



# Dockerfile Advanced Cheatsheet

Advanced configuration options for building your own Docker images



## USER <username>

# Example:

```
RUN groupadd -r grp
RUN useradd -m -r -g grp myuser
USER myuser
```

Run your Docker application as a non-root user, preventing potential security flaws and exploits from occurring in your container or host. **Be sure to create the user / group in the container with RUN.**

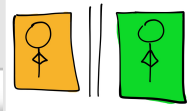


## ENV <setting>=<value>

# Examples:

```
ENV NODE_ENV=production
ENV some_secret=don't tell anyone
ENV my_setting=some value
```

Set environment variables in your Docker container. This can be used for many things, including setting NODE\_ENV or to set secrets / passwords for services used by your app.

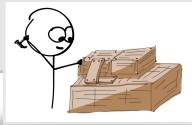


## ONBUILD <CMD> <options>

# Example:

```
ONBUILD RUN apt-get update
ONBUILD COPY . /var/app
ONBUILD RUN npm install
```

Run any valid Dockerfile command in a child image that inherits FROM this one, after that image is built. Useful for common application related commands that should not be run directly in the base image, like "npm install" etc.



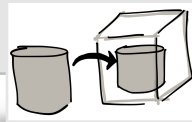
## VOLUME <folder>

# Examples:

```
VOLUME /var/data
VOLUME /var/app/images
VOLUME /var/app/uploads
```

Provide default files from an image, when mounting a volume from a host. Files in the image's volume will be copied to the host volume if they doesn't exist already.

Useful for empty database files, default images, etc.



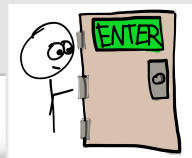
## ENTRYPOINT ["<cmd>"]

# Examples:

```
ENTRYPOINT ["node"]
ENTRYPOINT ["/bin/sh -c"]
ENTRYPOINT ["entry.sh"]
```

Specified in a base image, this command will be executed to start a container's application. It is also responsible for executing the container's CMD, which is passed to the entry point as a list of command-line parameters.

See [this WatchMeCode screencast](#) for a complete discussion of ENTRYPOINT and CMD.



For more configuration options and examples, see [the complete Dockerfile reference](#).

