expireTime, TimeUnit timeUnit)
{
    //获得前缀
    String redisKey = keyPrefix + id;
    //查询redis
    String json = stringRedisTemplate.opsForValue().get(redisKey);
    //判断是否为空
    if (StrUtil.isNotBlank(json))
    {
        //不为空，返回
        return JSONUtil.toBean(json, type);
    }
    //判断是否为空串
    if (json != null)
    {
        //空串
        return null;
    }
    //null
    //查数据库
    R r = dbFallback.apply(id);
    //判断
    if (r == null)
    {
        //数据库也为空，缓存空值
        this.set(redisKey, "", expireTime, timeUnit);
        return null;
    }
    //数据库存在，写入redis
    this.set(redisKey, r, expireTime, timeUnit);
    //返回
    return r;
}

```

```

/**
 * 查询数据，有缓存，解决缓存穿透问题，解决缓存雪崩问题
 *
 * @param keyPrefix          redisKey的前缀
 * @param id                 id
 * @param type               返回值的类型
 * @param dbFallback         查询数据库的函数
 * @param expireTime         过期时间
 * @param timeUnit           时间单位
 * @param <R>                返回值的类型
 * @param <ID>               id的类型
 * @param maxTimeSecondsByCacheAvalanche this.set(redisKey, r,
 *
timeUnit.toSeconds(expireTime)+getIntRandom(0,maxTimeSecondsByCacheAvalanche),
TimeUnit.SECONDS);
 * @return 泛型R
 */
public <R, ID> R queryWithPassThroughAndCacheAvalanche(String keyPrefix, ID
id, Class<R> type,
                                                    Function<ID, R>
dbFallback, Long expireTime, TimeUnit timeUnit,
                                                    Integer
maxTimeSecondsByCacheAvalanche)
{
    //获得前缀
    String redisKey = keyPrefix + id;
    //查询redis
    String json = stringRedisTemplate.opsForValue().get(redisKey);
    //判断是否为空
    if (StringUtil.isNotBlank(json))
    {
        //不为空，返回
        return JSONUtil.toBean(json, type);
    }
    //判断是否为空串
    if (json != null)
    {
        //空串
        return null;
    }
    //null
    //查数据库
    R r = dbFallback.apply(id);
    //判断
    if (r == null)
    {
        //数据库也为空，缓存空值
        this.set(redisKey, "",
            timeUnit.toSeconds(expireTime) + getIntRandom(0,
maxTimeSecondsByCacheAvalanche),
            TimeUnit.SECONDS);
        return null;
    }
    //数据库存在，写入redis
    this.set(redisKey, r,
        timeUnit.toSeconds(expireTime) + getIntRandom(0,
maxTimeSecondsByCacheAvalanche),

```

```

        TimeUnit.SECONDS);

        //返回
        return r;
    }

    /**
     * 查询数据，解决缓存穿透，互斥锁方法解决缓存击穿，解决缓存雪崩
     *
     * @param keyPrefix          redisKey的前缀
     * @param lockKeyPrefix      锁的前缀
     * @param id                  id
     * @param type                返回值的类型
     * @param dbFallback          查询数据库的函数
     * @param expireTime          过期时间
     * @param timeUnit            时间单位
     * @param <R>                  返回值的类型
     * @param <ID>                 id的类型
     * @param maxTimeSecondsByCacheAvalanche this.set(redisKey, r,
     *
     * timeUnit.toSeconds(expireTime)+getIntrandom(0,maxTimeSecondsByCacheAvalanche),
     * TimeUnit.SECONDS);
     * @return 泛型R
     */
    public <R, ID> R query(String keyPrefix, String lockKeyPrefix, ID id,
        Class<R> type,
                               Function<ID, R> dbFallback, Long expireTime, TimeUnit
        timeUnit,
                               Integer maxTimeSecondsByCacheAvalanche)
    {
        //获取redisKey
        String redisKey = keyPrefix + id;
        //从redis中查询信息，根据id
        String json = stringRedisTemplate.opsForValue().get(redisKey);
        //判断取出的数据是否为空
        if (StrUtil.isNotBlank(json))
        {
            //不是空，redis里有，返回
            return JSONUtil.toBean(json, type);
        }
        //是空串，不是null，返回
        if (json != null)
        {
            return null;
        }
        //锁的key
        String lockKey = lockKeyPrefix + id;

        R r = null;
        try
        {
            //获取互斥锁
            boolean lock = tryLock(lockKey);
            //判断锁是否获取成功
            if (!lock)
            {
                //没有获取到锁
                //200毫秒后再次获取
                Thread.sleep(200);
            }
        }
    }

```

```

        //递归调用
        return query(keyPrefix, lockKeyPrefix, id, type, dbFallback,
            expireTime, timeUnit, maxTimeSecondsByCacheAvalanche);
    }
    //得到了锁
    //null, 查数据库
    r = dbFallback.apply(id);
    //判断数据库里的信息是否为空
    if (r == null)
    {
        //数据库也为空, 缓存空值
        this.set(redisKey, "",
            timeUnit.toSeconds(expireTime) + getIntRandom(0,
maxTimeSecondsByCacheAvalanche),
            TimeUnit.SECONDS);
        return null;
    }
    //存在, 回写到redis里, 设置随机的过期时间
    this.set(redisKey, r,
        timeUnit.toSeconds(expireTime) + getIntRandom(0,
maxTimeSecondsByCacheAvalanche),
        TimeUnit.SECONDS);
}
catch (InterruptedException e)
{
    throw new RuntimeException(e);
}
finally
{
    //释放锁
    this.unlock(lockKey);
}
//返回数据
return r;
}

/**
 * 更新数据
 *
 * @param id 要更新的主键
 * @param data 要更新的对象
 * @param keyPrefix redis的key前缀
 * @param dbFallback 更新数据库的函数, 返回值要为Boolean类型
 * @param <T> 要更新的对象泛型
 * @param <ID> 主键的类型
 * @return boolean
 */
public <T, ID> boolean update(ID id, T data, String keyPrefix, Function<T,
Boolean> dbFallback)
{
    //判断是否为空
    if (id == null)
    {
        return false;
    }
    //不为空
    //先更新数据库
    boolean b = dbFallback.apply(data);

```

```

        //更新失败, 返回
        if (!b)
        {
            return false;
        }
        //更新没有失败
        //删除redis里的数据, 下一次查询时自动添加进redis
        //redisKey
        String redisKey = keyPrefix + id;
        stringRedisTemplate.delete(redisKey);
        //返回响应
        return true;
    }

    /**
     * @param keyPrefix    redisKey的前缀
     * @param lockKeyPrefix 锁的前缀
     * @param id           id
     * @param type         返回值的类型
     * @param dbFallback   查询数据库的函数
     * @param time         过期时间
     * @param timeUnit     时间单位
     * @param <R>          返回值的类型
     * @param <ID>         id的类型
     * @return 泛型R
     */
    public <R, ID> R queryWithLogicalExpire(String keyPrefix, String
lockKeyPrefix, ID id, Class<R> type,
                                           Function<ID, R> dbFallback, Long
time, TimeUnit timeUnit)
    {
        //获得前缀
        String redisKey = keyPrefix + id;
        //查询redis
        String json = stringRedisTemplate.opsForValue().get(redisKey);
        //判断是否为空
        if (StrUtil.isBlank(json))
        {
            //空, 返回
            return null;
        }
        //不为空
        //json 反序列化为对象
        RedisData redisData = JSONUtil.toBean(json, RedisData.class);
        //获得过期时间
        LocalDateTime expireTime = redisData.getExpireTime();
        //获取数据
        R r = JSONUtil.toBean((JSONObject) redisData.getData(), type);
        //判断是否过期
        if (expireTime.isAfter(LocalDateTime.now()))
        {
            //未过期, 返回
            return r;
        }
        //过期, 缓存重建
        //获取互斥锁
        String lockKey = lockKeyPrefix + id;
        boolean isLock = tryLock(lockKey);
    }

```

```

//      if (isLock)
//      {
//          //获取锁成功
//          // 开辟独立线程
//          CACHE_REBUILD_EXECUTOR.submit(new Runnable()
//          {
//              @Override
//              public void run()
//              {
//                  try
//                  {
//                      R r1 = dbFallback.apply(id);
//                      setWithLogicalExpire(redisKey, r1, time, timeUnit);
//                  }
//                  catch (Exception e)
//                  {
//                      throw new RuntimeException(e);
//                  }
//                  finally
//                  {
//                      //释放锁
//                      unlock(lockKey);
//                  }
//              }
//          });
//      }
//      //没有获取到锁，使用旧数据返回
//      return r;
//  }

/**
 * 获取锁
 *
 * @param key redisKey
 * @return 获取锁成功，返回true，否则返回false
 */
private boolean tryLock(String key)
{
    Boolean result = stringRedisTemplate.opsForValue().setIfAbsent(key, "1",
        20, TimeUnit.SECONDS);
    return BooleanUtil.isTrue(result);
}

/**
 * 释放锁
 *
 * @param key redisKey
 */
private void unlock(String key)
{
    stringRedisTemplate.delete(key);
}

/**
 * 获取一个随机数，区间包含min和max
 *

```

```

    * @param min 最小值
    * @param max 最大值
    * @return int 型的随机数
    */
    @SuppressWarnings("all")
    private int getIntRandom(int min, int max)
    {
        if (min > max)
        {
            min = max;
        }
        return min + (int) (Math.random() * (max - min + 1));
    }
}

```

```

package mao.docker_compose_boot;

import com.baomidou.mybatisplus.generator.FastAutoGenerator;
import com.baomidou.mybatisplus.generator.config.OutputFile;

import java.io.IOException;
import java.io.InputStream;
import java.util.Collections;
import java.util.Properties;

/**
 * Project name(项目名称): Docker_compose_boot
 * Package(包名): mao.docker_compose_boot
 * Class(类名): MybatisPlusSqlGenerator
 * Author(作者): mao
 * Author QQ: 1296193245
 * GitHub: https://github.com/maomao124/
 * Date(创建日期): 2022/6/19
 * Time(创建时间): 13:16
 * Version(版本): 1.0
 * Description(描述):
 */

public class MybatisPlusSqlGenerator
{
    public static void main(String[] args) throws IOException
    {
        /**
         //旧
         // 1、创建代码生成器
         AutoGenerator autoGenerator = new AutoGenerator();

         // 2、全局配置
         GlobalConfig globalConfig = new GlobalConfig();
         String projectPath = System.getProperty("user.dir");
         // 项目输出根目录
         globalConfig.setOutputDir(projectPath + "/src/main/java");

```

```

// 作者
globalConfig.setAuthor("mao");
//生成后是否打开资源管理器
globalConfig.setOpen(false);
//去掉Service接口的首字母I
globalConfig.setServiceName("%sService");
//主键策略
globalConfig.setIdType(IdType.AUTO);
//开启Swagger2模式
globalConfig.setSwagger2(true);
autoGenerator.setGlobalConfig(globalConfig);

// 3、数据源配置
InputStream inputStream =
MybatisPlusSqlGenerator.class.getClassLoader().getResourceAsStream("jdbc.properties");

Properties properties=new Properties();
properties.load(inputStream);
DataSourceConfig dataSourceConfig = new DataSourceConfig();
dataSourceConfig.setUrl(properties.getProperty("jdbc.url"));
dataSourceConfig.setDriverName(properties.getProperty("jdbc.driver"));
dataSourceConfig.setUsername(properties.getProperty("jdbc.username"));
dataSourceConfig.setPassword(properties.getProperty("jdbc.password"));
//设置方言
dataSourceConfig.setDbType(DbType.MYSQL);
autoGenerator.setDataSource(dataSourceConfig);

// 4、包配置
PackageConfig packageConfig = new PackageConfig();
// 生成的文件放在那个目录下 若没有该文件 则会创建
packageConfig.setParent("com.alibaba.product");
//此对象与数据库表结构一一对应，通过 DAO 层向上传输数据源对象。
packageConfig.setEntity("pojo.entity");
autoGenerator.setPackageInfo(packageConfig);

// 5、策略配置
StrategyConfig strategy = new StrategyConfig();
//数据库表映射到实体的命名策略
strategy.setNaming(NamingStrategy.underline_to_camel);
//数据库表字段映射到实体的命名策略
strategy.setColumnNaming(NamingStrategy.underline_to_camel);
// Lombok
strategy.setEntityLombokModel(false);
//逻辑删除字段名
strategy.setLogicDeleteFieldName("is_deleted");
//去掉布尔值的is_前缀（确保tinyint(1)）
strategy.setEntityBooleanColumnRemoveIsPrefix(true);
//restful api风格控制器
strategy.setRestControllerStyle(true);
autoGenerator.setStrategy(strategy);

// 6、执行
autoGenerator.execute();
*/

//新

```



```

        /*InputStream inputStream =
MybatisPlusSqlGenerator.class.getClassLoader().getResourceAsStream("jdbc.properties");

        Properties properties = new Properties();
        properties.load(inputStream);
        FastAutoGenerator.create(properties.getProperty("jdbc.url"),
                                properties.getProperty("jdbc.username"),
                                properties.getProperty("jdbc.password"))
            .globalConfig(builder ->
            {
                builder.author("mao") // 设置作者
                    .enableSwagger() // 开启 swagger 模式
                    .fileOverride() // 覆盖已生成文件
                    .outputDir("src\\main\\java"); // 指定输出目录
            })
            .packageConfig(builder ->
            {
                builder.parent("mao.docker_compose_boot") // 设置父包名
                    //.moduleName("system") // 设置父包模块名

            })
            .pathInfo(Collections.singletonMap(OutputFile.mapperXml,
"src\\main\\resources\\mapper")); // 设置mapperXml生成路径
            })
            .strategyConfig(builder ->
            {
                builder.addInclude("student"); // 设置需要生成的表名
                    //.addTablePrefix("t_", "c_"); // 设置过滤表前缀
            })
            //.templateEngine(new FreemarkerTemplateEngine()) // 使用
Freemarker引擎模板，默认的是Velocity引擎模板
            .execute();*/
    }
}

```

启动测试

<localhost:8080/student/202012340112>

<localhost:8080/student/202012340113>

```
C:\Users\mao\.jdk\openjdk-16.0.2\bin\java.exe -XX:TieredStopAtLevel=1 -noverify
-Dspring.output.ansi.enabled=always "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2021.2.2\lib\idea_rt.jar=53775:C:\Program
Files\JetBrains\IntelliJ IDEA 2021.2.2\bin" -Dcom.sun.management.jmxremote -
Dspring.jmx.enabled=true -Dspring.liveBeansView.mbeanDomain -
Dspring.application.admin.enabled=true -Dfile.encoding=UTF-8 -classpath H:\程序\大
三下期
```

```
\Docker_compose_boot\target\classes;C:\Users\mao\.m2\repository\org\springframew
ork\boot\spring-boot-starter-web\2.7.0\spring-boot-starter-web-
2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter\2.7.0\spring-boot-starter-
2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter-logging\2.7.0\spring-boot-starter-logging-
2.7.0.jar;C:\Users\mao\.m2\repository\ch\qos\logback\logback-
classic\1.2.11\logback-classic-
1.2.11.jar;C:\Users\mao\.m2\repository\ch\qos\logback\logback-
core\1.2.11\logback-core-
1.2.11.jar;C:\Users\mao\.m2\repository\org\apache\logging\log4j\log4j-to-
slf4j\2.17.2\log4j-to-slf4j-
2.17.2.jar;C:\Users\mao\.m2\repository\org\apache\logging\log4j\log4j-
api\2.17.2\log4j-api-2.17.2.jar;C:\Users\mao\.m2\repository\org\slf4j\jul-to-
slf4j\1.7.36\jul-to-slf4j-
1.7.36.jar;C:\Users\mao\.m2\repository\jakarta\annotation\jakarta.annotation-
api\1.3.5\jakarta.annotation-api-
1.3.5.jar;C:\Users\mao\.m2\repository\org\yaml\snakeyaml\1.30\snakeyaml-
1.30.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter-json\2.7.0\spring-boot-starter-json-
2.7.0.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\core\jackson-
databind\2.13.3\jackson-databind-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\core\jackson-
core\2.13.3\jackson-core-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\datatype\jackson-
datatype-jdk8\2.13.3\jackson-datatype-jdk8-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\datatype\jackson-
datatype-jsr310\2.13.3\jackson-datatype-jsr310-
2.13.3.jar;C:\Users\mao\.m2\repository\com\fasterxml\jackson\module\jackson-
module-parameter-names\2.13.3\jackson-module-parameter-names-
2.13.3.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-
starter-tomcat\2.7.0\spring-boot-starter-tomcat-
2.7.0.jar;C:\Users\mao\.m2\repository\org\apache\tomcat\embed\tomcat-embed-
core\9.0.63\tomcat-embed-core-
9.0.63.jar;C:\Users\mao\.m2\repository\org\apache\tomcat\embed\tomcat-embed-
el\9.0.63\tomcat-embed-el-
9.0.63.jar;C:\Users\mao\.m2\repository\org\apache\tomcat\embed\tomcat-embed-
websocket\9.0.63\tomcat-embed-websocket-
9.0.63.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
web\5.3.20\spring-web-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
beans\5.3.20\spring-beans-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
webmvc\5.3.20\spring-webmvc-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
aop\5.3.20\spring-aop-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
context\5.3.20\spring-context-
5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-
expression\5.3.20\spring-expression-
5.3.20.jar;C:\Users\mao\.m2\repository\mysql\mysql-connector-java\8.0.29\mysql-
```

connector-java-8.0.29.jar;C:\Users\mao\.m2\repository\com\alibaba\druid-spring-boot-starter\1.2.8\druid-spring-boot-starter-1.2.8.jar;C:\Users\mao\.m2\repository\com\alibaba\druid\1.2.8\druid-1.2.8.jar;C:\Users\mao\.m2\repository\javax\annotation\javax.annotation-api\1.3.2\javax.annotation-api-1.3.2.jar;C:\Users\mao\.m2\repository\org\slf4j\slf4j-api\1.7.36\slf4j-api-1.7.36.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-autoconfigure\2.7.0\spring-boot-autoconfigure-2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-starter-data-redis\2.7.0\spring-boot-starter-data-redis-2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\data\spring-data-redis\2.7.0\spring-data-redis-2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\data\spring-data-keyvalue\2.7.0\spring-data-keyvalue-2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\data\spring-data-commons\2.7.0\spring-data-commons-2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\spring-tx\5.3.20\spring-tx-5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-oxm\5.3.20\spring-oxm-5.3.20.jar;C:\Users\mao\.m2\repository\org\springframework\spring-context-support\5.3.20\spring-context-support-5.3.20.jar;C:\Users\mao\.m2\repository\io\lettuce\lettuce-core\6.1.8.RELEASE\lettuce-core-6.1.8.RELEASE.jar;C:\Users\mao\.m2\repository\io\netty\netty-common\4.1.77.Final\netty-common-4.1.77.Final.jar;C:\Users\mao\.m2\repository\io\netty\netty-handler\4.1.77.Final\netty-handler-4.1.77.Final.jar;C:\Users\mao\.m2\repository\io\netty\netty-resolver\4.1.77.Final\netty-resolver-4.1.77.Final.jar;C:\Users\mao\.m2\repository\io\netty\netty-buffer\4.1.77.Final\netty-buffer-4.1.77.Final.jar;C:\Users\mao\.m2\repository\io\netty\netty-codec\4.1.77.Final\netty-codec-4.1.77.Final.jar;C:\Users\mao\.m2\repository\io\netty\netty-transport\4.1.77.Final\netty-transport-4.1.77.Final.jar;C:\Users\mao\.m2\repository\io\projectreactor\reactor-core\3.4.18\reactor-core-3.4.18.jar;C:\Users\mao\.m2\repository\org\reactivestreams\reactive-streams\1.0.3\reactive-streams-1.0.3.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-devtools\2.7.0\spring-boot-devtools-2.7.0.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot\2.7.0\spring-boot-2.7.0.jar;C:\Users\mao\.m2\repository\com\baomidou\mybatis-plus-boot-starter\3.5.1\mybatis-plus-boot-starter-3.5.1.jar;C:\Users\mao\.m2\repository\com\baomidou\mybatis-plus\3.5.1\mybatis-plus-3.5.1.jar;C:\Users\mao\.m2\repository\com\baomidou\mybatis-plus-extension\3.5.1\mybatis-plus-extension-3.5.1.jar;C:\Users\mao\.m2\repository\com\baomidou\mybatis-plus-core\3.5.1\mybatis-plus-core-3.5.1.jar;C:\Users\mao\.m2\repository\com\baomidou\mybatis-plus-annotation\3.5.1\mybatis-plus-annotation-3.5.1.jar;C:\Users\mao\.m2\repository\com\github\jsqlparser\jsqlparser\4.3\jsqlparser-4.3.jar;C:\Users\mao\.m2\repository\org\mybatis\mybatis\3.5.9\mybatis-3.5.9.jar;C:\Users\mao\.m2\repository\org\mybatis\mybatis-spring\2.0.6\mybatis-spring-2.0.6.jar;C:\Users\mao\.m2\repository\org\springframework\boot\spring-boot-starter-jdbc\2.7.0\spring-boot-starter-jdbc-

```
\ \ / _-'_ -_- _(-)_ _ _ _ \\ \ \
( ( )\_ | '-' | '-' | '-' \_ _' | \ \ \ \
\ \ _ _ ) | | ) | | | | | | | | ( | | ) ) ) )
' | _ _ | .-| | | | | | | _ , | / / / /
===== | | ===== | _ _ = / / / /
```

:: Spring Boot :: (v2.7.0)

```
2022-06-19 13:51:02.020 INFO 19940 --- [ restartedMain]
m.d.DockerComposeBootApplication : Starting DockerComposeBootApplication
using Java 16.0.2 on mao with PID 19940 (H:\程序\大三下期
\docker_compose_boot\target\classes started by mao in H:\程序\大三下期
\docker_compose_boot)
2022-06-19 13:51:02.021 DEBUG 19940 --- [ restartedMain]
m.d.DockerComposeBootApplication : Running with Spring Boot v2.7.0,
Spring v5.3.20
2022-06-19 13:51:02.022 INFO 19940 --- [ restartedMain]
m.d.DockerComposeBootApplication : The following 1 profile is active:
"dev"
2022-06-19 13:51:02.056 INFO 19940 --- [ restartedMain]
.e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set
'spring.devtools.add-properties' to 'false' to disable
2022-06-19 13:51:02.056 INFO 19940 --- [ restartedMain]
.e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging
consider setting the 'logging.level.web' property to 'DEBUG'
2022-06-19 13:51:02.515 INFO 19940 --- [ restartedMain]
.s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring Data modules found,
entering strict repository configuration mode!
2022-06-19 13:51:02.516 INFO 19940 --- [ restartedMain]
.s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data Redis
repositories in DEFAULT mode.
2022-06-19 13:51:02.532 INFO 19940 --- [ restartedMain]
.s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository
scanning in 6 ms. Found 0 Redis repository interfaces.
2022-06-19 13:51:02.953 INFO 19940 --- [ restartedMain]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080
(http)
2022-06-19 13:51:02.960 INFO 19940 --- [ restartedMain]
o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-06-19 13:51:02.960 INFO 19940 --- [ restartedMain]
org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
Tomcat/9.0.63]
2022-06-19 13:51:03.024 INFO 19940 --- [ restartedMain] o.a.c.c.C.[Tomcat].
[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-06-19 13:51:03.024 INFO 19940 --- [ restartedMain]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
initialization completed in 967 ms
2022-06-19 13:51:03.090 INFO 19940 --- [ restartedMain]
c.a.d.s.b.a.DruidDataSourceAutoConfigure : Init DruidDataSource
2022-06-19 13:51:03.181 INFO 19940 --- [ restartedMain]
com.alibaba.druid.pool.DruidDataSource : {dataSource-1} inited
- - | - - | - - - - | - -
| | \ | / ( | | | \ | / | | | \
/ |
3.5.1
2022-06-19 13:51:03.878 INFO 19940 --- [ restartedMain]
o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port
35729
2022-06-19 13:51:03.913 INFO 19940 --- [ restartedMain]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080
(http) with context path ''
2022-06-19 13:51:03.924 INFO 19940 --- [ restartedMain]
m.d.DockerComposeBootApplication : Started DockerComposeBootApplication
in 2.235 seconds (JVM running for 2.794)
```

```

2022-06-19 13:51:17.541 INFO 19940 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].
[localhost].[/] : Initializing Spring DispatcherServlet
'dispatcherServlet'
2022-06-19 13:51:17.541 INFO 19940 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet : Initializing Servlet
'dispatcherServlet'
2022-06-19 13:51:17.543 INFO 19940 --- [nio-8080-exec-1]
o.s.web.servlet.DispatcherServlet : Completed initialization in 0 ms
2022-06-19 13:51:18.130 DEBUG 19940 --- [nio-8080-exec-1]
m.d.mapper.StudentMapper.selectById : ==> Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student WHERE
student_no=?
2022-06-19 13:51:18.146 DEBUG 19940 --- [nio-8080-exec-1]
m.d.mapper.StudentMapper.selectById : ==> Parameters: 202012340112(Long)
2022-06-19 13:51:18.162 DEBUG 19940 --- [nio-8080-exec-1]
m.d.mapper.StudentMapper.selectById : <== Total: 1
2022-06-19 13:51:37.222 DEBUG 19940 --- [nio-8080-exec-2]
m.d.mapper.StudentMapper.selectById : ==> Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student WHERE
student_no=?
2022-06-19 13:51:37.223 DEBUG 19940 --- [nio-8080-exec-2]
m.d.mapper.StudentMapper.selectById : ==> Parameters: 202012340113(Long)
2022-06-19 13:51:37.225 DEBUG 19940 --- [nio-8080-exec-2]
m.d.mapper.StudentMapper.selectById : <== Total: 1

```

redis和mysql工作正常，代码运行正常

项目地址：

https://github.com/maomao124/Docker_compose_boot.git

更改配置

将配置文件里的127.0.0.1或者localhost更改成docker容器名称

```

#驱动
jdbc.driver=com.mysql.cj.jdbc.Driver
#url
jdbc.url=jdbc:mysql://compose_mysql:3306/test
#用户名
jdbc.username=root
#密码
jdbc.password=20010713

```

```
redis.host=compose_redis
redis.port=6379
redis.password=123456
max-active=8
max-idle=8
max-wait=2000
```

生成jar包

```
mvn package
```

或者

```
mvn package -DskipTests
```

将jar包放入指定目录

我放入的目录是C:\Users\mao\Desktop\test

```
PS C:\Users\mao\Desktop\test> ls
```

目录: C:\Users\mao\Desktop\test

| Mode | LastWriteTime | Length | Name |
|--------|-----------------|----------|--|
| -a---- | 2022/6/19 14:09 | 49238153 | Docker_compose_boot-0.0.1-SNAPSHOT.jar |

```
PS C:\Users\mao\Desktop\test>
```

新建Dockerfile文件

在刚才的目录下建立一个Dockerfile文件，要和jdk的存放目录一致

```
PS C:\Users\mao\Desktop\test> ls
```

目录: C:\Users\mao\Desktop\test

| Mode | LastWriteTime | Length | Name |
|--------|-----------------|----------|--|
| -a---- | 2022/6/19 14:14 | 0 | Dockerfile |
| -a---- | 2022/6/19 14:09 | 49238153 | Docker_compose_boot-0.0.1-SNAPSHOT.jar |

```
PS C:\Users\mao\Desktop\test>
```

编写Dockerfile文件

使用记事本打开，输入以下命令

```
# 基础镜像使用java
FROM java17:1.0
# 作者
MAINTAINER mao
# VOLUME 指定临时文件目录为/tmp，在主机/var/lib/docker目录下创建了一个临时文件并链接到容器的/tmp
VOLUME /tmp
# 将jar包添加到容器中并更名为Docker_compose_boot.jar
ADD Docker_compose_boot-0.0.1-SNAPSHOT.jar Docker_compose_boot.jar
# 运行jar包
RUN bash -c 'touch Docker_compose_boot.jar'
ENTRYPOINT ["java", "-jar", "Docker_compose_boot.jar"]
#暴露端口作为微服务
EXPOSE 8080
```

构建

运行命令:

```
docker build -t docker_compose_boot:1.0 .
```

```
PS C:\Users\mao\Desktop\test> docker build -t docker_compose_boot:1.0 .
[+] Building 1.4s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
```

0.0s


```

=> => transferring dockerfile: 541B
0.0s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [internal] load metadata for docker.io/library/java17:1.0
0.0s
=> [internal] load build context
0.4s
=> => transferring context: 49.25MB
0.4s
=> CACHED [1/3] FROM docker.io/library/java17:1.0
0.0s
=> [2/3] ADD Docker_compose_boot-0.0.1-SNAPSHOT.jar Docker_compose_boot.jar
0.2s
=> [3/3] RUN bash -c 'touch Docker_compose_boot.jar'
0.5s
=> exporting to image
0.2s
=> => exporting layers
0.2s
=> => writing image
sha256:f6b5051647db9b7337febf10aabbcf57c75416745700083306d99b2da27712f4
0.0s
=> => naming to docker.io/library/docker_compose_boot:1.0
0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and
learn how to fix them
PS C:\Users\mao\Desktop\test>

```

查看镜像

docker images

```

PS C:\Users\mao\Desktop\test> docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
docker_compose_boot 1.0          f6b5051647db 48 seconds ago 588MB
docker_boot          1.0          d25f3f4955ae 25 hours ago   525MB
java17               1.0          282982c69086 40 hours ago   489MB
tomcat               latest       c795915cb678 3 weeks ago    680MB
redis                latest       53aa81e8adfa 3 weeks ago    117MB
mysql                latest       65b636d5542b 3 weeks ago    524MB
ubuntu               latest       d2e4e1f51132 7 weeks ago    77.8MB
PS C:\Users\mao\Desktop\test>

```

编写docker-compose.yml文件

```
version: "3"

services:
  boot:
    # 镜像名称
    image: docker_compose_boot:1.0
    # 生成后的容器名称
    container_name: docker_compose_boot1
    #映射的端口号
    ports:
      - "8080:8080"
    #容器数据卷
    #volumes:
      # - H:/Docker/docker_compose_boot1/:/usr/local/
    #使用的网络
    networks:
      - mao_net
    #依赖
    depends_on:
      - redis
      - mysql

  redis:
    # 镜像名称
    image: redis
    # 生成后的容器名称
    container_name: compose_redis
    #映射的端口号
    ports:
      - "6383:6379"
    #容器数据卷
    volumes:
      - H:/Docker/docker_compose_redis/conf/redis.conf:/etc/redis/redis.conf
      - H:/Docker/docker_compose_redis/data:/data
    #使用的网络
    networks:
      - mao_net
    #命令
    command: redis-server /etc/redis/redis.conf

  mysql:
    # 镜像名称
    image: mysql
    # 生成后的容器名称
    container_name: compose_mysql
    # 环境配置
    environment:
    #密码
      MYSQL_ROOT_PASSWORD: '123456'
```

```

    MYSQL_DATABASE: 'test'
# 端口
ports:
  - "3310:3306"
#容器数据卷
volumes:
  - H:/Docker/docker_compose_mysql/data:/var/lib/mysql
  - H:/Docker/docker_compose_mysql/conf/:/etc/mysql/conf.d
  - H:/Docker/docker_compose_mysql/log/:/var/log/mysql
# 使用的网络
networks:
  - mao_net

#command: --default-authentication-plugin=mysql_native_password #解决外部无法访问

#网络
networks:
  mao_net:

```

需要redis配置文件

验证compose文件

```

PS C:\Users\mao\Desktop\test> docker-compose config
networks:
  mao_net: {}
services:
  boot:
    container_name: docker_compose_boot1
    depends_on:
      mysql:
        condition: service_started
      redis:
        condition: service_started
    image: docker_compose_boot:1.0
    networks:
      mao_net: null
    ports:
      - published: 8080
        target: 8080
  mysql:
    container_name: compose_mysql
    environment:
      MYSQL_DATABASE: test
      MYSQL_ROOT_PASSWORD: '123456'
    image: mysql
    networks:
      mao_net: null
    ports:

```

```

- published: 3310
  target: 3306
volumes:
- H:\Docker\docker_compose_mysql\data:/var/lib/mysql:rw
- H:\Docker\docker_compose_mysql\conf:/etc/mysql/conf.d:rw
- H:\Docker\docker_compose_mysql\log:/var/log/mysql:rw
redis:
  command: redis-server /etc/redis/redis.conf
  container_name: compose_redis
  image: redis
  networks:
    mao_net: null
  ports:
    - published: 6383
      target: 6379
  volumes:
    - H:\Docker\docker_compose_redis\conf\redis.conf:/etc/redis/redis.conf:rw
    - H:\Docker\docker_compose_redis\data:/data:rw
version: '3'

PS C:\Users\mao\Desktop\test>

```

启动

```
docker-compose up
```

或者

```
docker-compose up -d
```

```

PS C:\Users\mao\Desktop\test> docker-compose up
Creating compose_redis ... done
Creating compose_mysql ... done
Creating docker_compose_boot1 ... done
Attaching to compose_redis, compose_mysql, docker_compose_boot1
compose_mysql | 2022-06-19 07:03:25+00:00 [Note] [Entrypoint]: Entrypoint script
for MySQL Server 8.0.29-1debian10 started.
compose_mysql | 2022-06-19 07:03:26+00:00 [Note] [Entrypoint]: Switching to
dedicated user 'mysql'
compose_mysql | 2022-06-19 07:03:26+00:00 [Note] [Entrypoint]: Entrypoint script
for MySQL Server 8.0.29-1debian10 started.
compose_mysql | 2022-06-19T07:03:26.397467Z 0 [Warning] [MY-010918] [Server]
'default_authentication_plugin' is deprecated and will be removed in a future
release. Please use authentication_policy instead.
compose_mysql | 2022-06-19T07:03:26.397492Z 0 [System] [MY-010116] [Server]
/usr/sbin/mysqld (mysqld 8.0.29) starting as process 1

```

```

compose_mysql | 2022-06-19T07:03:26.402571Z 0 [Warning] [MY-010159] [Server]
Setting lower_case_table_names=2 because file system for /var/lib/mysql/ is case
insensitive
compose_mysql | 2022-06-19T07:03:26.431945Z 1 [System] [MY-013576] [InnoDB]
InnoDB initialization has started.
compose_redis | 1:C 19 Jun 2022 07:03:25.874 # oO00oO00oO00o Redis is starting
oO00oO00oO00o
compose_redis | 1:C 19 Jun 2022 07:03:25.874 # Redis version=7.0.0, bits=64,
commit=00000000, modified=0, pid=1, just started
compose_redis | 1:C 19 Jun 2022 07:03:25.874 # Configuration loaded
compose_redis | 1:M 19 Jun 2022 07:03:25.875 * monotonic clock: POSIX
clock_gettime
compose_redis |
compose_redis |
compose_redis | Redis 7.0.0 (00000000/0)
64 bit
compose_redis |
compose_redis | (
mode
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | |
compose_redis | 1:M 19 Jun 2022 07:03:25.875 # Server initialized
compose_redis | 1:M 19 Jun 2022 07:03:25.875 # WARNING overcommit_memory is set
to 0! Background save may fail under low memory condition. To fix this issue add
'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command
'sysctl vm.overcommit_memory=1' for this to take effect.
compose_redis | 1:M 19 Jun 2022 07:03:25.876 * The AOF directory appendonlydir
doesn't exist
compose_redis | 1:M 19 Jun 2022 07:03:25.878 * Loading RDB produced by version
7.0.0
compose_redis | 1:M 19 Jun 2022 07:03:25.878 * RDB age 647 seconds
compose_redis | 1:M 19 Jun 2022 07:03:25.878 * RDB memory usage when created
0.82 Mb
compose_redis | 1:M 19 Jun 2022 07:03:25.878 * Done loading RDB, keys loaded: 1,
keys expired: 0.
compose_redis | 1:M 19 Jun 2022 07:03:25.878 * DB loaded from disk: 0.002
seconds
compose_redis | 1:M 19 Jun 2022 07:03:25.878 * Ready to accept connections
docker_compose_boot1 |
docker_compose_boot1 |
docker_compose_boot1 | /\ / _' _ _ _ ( ) _ _ _ \ \ \ \
docker_compose_boot1 | ( ( ) \ _ | ' _ | ' _ | ' _ \ _ | \ \ \ \
docker_compose_boot1 | \ \ \ _ _ ) | | | | | | | | ( | | ) ) ) )
docker_compose_boot1 | ' | _ _ | . _ | | | | | | | | \ _ , | / / / /
docker_compose_boot1 | ===== | _ | ===== | _ _ / = / _ / _ /
docker_compose_boot1 | :: Spring Boot :: (v2.7.0)
docker_compose_boot1 |

```

```

docker_compose_boot1 | 2022-06-19 07:03:27.670 INFO 1 --- [          main]
m.d.DockerComposeBootApplication : Starting DockerComposeBootApplication
v0.0.1-SNAPSHOT using Java 17.0.3.1 on 2d0bb81733e1 with PID 1
(/usr/local/Docker_compose_boot.jar started by root in /usr/local)
docker_compose_boot1 | 2022-06-19 07:03:27.672 DEBUG 1 --- [          main]
m.d.DockerComposeBootApplication : Running with Spring Boot v2.7.0,
Spring v5.3.20
docker_compose_boot1 | 2022-06-19 07:03:27.673 INFO 1 --- [          main]
m.d.DockerComposeBootApplication : The following 1 profile is active:
"dev"
compose_mysql | 2022-06-19T07:03:27.849590Z 1 [System] [MY-013577] [InnoDB]
InnoDB initialization has ended.
compose_mysql | 2022-06-19T07:03:28.147059Z 0 [System] [MY-010229] [Server]
Starting XA crash recovery...
compose_mysql | 2022-06-19T07:03:28.156001Z 0 [System] [MY-010232] [Server] XA
crash recovery finished.
compose_mysql | 2022-06-19T07:03:28.379536Z 0 [Warning] [MY-010068] [Server] CA
certificate ca.pem is self signed.
compose_mysql | 2022-06-19T07:03:28.379949Z 0 [System] [MY-013602] [Server]
Channel mysql_main configured to support TLS. Encrypted connections are now
supported for this channel.
compose_mysql | 2022-06-19T07:03:28.390610Z 0 [Warning] [MY-011810] [Server]
Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is
accessible to all OS users. Consider choosing a different directory.
compose_mysql | 2022-06-19T07:03:28.433725Z 0 [System] [MY-011323] [Server] X
Plugin ready for connections. Bind-address: '::' port: 33060, socket:
/var/run/mysqld/mysqld.sock
compose_mysql | 2022-06-19T07:03:28.433910Z 0 [System] [MY-010931] [Server]
/usr/sbin/mysqld: ready for connections. Version: '8.0.29' socket:
'/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
docker_compose_boot1 | 2022-06-19 07:03:28.461 INFO 1 --- [          main]
.s.d.r.c.RepositoryConfigurationDelegate : Multiple Spring Data modules found,
entering strict repository configuration mode!
docker_compose_boot1 | 2022-06-19 07:03:28.464 INFO 1 --- [          main]
.s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data Redis
repositories in DEFAULT mode.
docker_compose_boot1 | 2022-06-19 07:03:28.484 INFO 1 --- [          main]
.s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository
scanning in 7 ms. Found 0 Redis repository interfaces.
docker_compose_boot1 | 2022-06-19 07:03:29.151 INFO 1 --- [          main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080
(http)
docker_compose_boot1 | 2022-06-19 07:03:29.160 INFO 1 --- [          main]
o.apache.catalina.core.StandardService : Starting service [Tomcat]
docker_compose_boot1 | 2022-06-19 07:03:29.160 INFO 1 --- [          main]
org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
Tomcat/9.0.63]
docker_compose_boot1 | 2022-06-19 07:03:29.220 INFO 1 --- [          main]
o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded
webApplicationContext
docker_compose_boot1 | 2022-06-19 07:03:29.221 INFO 1 --- [          main]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
initialization completed in 1483 ms
docker_compose_boot1 | 2022-06-19 07:03:29.322 INFO 1 --- [          main]
c.a.d.s.b.a.DruidDataSourceAutoConfigure : Init DruidDataSource
docker_compose_boot1 | 2022-06-19 07:03:29.539 INFO 1 --- [          main]
com.alibaba.druid.pool.DruidDataSource : {dataSource-1} inited
docker_compose_boot1 | _ _ | _ _ _ | _ _ _ _ | _ _

```

```

docker_compose_boot1 | | | | \ | _ ( _ | | | _ \ | _ ) | | _ | _ \
docker_compose_boot1 | | | | / | | | | | | | | | | | | | | | |
docker_compose_boot1 | | | | | | | | | | | | | | | | | | | | | | 3.5.1
docker_compose_boot1 | 2022-06-19 07:03:30.635 INFO 1 --- [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080
(http) with context path ''
docker_compose_boot1 | 2022-06-19 07:03:30.649 INFO 1 --- [           main]
m.d.DockerComposeBootApplication : Started DockerComposeBootApplication
in 3.65 seconds (JVM running for 4.144)

```

访问mysql

```
mysql -u root -P 3310 -p
```

密码为123456

使用test数据库

建表

```

DROP TABLE IF EXISTS `student`;
CREATE TABLE `student`
(
    `student_no`          bigint
    NOT NULL AUTO_INCREMENT COMMENT '学生学号',
    `name`                varchar(12) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '学生姓名',
    `sex`                 varchar(2) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '学生性别',
    `telephone_number`    varchar(15) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '电话号码',
    `family_telephone_number` varchar(15) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '家庭电话号码',
    `birthday`            varchar(12) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '生日信息，由程序填充，这里用varchar型',
    `address`             varchar(60) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '家庭地址',
    `id_card`             varchar(20) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '身份证号码',
    `email`               varchar(40) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '电子邮箱',
    `dormitory_number`    varchar(20) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '寝室号',
    `class_no`            bigint
    NOT NULL COMMENT '所属班级编号',
    `state`               varchar(4) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '状态，在读、退学或者毕业',

```

```

        `remarks`          varchar(255) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '备注',
        PRIMARY KEY (`student_no`) USING BTREE,
        UNIQUE INDEX `no` (`student_no`) USING BTREE,
        INDEX `class_no` (`class_no`) USING BTREE
    ) ENGINE = InnoDB
    AUTO_INCREMENT = 202012341078
    CHARACTER SET = utf8
    COLLATE = utf8_general_ci
    ROW_FORMAT = DYNAMIC;

```

插入数据

```

INSERT INTO `student`
VALUES (202012340101, '关亮', '男', '13004194935', '15305295308', '2000-04-21',
'澳门三路107号-9-5', '439419200004213162',
'ev0y07@aol.com', '12栋401', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340102, '幸茗羽', '女', '15508840091', '15303468961', '2000-11-28',
'武胜关广场37号-4-9', '422461200011287472',
'x7bqfpj@263.net', '1栋402', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340103, '五中言', '男', '13400211648', '13403154851', '2001-02-08',
'东光大厦97号-14-1', '422365200102086997',
'jlg05zq82@msn.com', '12栋605', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340104, '邱月', '女', '15002510556', '15900573259', '2003-06-08',
'韶关路12号-14-3', '432562200306087287',
'0fw8g1h19@yahoo.com', '5栋121', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340105, '寿泽敬', '男', '15205711313', '15207565368', '2000-03-23',
'武城广场63号-11-9', '425459200003233592',
'pvzem7@163.net', '11栋421', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340106, '明壮冠', '男', '13006938844', '15203703744', '2001-08-09',
'济南街66号-7-4', '432821200108097814',
'waygeq@live.com', '8栋614', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340107, '利慧瑞', '女', '13105082904', '13905854909', '2003-01-23',
'朝阳街82号-20-6', '427321200301237331',
's68he1w@gmail.com', '2栋406', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340108, '良爽婵', '女', '13301845205', '13208194646', '2000-06-23',
'咸阳支街88号-20-4', '435758200006233465',
'sqryex@live.com', '2栋219', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340109, '阳婉菊', '女', '15800602721', '13404982437', '2001-04-02',
'禹城街90号-4-9', '438892200104024530',
'giwjmp@sina.com', '5栋216', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340110, '璩璧', '女', '13600652524', '15505265367', '2000-11-05',
'黄台广场122号-6-9', '430529200011053854',
'pb72ppog@126.com', '1栋111', 1001, '在读', '');
INSERT INTO `student`

```



```
VALUES (202012340111, '洪成健', '男', '13100250691', '13300574325', '2000-02-17',
'大港沿147号-4-3', '429550200002175613',
'bslct5v@0355.net', '7栋523', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340112, '阳颖', '女', '15306118239', '13600833930', '2002-01-12',
'嘉祥路84号-8-10', '432213200201121560',
'cb0m4x0j@sina.com', '3栋312', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340113, '夔琳', '女', '15301204581', '13404558326', '2001-01-19',
'延安一街143号-12-10', '441050200101198751',
'hogajnhj@0355.net', '4栋514', 1001, '在读', '');
INSERT INTO `student`
VALUES (202012340114, '蒋馨凝', '女', '13705930578', '13508264247', '2003-04-01',
'信号山支路120号-9-7', '437998200304013069',
'ddj48omr@263.net', '2栋221', 1001, '在读', '');
```

```
C:\Users\mao>mysql -u root -P 3310 -p
```

```
Enter password: *****
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 8
```

```
Server version: 8.0.29 MySQL Community Server - GPL
```

```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> show databases
```

```
-> ;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
+-----+
5 rows in set (0.00 sec)
```

```
mysql>
```

```
mysql> use test
```

```
Database changed
```

```
mysql> DROP TABLE IF EXISTS `student`;
```

```
Query OK, 0 rows affected, 1 warning (0.04 sec)
```

```
mysql> CREATE TABLE `student`
```

```
-> (
```

```
->     `student_no`          bigint
      NOT NULL AUTO_INCREMENT COMMENT '学生学号',
```

```
->     `name`                varchar(12) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '学生姓名',
```

```

->      `sex`                varchar(2) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '学生性别',
->      `telephone_number`    varchar(15) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '电话号码',
->      `family_telephone_number` varchar(15) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '家庭电话号码',
->      `birthday`            varchar(12) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '生日信息, 由程序填充, 这里用varchar型',
->      `address`             varchar(60) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '家庭地址',
->      `id_card`             varchar(20) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT '身份证号码',
->      `email`               varchar(40) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '电子邮箱',
->      `dormitory_number`    varchar(20) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '寝室号',
->      `class_no`            bigint
NOT NULL COMMENT '所属班级编号',
->      `state`               varchar(4) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '状态, 在读、退学或者毕业',
->      `remarks`             varchar(255) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL COMMENT '备注',
->      PRIMARY KEY (`student_no`) USING BTREE,
->      UNIQUE INDEX `no` (`student_no`) USING BTREE,
->      INDEX `class_no` (`class_no`) USING BTREE
-> ) ENGINE = InnoDB
->   AUTO_INCREMENT = 202012341078
->   CHARACTER SET = utf8
->   COLLATE = utf8_general_ci
->   ROW_FORMAT = DYNAMIC;

```

Query OK, 0 rows affected, 24 warnings (0.53 sec)

mysql> INSERT INTO `student`

```

-> VALUES (202012340101, '关亮', '男', '13004194935', '15305295308', '2000-04-
21', '澳门三路107号-9-5', '439419200004213162',
->      'ev0y07@aol.com', '12栋401', 1001, '在读', '');

```

Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO `student`

```

-> VALUES (202012340102, '幸茗羽', '女', '15508840091', '15303468961', '2000-
11-28', '武胜关广场37号-4-9', '422461200011287472',
->      'x7bqfpj@263.net', '1栋402', 1001, '在读', '');

```

Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `student`

```

-> VALUES (202012340103, '五中言', '男', '13400211648', '13403154851', '2001-
02-08', '东光大厦97号-14-1', '422365200102086997',
->      'jlg05zq82@msn.com', '12栋605', 1001, '在读', '');

```

Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO `student`

```

-> VALUES (202012340104, '邱月', '女', '15002510556', '15900573259', '2003-06-
08', '韶关路12号-14-3', '432562200306087287',
->      '0fw8g1h19@yahoo.com', '5栋121', 1001, '在读', '');

```

Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO `student`

```

-> VALUES (202012340105, '寿泽敬', '男', '15205711313', '15207565368', '2000-03-23', '武城广场63号-11-9', '425459200003233592',
->         'pvzem7@163.net', '11栋421', 1001, '在读', '');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340106, '明壮冠', '男', '13006938844', '15203703744', '2001-08-09', '济南街66号-7-4', '432821200108097814',
->         'waygeq@live.com', '8栋614', 1001, '在读', '');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340107, '利慧瑞', '女', '13105082904', '13905854909', '2003-01-23', '朝阳街82号-20-6', '427321200301237331',
->         's68he1w@gmail.com', '2栋406', 1001, '在读', '');
Query OK, 1 row affected (0.24 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340108, '良爽婵', '女', '13301845205', '13208194646', '2000-06-23', '咸阳支街88号-20-4', '435758200006233465',
->         'sqryex@live.com', '2栋219', 1001, '在读', '');
Query OK, 1 row affected (0.14 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340109, '阳婉菊', '女', '15800602721', '13404982437', '2001-04-02', '禹城街90号-4-9', '438892200104024530',
->         'giwjmp@sina.com', '5栋216', 1001, '在读', '');
Query OK, 1 row affected (0.14 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340110, '璩璧', '女', '13600652524', '15505265367', '2000-11-05', '黄台广场122号-6-9', '430529200011053854',
->         'pb72ppog@126.com', '1栋111', 1001, '在读', '');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340111, '洪成健', '男', '13100250691', '13300574325', '2000-02-17', '大港沿147号-4-3', '429550200002175613',
->         'bslct5v@0355.net', '7栋523', 1001, '在读', '');
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340112, '阳颖', '女', '15306118239', '13600833930', '2002-01-12', '嘉祥路84号-8-10', '432213200201121560',
->         'cb0m4x0j@sina.com', '3栋312', 1001, '在读', '');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340113, '夔琳', '女', '15301204581', '13404558326', '2001-01-19', '延安一街143号-12-10', '441050200101198751',
->         'hogajnh@0355.net', '4栋514', 1001, '在读', '');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO `student`
-> VALUES (202012340114, '蒋馨凝', '女', '13705930578', '13508264247', '2003-04-01', '信号山支路120号-9-7', '437998200304013069',
->         'ddj48omr@263.net', '2栋221', 1001, '在读', '');
Query OK, 1 row affected (0.04 sec)

```

```
mysql> select * from student
```

```
-> ;
```

| student_no | name | sex | telephone_number | family_telephone_number | birthday | address | id_card | email | dormitory_number | class_no | state | remarks |
|--------------|------|-----|------------------|-------------------------|------------|----------------|--------------------|---------------------|------------------|----------|-------|---------|
| 202012340101 | 关亮 | 男 | 13004194935 | 15305295308 | 2000-04-21 | 澳门三路107号-9-5 | 439419200004213162 | ev0y07@ao1.com | 12栋401 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340102 | 幸茗羽 | 女 | 15508840091 | 15303468961 | 2000-11-28 | 武胜关广场37号-4-9 | 422461200011287472 | x7bqfpj@263.net | 1栋402 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340103 | 五中言 | 男 | 13400211648 | 13403154851 | 2001-02-08 | 东光大厦97号-14-1 | 422365200102086997 | j1g05zq82@msn.com | 12栋605 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340104 | 邱月 | 女 | 15002510556 | 15900573259 | 2003-06-08 | 韶关路12号-14-3 | 432562200306087287 | 0fw8g1h19@yahoo.com | 5栋121 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340105 | 寿泽敬 | 男 | 15205711313 | 15207565368 | 2000-03-23 | 武城广场63号-11-9 | 425459200003233592 | pvzem7@163.net | 11栋421 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340106 | 明壮冠 | 男 | 13006938844 | 15203703744 | 2001-08-09 | 济南街66号-7-4 | 432821200108097814 | waygeq@live.com | 8栋614 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340107 | 利慧瑞 | 女 | 13105082904 | 13905854909 | 2003-01-23 | 朝阳街82号-20-6 | 427321200301237331 | s68he1w@gmail.com | 2栋406 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340108 | 良爽婵 | 女 | 13301845205 | 13208194646 | 2000-06-23 | 咸阳支街88号-20-4 | 435758200006233465 | sqryex@live.com | 2栋219 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340109 | 阳婉菊 | 女 | 15800602721 | 13404982437 | 2001-04-02 | 禹城街90号-4-9 | 438892200104024530 | giwjmp@sina.com | 5栋216 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340110 | 璩璧 | 女 | 13600652524 | 15505265367 | 2000-11-05 | 黄台广场122号-6-9 | 430529200011053854 | pb72ppog@126.com | 1栋111 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340111 | 洪成健 | 男 | 13100250691 | 13300574325 | 2000-02-17 | 大港沿147号-4-3 | 429550200002175613 | bslct5v@0355.net | 7栋523 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340112 | 阳颖 | 女 | 15306118239 | 13600833930 | 2002-01-12 | 嘉祥路84号-8-10 | 432213200201121560 | cb0m4x0j@sina.com | 3栋312 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340113 | 夔琳 | 女 | 15301204581 | 13404558326 | 2001-01-19 | 延安一街143号-12-10 | 441050200101198751 | hogajnhhi@0355.net | 4栋514 | | | |
| | 1001 | 在读 | | | | | | | | | | |
| 202012340114 | 蒋馨凝 | 女 | 13705930578 | 13508264247 | 2003-04-01 | 信号山支路120号-9-7 | 437998200304013069 | ddj48omr@263.net | 2栋221 | | | |
| | 1001 | 在读 | | | | | | | | | | |

```
14 rows in set (0.00 sec)
```

```
mysql>
```

访问服务

<http://localhost:8080/student/>

<http://localhost:8080/student/202012340101>

<http://localhost:8080/student/202012340102>

```
docker_compose_boot1 | 2022-06-19 10:40:39.568 WARN 1 --- [nio-8080-exec-4]
c.a.druid.pool.DruidAbstractDataSource : discard long time none received
connection. , jdbcUrl : jdbc:mysql://compose_mysql:3306/test, version : 1.2.8,
lastPacketReceivedIdleMillis : 81682
docker_compose_boot1 | 2022-06-19 10:40:39.589 DEBUG 1 --- [nio-8080-exec-4]
m.d.mapper.StudentMapper.selectList      : ==> Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student
docker_compose_boot1 | 2022-06-19 10:40:39.590 DEBUG 1 --- [nio-8080-exec-4]
m.d.mapper.StudentMapper.selectList      : ==> Parameters:
docker_compose_boot1 | 2022-06-19 10:40:39.617 DEBUG 1 --- [nio-8080-exec-4]
m.d.mapper.StudentMapper.selectList      : <==      Total: 14
docker_compose_boot1 | 2022-06-19 10:40:40.453 DEBUG 1 --- [nio-8080-exec-5]
m.d.mapper.StudentMapper.selectList      : ==> Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student
docker_compose_boot1 | 2022-06-19 10:40:40.454 DEBUG 1 --- [nio-8080-exec-5]
m.d.mapper.StudentMapper.selectList      : ==> Parameters:
docker_compose_boot1 | 2022-06-19 10:40:40.463 DEBUG 1 --- [nio-8080-exec-5]
m.d.mapper.StudentMapper.selectList      : <==      Total: 14
docker_compose_boot1 | 2022-06-19 10:40:40.887 DEBUG 1 --- [nio-8080-exec-6]
m.d.mapper.StudentMapper.selectList      : ==> Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student
docker_compose_boot1 | 2022-06-19 10:40:40.888 DEBUG 1 --- [nio-8080-exec-6]
m.d.mapper.StudentMapper.selectList      : ==> Parameters:
docker_compose_boot1 | 2022-06-19 10:40:40.892 DEBUG 1 --- [nio-8080-exec-6]
m.d.mapper.StudentMapper.selectList      : <==      Total: 14
docker_compose_boot1 | 2022-06-19 10:40:41.077 DEBUG 1 --- [nio-8080-exec-7]
m.d.mapper.StudentMapper.selectList      : ==> Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student
docker_compose_boot1 | 2022-06-19 10:40:41.078 DEBUG 1 --- [nio-8080-exec-7]
m.d.mapper.StudentMapper.selectList      : ==> Parameters:
docker_compose_boot1 | 2022-06-19 10:40:41.082 DEBUG 1 --- [nio-8080-exec-7]
m.d.mapper.StudentMapper.selectList      : <==      Total: 14
docker_compose_boot1 | 2022-06-19 10:41:36.967 DEBUG 1 --- [nio-8080-exec-8]
m.d.mapper.StudentMapper.selectList      : ==> Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student
```

```

docker_compose_boot1 | 2022-06-19 10:41:36.968 DEBUG 1 --- [nio-8080-exec-8]
m.d.mapper.StudentMapper.selectList      : ==> Parameters:
docker_compose_boot1 | 2022-06-19 10:41:36.971 DEBUG 1 --- [nio-8080-exec-8]
m.d.mapper.StudentMapper.selectList      : <==      Total: 14
docker_compose_boot1 | 2022-06-19 10:41:37.507 DEBUG 1 --- [nio-8080-exec-9]
m.d.mapper.StudentMapper.selectList      : ==>   Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student
docker_compose_boot1 | 2022-06-19 10:41:37.508 DEBUG 1 --- [nio-8080-exec-9]
m.d.mapper.StudentMapper.selectList      : ==> Parameters:
docker_compose_boot1 | 2022-06-19 10:41:37.511 DEBUG 1 --- [nio-8080-exec-9]
m.d.mapper.StudentMapper.selectList      : <==      Total: 14
docker_compose_boot1 | 2022-06-19 10:41:37.718 DEBUG 1 --- [io-8080-exec-10]
m.d.mapper.StudentMapper.selectList      : ==>   Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student
docker_compose_boot1 | 2022-06-19 10:41:37.719 DEBUG 1 --- [io-8080-exec-10]
m.d.mapper.StudentMapper.selectList      : ==> Parameters:
docker_compose_boot1 | 2022-06-19 10:41:37.722 DEBUG 1 --- [io-8080-exec-10]
m.d.mapper.StudentMapper.selectList      : <==      Total: 14
docker_compose_boot1 | 2022-06-19 10:41:48.618 DEBUG 1 --- [nio-8080-exec-2]
m.d.mapper.StudentMapper.selectById      : ==>   Preparing: SELECT
student_no,name,sex,telephone_number,family_telephone_number,birthday,address,id
_card,email,dormitory_number,class_no,state,remarks FROM student WHERE
student_no=?
docker_compose_boot1 | 2022-06-19 10:41:48.618 DEBUG 1 --- [nio-8080-exec-2]
m.d.mapper.StudentMapper.selectById      : ==> Parameters: 202012340101(Long)
docker_compose_boot1 | 2022-06-19 10:41:48.622 DEBUG 1 --- [nio-8080-exec-2]
m.d.mapper.StudentMapper.selectById      : <==      Total: 1

```

```

C:\Users\mao>redis-cli -p 6383
127.0.0.1:6383> ping
PONG
127.0.0.1:6383> keys *
1) "docker_compose_boot:student:202012340101"
2) "docker_compose_boot:student:202012340102"
3) "a"
127.0.0.1:6383>

```

服务正常运行

portainer

是什么

Portainer 是一款轻量级的应用，它提供了图形化界面，用于方便地管理 Docker 环境，包括单机环境和集群环境。

安装

```
docker run -d -p 8000:8000 -p 9000:9000 --name portainer --restart=always -v \\.\pipe\docker_engine:\\.\pipe\docker_engine -v H:/Docker/portainer/data:/data portainer/portainer
```

```
docker run -d -p 8000:8000 -p 9000:9000 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer
```

访问

<http://localhost:9000/>

CAdvisor+InfluxDB+Granfana

CAdvisor监控收集+InfluxDB存储数据+Granfana展示图表

docker-compose.yml

```
version: '3.1'

volumes:
  grafana_data: {}

services:
  influxdb:
    image: tutum/influxdb:0.9
    restart: always
    environment:
      - PRE_CREATE_DB=cadvisor
    ports:
      - "8083:8083"
      - "8086:8086"
    volumes:
      - H:\Docker\influxdb:/data

  cadvisor:
```

```

image: google/cadvisor
links:
  - influxdb:influxsrv
command: -storage_driver=influxdb -storage_driver_db=cadvisor -
storage_driver_host=influxsrv:8086
restart: always
ports:
  - "8080:8080"
#volumes:
  #- H:\Docker\cadvisor:/rootfs:ro
  #- H:\Docker\cadvisor/run:/var/run:rw
  #- H:\Docker\cadvisor/sys:/sys:ro
  #- H:\Docker\cadvisor/docker/./var/lib/docker:ro

grafana:
user: "104"
image: grafana/grafana
user: "104"
restart: always
links:
  - influxdb:influxsrv
ports:
  - "3000:3000"
volumes:
  - H:\Docker\grafana:/var/lib/grafana
environment:
  - HTTP_USER=admin
  - HTTP_PASS=admin
  - INFLUXDB_HOST=influxsrv
  - INFLUXDB_PORT=8086
  - INFLUXDB_NAME=cadvisor
  - INFLUXDB_USER=root
  - INFLUXDB_PASS=root

```

```

PS C:\Users\mao\Desktop> docker-compose up
Creating network "desktop_default" with the default driver
Creating desktop_influxdb_1 ... done
Creating desktop_cadvisor_1 ... done
Creating desktop_grafana_1 ... done
Attaching to desktop_influxdb_1, desktop_grafana_1, desktop_cadvisor_1
influxdb_1 | => The initialization script had been executed before, skipping
...
influxdb_1 | => Starting InfluxDB in foreground ...
influxdb_1 |
influxdb_1 | 8888888      .d888 888      8888888b. 888888b.
influxdb_1 | 888      d88P" 888      888  "Y88b 888
"88b
influxdb_1 | 888      888 888      888      888 888
.88P
influxdb_1 | 888 888888b. 888888 888 888 888 888 888 888
88888888K.

```



```

influxdb_1 | 888 888 "88b 888 888 888 888 Y8bd8P' 888 888 888
"Y88b
influxdb_1 | 888 888 888 888 888 888 888 x88K 888 888 888
888
influxdb_1 | 888 888 888 888 888 Y88b 888 .d8""8b. 888 .d88P 888
d88P
influxdb_1 | 8888888 888 888 888 888 "Y88888 888 888 8888888P"
88888888P"
influxdb_1 |
influxdb_1 | 2022/06/19 12:01:42 InfluxDB starting, version 0.9.6.1, branch
0.9.6, commit 6d3a8603cfdaf1a141779ed88b093dcc5c528e5e, built 2015-12-
10T23:40:23+0000
influxdb_1 | 2022/06/19 12:01:42 Go version go1.4.2, GOMAXPROCS set to 16
influxdb_1 | 2022/06/19 12:01:42 Using configuration at: /config/config.toml
influxdb_1 | [metastore] 2022/06/19 12:01:42 Using data dir: /data/meta
influxdb_1 | [metastore] 2022/06/19 12:01:43 Node at localhost:8088 [Follower]
influxdb_1 | [metastore] 2022/06/19 12:01:43 Skipping cluster join: already
member of cluster: nodeId=1 raftEnabled=true peers=[localhost:8088]
cadvisor_1 | E0619 12:01:44.530155 1 info.go:140] Failed to get system
UUID: open /etc/machine-id: no such file or directory
cadvisor_1 | W0619 12:01:44.530231 1 info.go:53] Couldn't collect info
from any of the files in "/etc/machine-id,/var/lib/dbus/machine-id"
cadvisor_1 | W0619 12:01:44.533160 1 manager.go:349] Could not configure
a source for OOM detection, disabling OOM events: open /dev/kmsg: no such file
or directory
grafana_1 | Error checking server process execution privilege. error: could not
get current OS user to detect process privileges
grafana_1 | logger=settings t=2022-06-19T12:01:44.60664344Z level=info
msg="Starting Grafana" version=9.0.0 commit=b5c56f6371 branch=HEAD
compiled=2022-06-13T12:06:48Z
grafana_1 | logger=settings t=2022-06-19T12:01:44.606814907Z level=info
msg="Config loaded from" file=/usr/share/grafana/conf/defaults.ini
grafana_1 | logger=settings t=2022-06-19T12:01:44.606845927Z level=info
msg="Config loaded from" file=/etc/grafana/grafana.ini
grafana_1 | logger=settings t=2022-06-19T12:01:44.606852886Z level=info
msg="Config overridden from command line"
arg="default.paths.data=/var/lib/grafana"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606857126Z level=info
msg="Config overridden from command line"
arg="default.paths.logs=/var/log/grafana"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606860616Z level=info
msg="Config overridden from command line"
arg="default.paths.plugins=/var/lib/grafana/plugins"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606863766Z level=info
msg="Config overridden from command line"
arg="default.paths.provisioning=/etc/grafana/provisioning"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606866896Z level=info
msg="Config overridden from command line" arg="default.log.mode=console"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606870246Z level=info
msg="Config overridden from Environment variable"
var="GF_PATHS_DATA=/var/lib/grafana"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606873506Z level=info
msg="Config overridden from Environment variable"
var="GF_PATHS_LOGS=/var/log/grafana"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606876846Z level=info
msg="Config overridden from Environment variable"
var="GF_PATHS_PLUGINS=/var/lib/grafana/plugins"

```

```
grafana_1 | logger=settings t=2022-06-19T12:01:44.606880666Z level=info
msg="Config overridden from Environment variable"
var="GF_PATHS_PROVISIONING=/etc/grafana/provisioning"
grafana_1 | logger=settings t=2022-06-19T12:01:44.606884216Z level=info
msg="Path Home" path=/usr/share/grafana
grafana_1 | logger=settings t=2022-06-19T12:01:44.606887446Z level=info
msg="Path Data" path=/var/lib/grafana
grafana_1 | logger=settings t=2022-06-19T12:01:44.606890946Z level=info
msg="Path Logs" path=/var/log/grafana
grafana_1 | logger=settings t=2022-06-19T12:01:44.606895156Z level=info
msg="Path Plugins" path=/var/lib/grafana/plugins
grafana_1 | logger=settings t=2022-06-19T12:01:44.606898586Z level=info
msg="Path Provisioning" path=/etc/grafana/provisioning
grafana_1 | logger=settings t=2022-06-19T12:01:44.606902416Z level=info
msg="App mode production"
grafana_1 | logger=sqlstore t=2022-06-19T12:01:44.607202306Z level=info
msg="Connecting to DB" dbtype=sqlite3
grafana_1 | logger=migrator t=2022-06-19T12:01:44.638747149Z level=info
msg="Starting DB migrations"
grafana_1 | logger=migrator t=2022-06-19T12:01:44.646891742Z level=info
msg="migrations completed" performed=0 skipped=422 duration=475.982µs
grafana_1 | logger=plugin.manager t=2022-06-19T12:01:44.674620798Z level=info
msg="Plugin registered" pluginId=input
grafana_1 | logger=secrets t=2022-06-19T12:01:44.678725671Z level=info
msg="Envelope encryption state" enabled=true currentprovider=secretKey.v1
grafana_1 | logger=query_data t=2022-06-19T12:01:44.682881351Z level=info
msg="Query Service initialization"
grafana_1 | logger=live.push_http t=2022-06-19T12:01:44.686075391Z level=info
msg="Live Push Gateway initialization"
influxdb_1 | [metastore] 2022/06/19 12:01:44 Node at localhost:8088 [Leader].
peers=[localhost:8088]
grafana_1 | logger=infra.usagestats.collector t=2022-06-19T12:01:44.825511242Z
level=info msg="registering usage stat providers" usageStatsProvidersLen=2
grafana_1 | logger=ngalert t=2022-06-19T12:01:44.825912226Z level=info
msg="warming cache for startup"
grafana_1 | logger=grafanaStorageLogger t=2022-06-19T12:01:44.826668689Z
level=info msg="storage starting"
grafana_1 | logger=ngalert.multiorg.alertmanager t=2022-06-
19T12:01:44.8301066Z level=info msg="starting MultiOrg Alertmanager"
grafana_1 | logger=http.server t=2022-06-19T12:01:44.830362457Z level=info
msg="HTTP Server Listen" address=[::]:3000 protocol=http subUrl= socket=
influxdb_1 | [metastore] 2022/06/19 12:01:44 spun up monitoring for 1
influxdb_1 | [store] 2022/06/19 12:01:44 Using data dir: /data/db
influxdb_1 | [metastore] 2022/06/19 12:01:44 Updated node id=1
hostname=localhost:8088
influxdb_1 | [wal] 2022/06/19 12:01:45 WAL starting with 30720 ready series
size, 0.50 compaction threshold, and 52428800 partition size threshold
influxdb_1 | [wal] 2022/06/19 12:01:45 WAL writing to
/data/wal/_internal/monitor/1
influxdb_1 | [handoff] 2022/06/19 12:01:45 Starting hinted handoff service
influxdb_1 | [monitor] 2022/06/19 12:01:45 'hh' registered for diagnostics
monitoring
influxdb_1 | [handoff] 2022/06/19 12:01:45 Using data dir: /data/hh
influxdb_1 | [subscriber] 2022/06/19 12:01:45 opened service
influxdb_1 | [monitor] 2022/06/19 12:01:45 Starting monitor system
influxdb_1 | [monitor] 2022/06/19 12:01:45 'build' registered for diagnostics
monitoring
```

[illegible]

CAdvisor监控收集:

<http://localhost:8080/>

influxdb存储数据:

<http://localhost:8083/>

grafana:

<http://localhost:3000/>

end

by mao

2022/06/20
