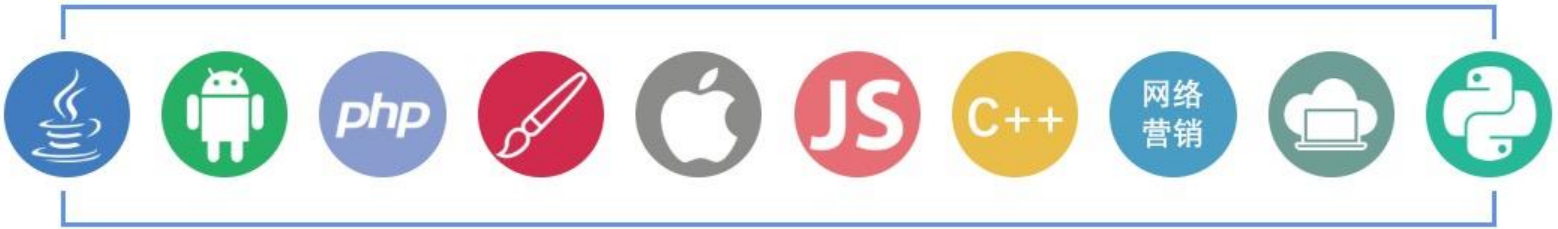
Docker核心技术之容器



Docker核心技术之容器-课程概要

- 容器简介
- 容器与虚拟机
- 容器生命周期
- 容器生命周期管理
- 容器信息查看
- 容器运行时操作
- 容器总结

Docker核心技术之容器

容器简介

什么是容器

容器（Container）：容器是一种轻量级、可移植、并将应用程序进行的打包的技术，使应用程序可以在几乎任何地方以相同的方式运行

- Docker将镜像文件运行起来后，产生的对象就是容器。容器相当于是镜像运行起来的一个实例。
- 容器具备一定的生命周期。
- 另外，可以借助docker ps命令查看运行的容器，如同在linux上利用ps命令查看运行着的进程那样。

Docker核心技术之容器

容器与虚拟机

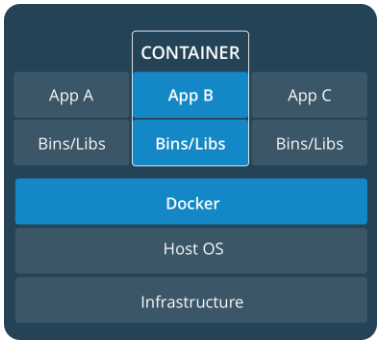
Docker容器与虚拟机相同点

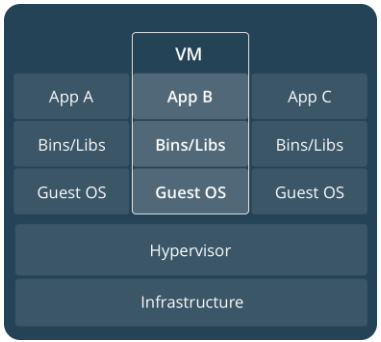
- 容器和虚拟机一样，都会对物理硬件资源进行共享使用。
- 容器和虚拟机的生命周期比较相似（创建、运行、暂停、关闭等等）。
- 容器中或虚拟机中都可以安装各种应用，如redis、mysql、nginx等。也就是说，在容器中的操作，如同在一个虚拟机(操作系统)中操作一样。
- 同虚拟机一样，容器创建后，会存储在宿主机上：linux上位于/var/lib/docker/containers下

Docker容器与虚拟机不同点

注意：容器并不是虚拟机，但它们有很多相似的地方

- 虚拟机的创建、启动和关闭都是基于一个完整的操作系统。一个虚拟机就是一个完整的操作系统。而容器直接运行在宿主机的内核上，其本质上以一系列进程的结合。
- 容器是轻量级的，虚拟机是重量级的。首先容器不需要额外的资源来管理(不需要Hypervisor、Guest OS)，虚拟机额外更多的性能消耗；其次创建、启动或关闭容器，如同创建、启动或者关闭进程那么轻松，而创建、启动、关闭一个操作系统就没那么方便了。
- 也因此，意味着在给定的硬件上能运行更多数量的容器，甚至可以直接把Docker运行在虚拟机上。

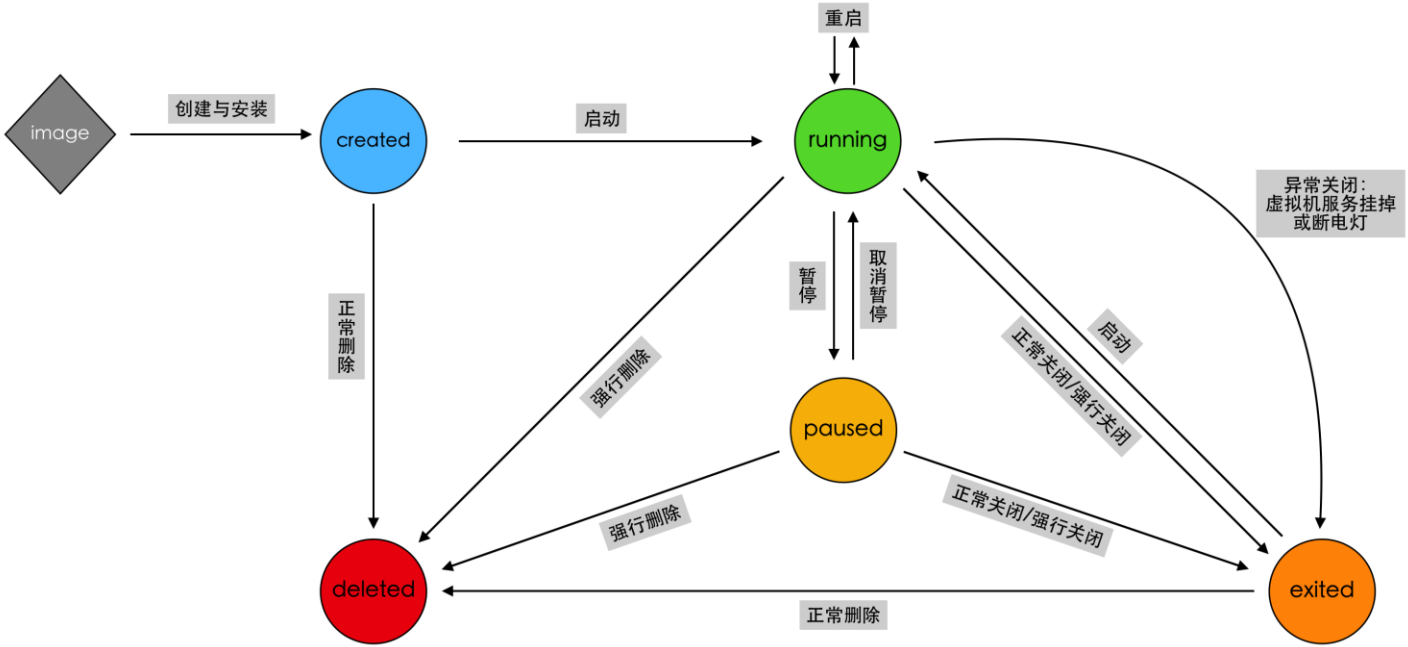




Docker核心技术之容器

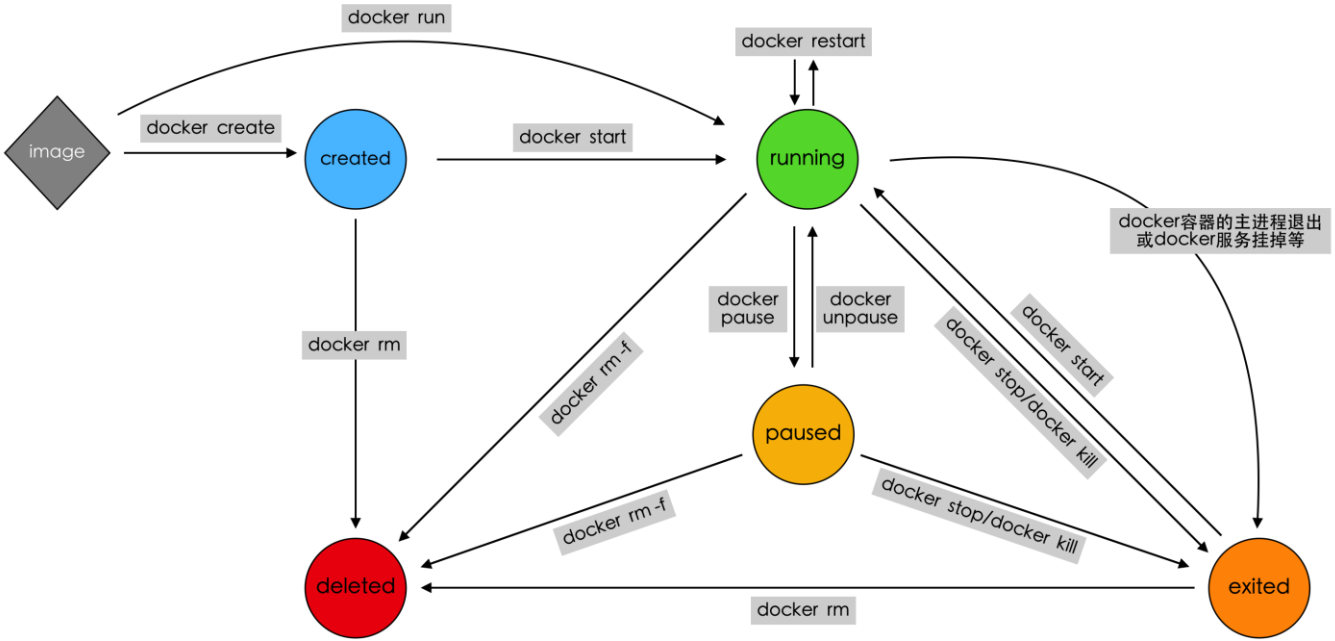
容器的生命周期

虚拟机的生命周期



虚拟机的生命周期

容器的生命周期



Docker容器的生命周期

Docker核心技术之容器

容器的生命周期管理

容器创建 – docker create

- 作用：

  利用镜像创建出一个Created 状态的待启动容器

- 命令格式：

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

- 命令参数(OPTIONS)：查看更多

  -t, --tty　　分配一个伪TTY，也就是分配虚拟终端

  -i, --interactive　　即使没有连接，也要保持STDIN打开

  --name　　为容器起名，如果没有指定将会随机产生一个名称

- 命令参数（COMMAND\ARG）：

  COMMAND 表示容器启动后，需要在容器中执行的命令，如ps、ls 等命令

  ARG 表示执行 COMMAND 时需要提供的一些参数，如ps 命令的 aux、ls命令的-a等等

容器创建 – docker create

- 命令演示：

```
[root@centos-linux ~]# docker create --name test-container centos ps -A
65ebe719d77c859d660e843a685cc24b323ed801fbb80dbd4d6124796075be97
[root@centos-linux ~]# docker create -ti --name test-container2 centos /bin/bash
1f1e93707cada941e21ce64db6fcefd92eded35cfd6ee06a6aa521123aa67c4a
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         12 seconds ago      Created                                 test-container2
65ebe719d77c        centos              "ps -A"             25 seconds ago      Created                                 test-container
```

容器启动 – docker start

- 作用：

  将一个或多个处于创建状态或关闭状态的容器启动起来

- 命令格式：

  docker start [OPTIONS] CONTAINER [CONTAINER...]

- 命令参数(OPTIONS)：

-a, --attach	**将当前shell的 STDOUT/STDERR 连接到容器上**
-i, --interactive	**将当前shell的 STDIN连接到容器上**

容器启动 – docker start

- 命令演示：

```
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         15 minutes ago      Created                                 test-container2
65ebe719d77c        centos              "ps -A"             15 minutes ago      Created                                 test-container
```

```
[root@centos-linux ~]# docker start -a 65eb
  PID TTY          TIME CMD
    1 ?        00:00:00 ps
[root@centos-linux ~]# docker start test-container2
test-container2
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                          PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         18 minutes ago      Up 4 seconds                                        test-container2
65ebe719d77c        centos              "ps -A"             18 minutes ago      Exited (0) 10 seconds ago                           test-container
```

容器创建并启动 – docker run

- 作用：

  利用镜像创建并启动一个容器

- 命令格式：

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

- 命令参数(OPTIONS)：查看更多

-t, --tty	**分配一个伪TTY，也就是分配虚拟终端**
-i, --interactive	**即使没有连接，也要保持STDIN打开**
--name	**为容器起名，如果没有指定将会随机产生一个名称**
-d, --detach	**在后台运行容器并打印出容器ID**
--rm	**当容器退出运行后，自动删除容器**

- 命令参数（COMMAND\ARG）：

  COMMAND 表示容器启动后，需要在容器中执行的命令，如ps、ls 等命令

  ARG 表示执行 COMMAND 时需要提供的一些参数，如ps 命令的 aux、ls命令的-a等等

容器创建并启动 – docker run

- 命令演示：

```
[root@centos-linux ~]# docker run centos ps -A
  PID TTY          TIME CMD
    1 ?        00:00:00 ps
[root@centos-linux ~]# docker run -d centos ps -A
6676290c751df9c58f8929b8fc40b83104d317f77926b5ee47d325f97b14502c
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
6676290c751d        centos              "ps -A"             4 seconds ago       Exited (0) 3 seconds ago                        hungry_lamport
9d85ade78dc9        centos              "ps -A"             9 seconds ago       Exited (0) 8 seconds ago                        stupefied_bhaskara
```

```
[root@centos-linux ~]# docker run --rm -d centos ps -A
05c177fbefad5df3ea7c47392783719925db7af9e7cf057ee15e1a59151086da
[root@centos-linux ~]# docker run -d centos ps -A
5ab2793d95351f677b609fdd6a2d99fd1a9d21ed8af6783b1b0b8f8dd44cbba0
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
5ab2793d9535        centos              "ps -A"             3 seconds ago       Exited (0) 2 seconds ago                        thirsty_boyd
[root@centos-linux ~]#
```

docker run 与 docker create + docker start

- **docker run** 相当于 **docker create + docker start –a** **前台模式**
- **docker run -d** 相当于 **docker create + docker start** **后台模式**

容器暂停 – docker pause

- 作用：

  暂停一个或多个处于运行状态的容器

- 命令格式：

  docker pause CONTAINER [CONTAINER...]

- 命令参数(OPTIONS)：

  无

容器暂停 – docker pause

- 命令演示：

```
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Up About an hour                                test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) About an hour ago                    test-container
[root@centos-linux ~]# docker pause test-container2
test-container2
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Up About an hour (Paused)                       test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) About an hour ago                    test-container
[root@centos-linux ~]#
```

容器取消暂停 – docker unpause

- 作用：

  取消一个或多个处于暂停状态的容器，恢复运行

- 命令格式：

  docker unpause CONTAINER [CONTAINER...]

- 命令参数(OPTIONS)：

  无

容器取消暂停 – docker unpause

- 命令演示：

```
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                       PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Up About an hour (Paused)                        test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) About an hour ago                     test-container
[root@centos-linux ~]# docker unpause test-container2
test-container2
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                       PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Up About an hour                                 test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) About an hour ago                     test-container
[root@centos-linux ~]#
```

容器关闭 – docker stop

- 作用：

  关闭一个或多个处于暂停状态或者运行状态的容器

- 命令格式：

  docker stop [OPTIONS] CONTAINER [CONTAINER...]

- 命令参数(OPTIONS)：

  -t, --time int　　关闭前，等待的时间，单位秒(默认 10s)

容器关闭 – docker stop

- 命令演示：

```
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Up About an hour                                test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) About an hour ago                    test-container
[root@centos-linux ~]# docker stop -t 1 1f1e
1f1e
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Exited (137) 2 seconds ago                      test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) About an hour ago                    test-container
[root@centos-linux ~]#
```

容器终止 – docker kill

- 作用：

  强制并立即关闭一个或多个处于暂停状态或者运行状态的容器

- 命令格式：

  docker kill [OPTIONS] CONTAINER [CONTAINER...]

- 命令参数(OPTIONS)：

  -s, --signal string　　指定发送给容器的关闭信号 (默认“KILL”信号)

容器终止 – docker kill

- 命令演示：

```
[root@centos-linux ~]# docker start test-container2
test-container2
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Up 3 seconds                                    test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) 2 hours ago                          test-container
[root@centos-linux ~]# docker kill test-container2
test-container2
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Exited (137) 3 seconds ago                      test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) 2 hours ago                          test-container
```

docker stop和docker kill的区别

- 前提知识点：
  - Linux其中两种终止进程的信号是：SIGTERM和SIGKILL
  - **SIGKILL**信号：无条件终止进程信号。进程接收到该信号会立即终止，不进行清理和暂存工作。该信号不能被忽略、处理和阻塞，它向系统管理员提供了可以杀死任何进程的方法。
  - **SIGTERM**信号：程序终结信号，可以由kill命令产生。与SIGKILL不同的是，SIGTERM信号**可以被阻塞和终止**，以便程序在退出前可以保存工作或清理临时文件等。

- docker stop 会先发出SIGTERM信号给进程，告诉进程即将会被关闭。在-t指定的等待时间过了之后，将会立即发出SIGKILL信号，直接关闭容器。
- docker kill 直接发出SIGKILL信号关闭容器。但也可以通过-s参数修改发出的信号。

- 因此会发现在docker stop的等过过程中，如果终止docker stop的执行，容器最终没有被关闭。而docker kill几乎是立刻发生，无法撤销。

- 此外还有些异常原因也会导致容器被关闭，比如docker daemon重启、容器内部进程运行发生错误等等“异常原因”。

容器重启 – docker restart

- 作用：

  重启一个或多个处于运行状态、暂停状态、关闭状态或者新建状态的容器

  该命令相当于stop和start命令的结合

- 命令格式：

  docker restart [OPTIONS] CONTAINER [CONTAINER...]

- 命令参数(OPTIONS)：

-t, --time int	**重启前，等待的时间，单位秒(默认 10s)**
	实则是关闭前等待的时间

容器删除 – docker container rm

- 作用：

  删除一个或多个容器

- 命令格式：

  docker container rm [OPTIONS] CONTAINER [CONTAINER...]

  或者 docker rm [OPTIONS] CONTAINER [CONTAINER...]

- 命令参数(OPTIONS)：

-f, --force	**强行删除容器(会使用 SIGKILL信号)**
-v, --volumes	**同时删除绑定在容器上的数据卷**

容器删除 – docker container rm

- 命令演示：

```
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
1f1e93707cad        centos              "/bin/bash"         2 hours ago         Up 9 minutes                                    test-container2
65ebe719d77c        centos              "ps -A"             2 hours ago         Exited (0) 9 minutes ago                        test-container
[root@centos-linux ~]# docker rm -f 1f1e test-container
1f1e
test-container
[root@centos-linux ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
[root@centos-linux ~]#
```

Docker核心技术之容器

容器信息查看

容器详细信息 – docker container inspect

- 作用：

  查看本地一个或多个容器的详细信息

- 命令格式：

  docker container inspect [OPTIONS] CONTAINER [CONTAINER...]

  或者 **docker inspect [OPTIONS] CONTAINER [CONTAINER...]**

- 命令参数(OPTIONS)：

  -f, --format string　　利用特定Go语言的format格式输出结果

  -s, --size　　显示总大小

容器详细信息 – docker container inspect

- 命令演示：

```
[root@centos-linux ~]# docker container inspect -f "{{json .State.Status}}" 5ab2
"exited"
[root@centos-linux ~]# docker container inspect 5ab2
[
    {
        "Id": "5ab2793d95351f677b609fdd6a2d99fd1a9d21ed8af6783b1b0b8f8dd44cbba0",
        "Created": "2018-05-11T13:27:06.494039894Z",
        "Path": "ps",
        "Args": [
            "-A"
        ],
        "State": {
            "Status": "exited",
            "Running": false,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 0,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2018-05-11T13:27:06.720966292Z",
            "FinishedAt": "2018-05-11T13:27:06.730018594Z"
        ……
```

容器日志信息 – docker logs

- 作用：

  查看容器的日志信息

- 命令格式：

  docker logs [OPTIONS] CONTAINER

- 命令参数(OPTIONS)：

--details	**显示日志的额外信息**
-f, --follow	**动态跟踪显示日志信息**
--since string	**只显示某事时间节点之后的**
--tail string	**显示倒数的行数(默认全部)**
-t, --timestamps	**显示timestamps时间**
--until string	**只显示某事时间节点之前的**

- 注意：

  容器日志中记录的是容器主进程的输出STDOUT\STDERR

容器重命名 – docker rename

- 作用：

  修改容器的名称

- 命令格式：

  docker rename CONTAINER NEW_NAME

- 命令参数(OPTIONS)：

  无

Docker核心技术之容器

容器运行时操作

容器连接 – docker attach

- 作用：

  将当前终端的STDIN、STDOUT、STDERR绑定到正在运行的容器的主进程上实现连接

- 命令格式：

  docker attach [OPTIONS] CONTAINER

- 命令参数(OPTIONS)：

  --no-stdin　　不绑定STDIN

容器中执行新命令 – docker exec

- 作用：

  在容器中运行一个命令

- 命令格式：

  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

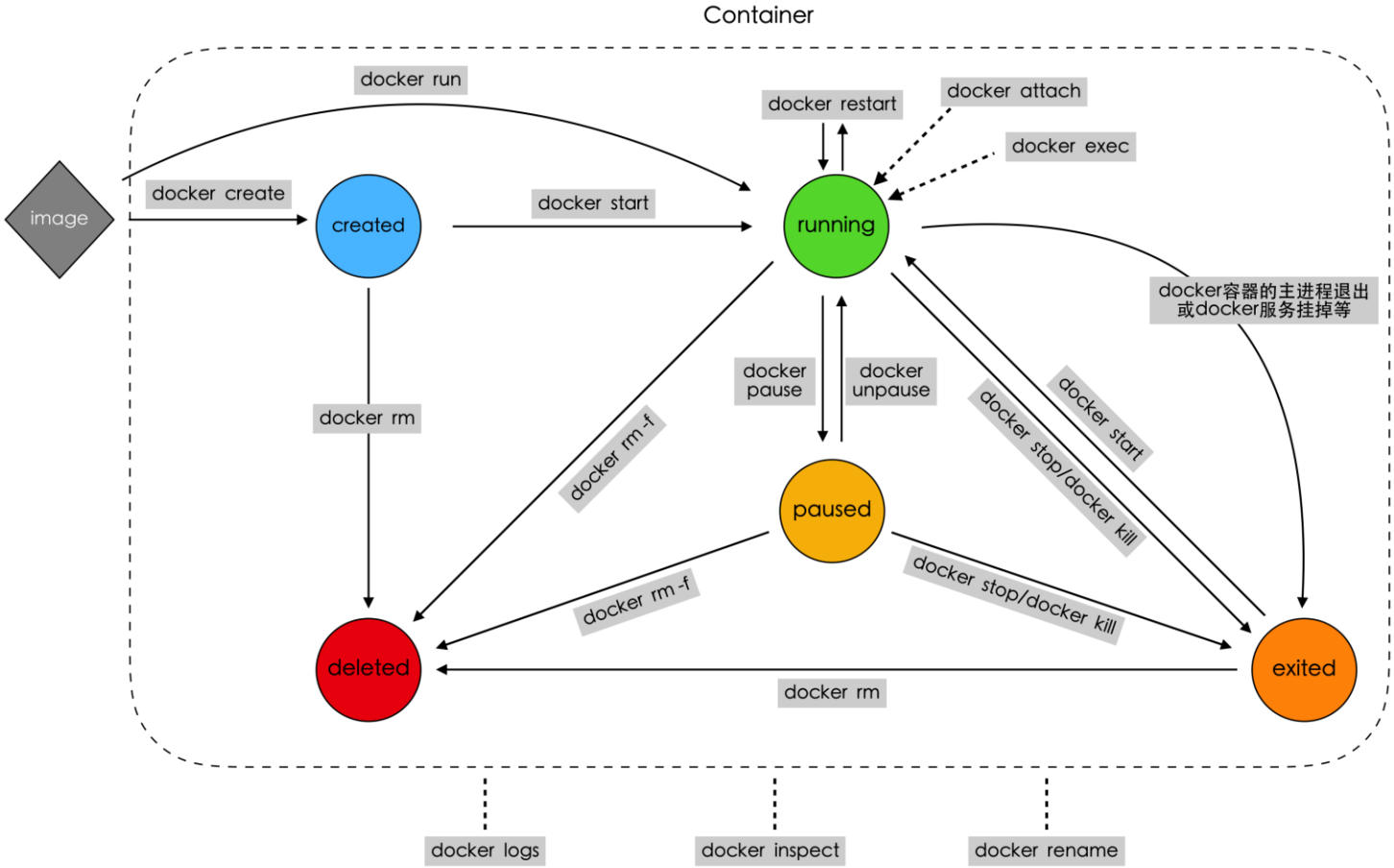
- 命令参数(OPTIONS)：

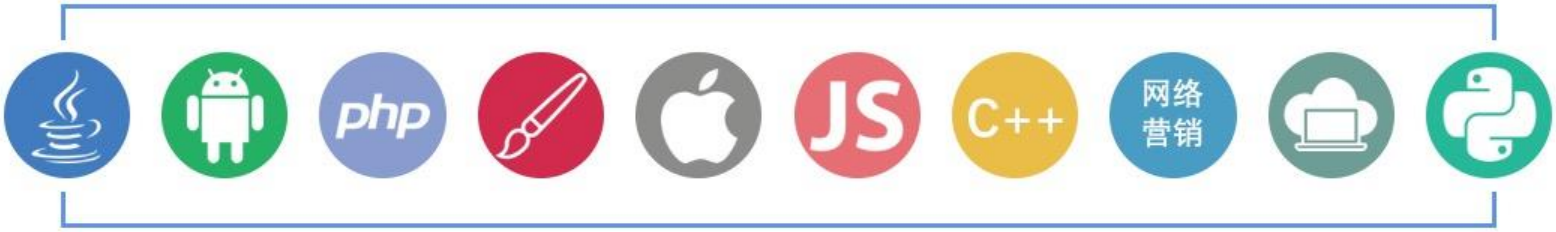
-d, --detach	**后台运行命令**
-i, --interactive	**即使没连接容器，也将当前的STDIN绑定上**
-t, --tty	**分配一个虚拟终端**
-w, --workdir string	**指定在容器中的工作目录**
-e, --env list	**设置容器中运行时的环境变量**

Docker核心技术之容器

容器总结

Docker核心技术之容器 - 总结





Thank you!

改变中国IT教育，我们正在行动