Course › Chapter 4: M... › Manipulatin... › Modifying I...

Modifying Images

Ideally, all container images should be built using a **`Dockerfile`** to create a clean and slim set of image layers, without log files, temporary files, or other artifacts created by the container customization. Despite these recommendations, some container images may be provided as they are, without any **`Dockerfile`** available. As an alternative approach to creating new images, a running container can be changed in place and its layers saved to create a new container image. This feature is provided by the **`docker commit`** command.

Warning

Even though the **`docker commit`** command is the simplest approach to creating new images, it is not recommended because of the image size (logs and process ID files are kept in the captured layers during the **`commit`** execution), and the lack of change traceability. **`Dockerfile`** provides a robust mechanism to customize and implement changes to a container using a readable set of commands without the set of files that are generated by a running container for OS management purposes.

The syntax for the **`docker commit`** command is as follows:

```
# docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

The following table shows the important options available for the **`docker commit`** command:

Option	Description
`--author=""`	Identifies the author responsible for the container image creation.
`--message=""`	Includes a commit message to the registry.

To identify a running container in **`docker`**, run the **`docker ps`** command:

```
# docker ps
CONTAINER ID IMAGE COMMAND                CREATED
STATUS        PORTS    NAMES
87bdfcc7c656 mysql "/entrypoint.sh mysql" 14 seconds ago
Up 13 seconds 3306/tcp mysql-basic
```

Eventually, administrators might customize the image and set the container to the desired state. To identify which files were changed, created, or deleted since the container was started, **`docker`** client has a verb called **`diff`** to identify the changes made to a container. The **`diff`** verb only requires the container name or container ID:

```
# docker diff mysql-basic
C /run
C /run/mysqld
A /run/mysqld/mysqld.pid
A /run/mysqld/mysqld.sock
A /run/mysqld/mysqld.sock.lock
A /run/secrets
```

Any added file is marked with an **A**, and any changed file is marked with a **C**.

To commit the changes to another image, run the following command:

```
# docker commit mysql-basic mysql-custom
```