

NAME

docker-container-update - Update configuration of one or more containers

SYNOPSIS

docker container update [OPTIONS] CONTAINER [CONTAINER...]

DESCRIPTION

The **docker container update** command dynamically updates container configuration. You can use this command to prevent containers from consuming too many resources from their Docker host. With a single command, you can place limits on a single container or on many. To specify more than one container, provide space-separated list of container names or IDs.

With the exception of the **--kernel-memory** option, you can specify these options on a running or a stopped container. On kernel version older than 4.6, You can only update **--kernel-memory** on a stopped container or on a running container with kernel memory initialized.

OPTIONS**kernel-memory**

Kernel memory limit (format: <number> [<unit>], where unit = b, k, m or g)

Note that on kernel version older than 4.6, you can not update kernel memory on a running container if the container is started without kernel memory initialized, in this case, it can only be updated after it's stopped. The new setting takes effect when the container is started.

memory

Memory limit (format: , where unit = b, k, m or g)

Note that the memory should be smaller than the already set swap memory limit. If you want update a memory limit bigger than the already set swap memory limit, you should update swap memory limit at the same time. If you don't set swap memory limit on docker create/run but only memory limit, the swap memory is double the memory limit.

EXAMPLES

The following sections illustrate ways to use this command.

Update a container's cpu-shares

To limit a container's cpu-shares to 512, first identify the container name or ID. You can use **docker ps** to find these values. You can also use the ID returned from the **docker run** command. Then, do the following:

```
$ docker container update --cpu-shares 512 abebf7571666
```

Update a container with cpu-shares and memory

To update multiple resource configurations for multiple containers:

```
$ docker container update --cpu-shares 512 -m 300M abebf7571666 hopeful_morse
```

Update a container's kernel memory constraints

You can update a container's kernel memory limit using the **--kernel-memory** option. On kernel version older than 4.6, this option can be updated on a running container only if the container was started with **--kernel-memory**. If the container was started *without* **--kernel-memory** you need to stop the container before updating kernel memory.

NOTE: The **--kernel-memory** option has been deprecated since Docker 20.10.

For example, if you started a container with this command:

```
$ docker run -dit --name test --kernel-memory 50M ubuntu bash
```

You can update kernel memory while the container is running:

```
$ docker container update --kernel-memory 80M test
```

If you started a container *without* kernel memory initialized:

```
$ docker run -dit --name test2 --memory 300M ubuntu bash
```

Update kernel memory of running container `test2` will fail. You need to stop the container before updating the **--kernel-memory** setting. The next time you start it, the container uses the new value.

Kernel version newer than (include) 4.6 does not have this limitation, you can use **--kernel-memory** the same way as other options.

Update a container's restart policy

You can change a container's restart policy on a running container. The new restart policy takes effect instantly after you run `docker container update` on a container.

To update restart policy for one or more containers:

```
$ docker container update --restart=on-failure:3 abebf7571666 hopeful_morse
```

Note that if the container is started with **--rm** flag, you cannot update the restart policy for it. The `AutoRemove` and `RestartPolicy` are mutually exclusive for the container.

OPTIONS

--blkio-weight=0 Block IO (relative weight), between 10 and 1000, or 0 to disable (default 0)

--cpu-period=0	Limit CPU CFS (Completely Fair Scheduler) period
--cpu-quota=0	Limit CPU CFS (Completely Fair Scheduler) quota
--cpu-rt-period=0	Limit the CPU real-time period in microseconds
--cpu-rt-runtime=0	Limit the CPU real-time runtime in microseconds
-c, --cpu-shares=0	CPU shares (relative weight)
--cpus=	Number of CPUs
--cpuset-cpus=""	CPUs in which to allow execution (0-3, 0,1)
--cpuset-mems=""	MEMs in which to allow execution (0-3, 0,1)
-h, --help[=false]	help for update
--kernel-memory=0	Kernel memory limit
-m, --memory=0	Memory limit
--memory-reservation=0	Memory soft limit
--memory-swap=0	Swap limit equal to memory plus swap: '-1' to enable unlimited swap
--pids-limit=0	Tune container pids limit (set -1 for unlimited)
--restart=""	Restart policy to apply when a container exits

SEE ALSO**docker-container(1)**