**docker常用基础命令**

## [Docker](https://so.csdn.net/so/search?q=Docker&spm=1001.2101.3001.7020) 环境信息命令

### 、1.1、docker info

显示 Docker 系统信息，包括镜像、容器数、仓库镜像、架构等。

docker info [OPTIONS]

king@king-server:~$ docker info

Client: Docker Engine - Community

Version: 24.0.7

Context: default

Debug Mode: false

Plugins:

buildx: Docker Buildx (Docker Inc.)

Version: v0.11.2

Path: /usr/libexec/docker/cli-plugins/docker-buildx

compose: Docker Compose (Docker Inc.)

Version: v2.21.0

Path: /usr/libexec/docker/cli-plugins/docker-compose

Server:

Containers: 6

Running: 1

Paused: 0

Stopped: 5

Images: 3

Server Version: 24.0.7

Storage Driver: overlay2

Backing Filesystem: extfs

Supports d\_type: true

Using metacopy: false

Native Overlay Diff: true

userxattr: false

Logging Driver: json-file

Cgroup Driver: cgroupfs

Cgroup Version: 1

Plugins:

Volume: local

Network: bridge host ipvlan macvlan null overlay

Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog

Swarm: inactive

Runtimes: runc io.containerd.runc.v2

Default Runtime: runc

Init Binary: docker-init

containerd version: 3dd1e886e55dd695541fdcd67420c2888645a495

runc version: v1.1.10-0-g18a0cb0

init version: de40ad0

Security Options:

apparmor

seccomp

Profile: builtin

Kernel Version: 5.4.0-169-generic

Operating System: Ubuntu 20.04.5 LTS

OSType: linux

Architecture: x86\_64

CPUs: 4

Total Memory: 3.797GiB

Name: king-server

ID: 2cf468ba-1204-4a1a-bebf-e859aaab7ca2

Docker Root Dir: /var/lib/docker

Debug Mode: false

Experimental: false

Insecure Registries:

127.0.0.0/8

Live Restore Enabled: false

### 1.2、docker version

显示 Docker 版本信息。

docker version [OPTIONS]

king@king-server:~$ docker version

Client: Docker Engine - Community

Version: 24.0.7

API version: 1.43

Go version: go1.20.10

Git commit: afdd53b

Built: Thu Oct 26 09:08:01 2023

OS/Arch: linux/amd64

Context: default

Server: Docker Engine - Community

Engine:

Version: 24.0.7

API version: 1.43 (minimum version 1.12)

Go version: go1.20.10

Git commit: 311b9ff

Built: Thu Oct 26 09:08:01 2023

OS/Arch: linux/amd64

Experimental: false

containerd:

Version: 1.6.26

GitCommit: 3dd1e886e55dd695541fdcd67420c2888645a495

runc:

Version: 1.1.10

GitCommit: v1.1.10-0-g18a0cb0

docker-init:

Version: 0.19.0

GitCommit: de40ad0

## 2、系统日志信息常用命令

### 2.1、docker events

从[服务器](https://so.csdn.net/so/search?q=%E6%9C%8D%E5%8A%A1%E5%99%A8&spm=1001.2101.3001.7020" \t "_blank)获取实时事件（监听docker事件）。

docker events [OPTIONS]

OPTIONS说明：

-f ：根据条件过滤事件；

--since ：从指定的时间戳后显示所有事件;

--until ：流水时间显示到指定的时间为止；

# 第一个终端执行

docker events

# 第二个终端操作容器，比如docker run -d nginx

docker start/stop/restart

# 查看第一个终端输出

、# 启动容器事件

2022-12-08T14:18:01.476501682Z container create 36f6736b9c6289da6db95898507796a378cd03e29b51166039c5a5a593377fb0 (image=nginx, maintainer=NGINX Docker Maintainers <docker-maint@nginx.com>, name=stupefied\_newton)

2022-12-08T14:18:01.578704583Z network connect fdb00a147400b60e2c0706129ed2b143a13c02c5675b5fc2871c5d4ed42aa478 (container=36f6736b9c6289da6db95898507796a378cd03e29b51166039c5a5a593377fb0, name=bridge, type=bridge)

2022-12-08T14:18:03.310080179Z container start 36f6736b9c6289da6db95898507796a378cd03e29b51166039c5a5a593377fb0 (image=nginx, maintainer=NGINX Docker Maintainers <docker-maint@nginx.com>, name=stupefied\_newton)

# stop事件

2022-12-08T14:20:19.252053300Z container kill 36f6736b9c6289da6db95898507796a378cd03e29b51166039c5a5a593377fb0 (image=nginx, maintainer=NGINX Docker Maintainers <docker-maint@nginx.com>, name=stupefied\_newton, signal=3)

2022-12-08T14:20:19.365758840Z container die 36f6736b9c6289da6db95898507796a378cd03e29b51166039c5a5a593377fb0 (exitCode=0, image=nginx, maintainer=NGINX Docker Maintainers <docker-maint@nginx.com>, name=stupefied\_newton)

2022-12-08T14:20:19.468679557Z network disconnect fdb00a147400b60e2c0706129ed2b143a13c02c5675b5fc2871c5d4ed42aa478 (container=36f6736b9c6289da6db95898507796a378cd03e29b51166039c5a5a593377fb0, name=bridge, type=bridge)

2022-12-08T14:20:19.535147938Z container stop 36f6736b9c6289da6db95898507796a378cd03e29b51166039c5a5a593377fb0 (image=nginx, maintainer=NGINX Docker Maintainers <docker-maint@nginx.com>, name=stupefied\_newton)

### 2.2、docker logs

docker logs，通常用于排查容器里面的问题。比如容器里部署的[程序执行](https://so.csdn.net/so/search?q=%E7%A8%8B%E5%BA%8F%E6%89%A7%E8%A1%8C&spm=1001.2101.3001.7020)不起来，由于docker做了隔离，就需要使用docker logs来排除问题。

docker logs [OPTIONS] CONTAINER

OPTIONS说明：

-f : 跟踪日志输出

--since :显示某个开始时间的所有日志

-t : 显示时间戳

--tail :仅列出最新N条容器日志

-n :显示最后多少行的日志

king@king-server:~$ docker logs -f affectionate\_pare

/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration

/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/

/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh

10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf

10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf

/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh

/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh

/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh

/docker-entrypoint.sh: Configuration complete; ready for start up

2024/01/21 11:12:09 [notice] 1#1: using the "epoll" event method

2024/01/21 11:12:09 [notice] 1#1: nginx/1.25.3

2024/01/21 11:12:09 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)

2024/01/21 11:12:09 [notice] 1#1: OS: Linux 5.4.0-169-generic

2024/01/21 11:12:09 [notice] 1#1: getrlimit(RLIMIT\_NOFILE): 1048576:1048576

2024/01/21 11:12:09 [notice] 1#1: start worker processes

2024/01/21 11:12:09 [notice] 1#1: start worker process 29

2024/01/21 11:12:09 [notice] 1#1: start worker process 30

2024/01/21 11:12:09 [notice] 1#1: start worker process 31

2024/01/21 11:12:09 [notice] 1#1: start worker process 32’

king@king-server:~$ docker logs -n 10 affectionate\_pare

2024/01/21 11:12:09 [notice] 1#1: using the "epoll" event method

2024/01/21 11:12:09 [notice] 1#1: nginx/1.25.3

2024/01/21 11:12:09 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)

2024/01/21 11:12:09 [notice] 1#1: OS: Linux 5.4.0-169-generic

2024/01/21 11:12:09 [notice] 1#1: getrlimit(RLIMIT\_NOFILE): 1048576:1048576

2024/01/21 11:12:09 [notice] 1#1: start worker processes

2024/01/21 11:12:09 [notice] 1#1: start worker process 29

2024/01/21 11:12:09 [notice] 1#1: start worker process 30

2024/01/21 11:12:09 [notice] 1#1: start worker process 31

2024/01/21 11:12:09 [notice] 1#1: start worker process 32

### 2.3、docker history

查看指定镜像的创建历史。

docker history [OPTIONS] IMAGE

OPTIONS说明：

-H :以可读的格式打印镜像大小和日期，默认为true；

--no-trunc :显示完整的提交记录；

-q :仅列出提交记录ID

king@king-server:~$ docker history affectionate\_pare

Error response from daemon: No such image: affectionate\_pare:latest

king@king-server:~$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

registry 2 909c3ff012b7 7 weeks ago 25.4MB

nginx latest d453dd892d93 2 months ago 187MB

hello-world latest d2c94e258dcb 8 months ago 13.3kB

king@king-server:~$ docker history nginx

IMAGE CREATED CREATED BY SIZE COMMENT

d453dd892d93 2 months ago CMD ["nginx" "-g" "daemon off;"] 0B buildkit.dockerfile.v0

<missing> 2 months ago STOPSIGNAL SIGQUIT 0B buildkit.dockerfile.v0

<missing> 2 months ago EXPOSE map[80/tcp:{}] 0B buildkit.dockerfile.v0

<missing> 2 months ago ENTRYPOINT ["/docker-entrypoint.sh"] 0B buildkit.dockerfile.v0

<missing> 2 months ago COPY 30-tune-worker-processes.sh /docker-ent… 4.62kB buildkit.dockerfile.v0

<missing> 2 months ago COPY 20-envsubst-on-templates.sh /docker-ent… 3.02kB buildkit.dockerfile.v0

<missing> 2 months ago COPY 15-local-resolvers.envsh /docker-entryp… 298B buildkit.dockerfile.v0

<missing> 2 months ago COPY 10-listen-on-ipv6-by-default.sh /docker… 2.12kB buildkit.dockerfile.v0

<missing> 2 months ago COPY docker-entrypoint.sh / # buildkit 1.62kB buildkit.dockerfile.v0

<missing> 2 months ago RUN /bin/sh -c set -x && groupadd --syst… 112MB buildkit.dockerfile.v0

<missing> 2 months ago ENV PKG\_RELEASE=1~bookworm 0B buildkit.dockerfile.v0

<missing> 2 months ago ENV NJS\_VERSION=0.8.2 0B buildkit.dockerfile.v0

<missing> 2 months ago ENV NGINX\_VERSION=1.25.3 0B buildkit.dockerfile.v0

<missing> 2 months ago LABEL maintainer=NGINX Docker Maintainers <d… 0B buildkit.dockerfile.v0

<missing> 2 months ago /bin/sh -c #(nop) CMD ["bash"] 0B

<missing> 2 months ago /bin/sh -c #(nop) ADD file:ac3cd70031d35e46d… 74.8MB

## 3、容器的生命周期管理命令

### 3.1、docker create

创建一个新的容器但不启动它，语法同docker run。

docker create nginx

### 3.2、docker run

创建一个新的容器并运行一个命令。

docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

常用选项：

* –add-host：容器中hosts文件添加 host:ip 映射记录
* docker run --rm -it --add-host db-static:86.75.30.9 ubuntu cat /etc/hosts
* -a, --attach：附加到 STDIN, STDOUT 或 STDERR。

docker run --rm -a stdin -a stdout -i -t ubuntu /bin/bash

* –cidfile：将容器id写入到指定文件。
* docker run --rm --cidfile /tmp/idfile ubuntu

-d, --detach：后台运行容器并打印容器id。

–detach-keys：指定将容器切回后台的按键，默认：CTRL-p+CTRL-q。

–disable-content-trust：跳过镜像验证，默认为TRUE。

–domainname：设置容器网络服务域名。

–entrypoint：覆盖镜像默认程序入口。

docker run --rm -it --entrypoint /bin/bash nginx

* -e, --env：设置环境变量。
* –env-file：从文件中读取环境变量。
* –expose：暴露一个或多个端口。
* –group-add：为容器用户添加更多用户组。

docker run --rm --group-add audio --group-add nogroup --group-add 777 ubuntu id

* -h, --hostname string：设置主机名。
* –init 在容器内运行init，转发信号并捕获进程。

# ps 对比容器的元祖进程

docker run --rm -it ubuntu bash

docker run --rm -it --init ubuntu bash

-i, --interactive：保持标准输入设备处于打开状态。

-l, --label list：设置容器元数据labels。

–label-file list：从文件中读取以换行作为分隔符的label。

–link：添加一个其他容器的链接，及修改当容器的/etc/hosts文件。

–log-driver：容器日志驱动，none、local、json-file、syslog、journald、gelf、fluentd、awslogs、splunk、etwlogs、gcplogs、logentries。

–log-opt：日志驱动选项。

–mount：将文件系统挂载到容器，与 -v 类似。

# 挂载volume

# 挂载volume

docker run -t -i --rm --mount type=volume,target=/data ubuntu bash

# 创建目录

mkdir /tmp/data

# 修改目录所有权，仅当使用userns的情况下需要

sudo chown 165536:root /tmp/data

# bind 磁盘已存在目录

docker run -t -i --rm --mount type=bind,src=/tmp/data,dst=/data ubuntu bash

–name：指定容器名称。

–privileged：授予容器超级权限。

-p, --publish：发布容器端口都主机端口。

-P, --publish-all：将所有暴露的端口发布到主机随机端口。

–pull：运行前拉取图像，（always|missing|never）,默认 missing。

–pids-limit：限制容器中pid个数。

–read-only ：将容器跟文件系统装载为只读。

–restart：当容器退出时的重启策略，默认为no。

–rm：当容器退出时，自动删除容器。

–sig-proxy：将接收到的信号代理到进程（默认为true）。

–stop-signal：停止容器的信号（默认为“SIGTERM”）。

–stop-timeout：容器停止超时时长，单位s。

–tmpfs：装载tmpfs目录。

docker run -dit --rm --tmpfs /run1:rw,noexec,nosuid,size=65536k ubuntu bash   
-t, --tty：分配一个伪终端设备。

-u, --user： 用户名或用户ID (格式: <name|uid>[:<group|gid>])。

-v, --volume ：绑定数据卷。

–volume-driver：容器的可选卷驱动程序。

–volumes-from：从指定的容器装载数据卷。

-w, --workdir：容器内的工作目录。