



## ***Rocky Enterprise Linux 9.2 Manual Pages on command 'docker-container-commit.1'***

**C:\>man docker-container-commit.1**

DOCKER(1) Docker User Manuals DOCKER(1)

### NAME

docker-container-commit - Create a new image from a container's changes

### SYNOPSIS

docker container commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

### DESCRIPTION

Create a new image from an existing container specified by name or container ID.

The new image will contain the contents of the container filesystem, excluding any data volumes. Refer to `docker-tag(1)` for more information about valid image and tag names.

While the `docker commit` command is a convenient way of extending an existing image, you should prefer the use of a `Dockerfile` and `docker build` for generating images that you intend to share with other people.

### EXAMPLES

Creating a new image from an existing container

An existing Fedora based container has had Apache installed while running in `inter?` active mode with the bash shell. Apache is also running. To create a new image run `docker ps` to find the container's ID and then run:

```
$ docker commit -m="Added Apache to Fedora base image" \  
-a="A D Ministrator" 98bd7fc99854 fedora/fedora_httpd:20
```

Note that only `a-z0-9-_.` are allowed when naming images from an existing container.

Apply specified `Dockerfile` instructions while committing the image

If an existing container was created without the DEBUG environment variable set to "true", you can create a new image based on that container by first getting the container's ID with `docker ps` and then running:

```
$ docker container commit -c="ENV DEBUG=true" 98bd7fc99854 debug-image
```

#### OPTIONS

- a, --author="" Author (e.g., "John Hannibal Smith hannibal@a-team.com?mailto:hannibal@a-team.com?")
- c, --change= Apply Dockerfile instruction to the created image
- h, --help[=false] help for commit
- m, --message="" Commit message
- p, --pause[=true] Pause container during commit

#### SEE ALSO

`docker-container(1)`

Docker Community

Feb 2022

DOCKER(1)