# Dockerfile cheatsheet

## Inheritance

```
FROM ruby:2.2.2
```

## Run commands in strict shell

```
ENV my_var
SHELL ["/bin/bash", "-euo", "pipefail", "-c"]

# With strict mode:
RUN false # fails build like using &&
RUN echo "$myvar" # will throw error due to type
RUN true | false # will bail out of pipe
```

Using `shell` will turn on strict mode for shell commands.

## Metadata

```
LABEL version="1.0"
```

```
LABEL "com.example.vendor"="ACME Incorporated"
LABEL com.example.label-with-value="foo"
```

```
LABEL description="This text illustrates \
that label-values can span multiple lines."
```

## Variables

```
ENV APP_HOME /myapp
RUN mkdir $APP_HOME
```

```
ARG APP_HOME=""
RUN mkdir $APP_HOME
```

## Onbuild

```
ONBUILD RUN bundle install
# when used with another file
```

## Entrypoint

```
ENTRYPOINT ["executable", "param1", "param2"]
ENTRYPOINT command param1 param2
```

Configures a container that will run as an executable.

```
ENTRYPOINT exec top -b
```

This will use shell processing to substitute shell variables, and will ignore any `CMD` or `docker run` command line arguments.

## Initialization

```
RUN bundle install
```

```
WORKDIR /myapp
```

```
VOLUME ["/data"]
# Specification for mount point
```

```
ADD file.xyz /file.xyz
COPY --chown=user:group host_file.xyz /path/cont
```

## Commands

```
EXPOSE 5900
CMD    ["bundle", "exec", "rails", "server"]
```

# See also

- https://docs.docker.com/engine/reference/builder/

⚡

▶ 💬 **1 Comment** for this cheatsheet.  Write yours!

devhints.io / Search 357+ cheatsheets 🔍  🏠

←

Over 357 curated cheatsheets, by developers for developers.

**Devhints home**

## Other Devops cheatsheets

**docker-compose**
cheatsheet ●

Chef
cheatsheet ●

Deis
cheatsheet ●

AWS CLI
cheatsheet ●

CircleCI
cheatsheet

Docker CLI
cheatsheet ●

## Top cheatsheets

**Elixir**
cheatsheet ●

React.js
cheatsheet ●

Vim
cheatsheet ●

ES2015+
cheatsheet ●

Vimdiff
cheatsheet ●

Vim scripting
cheatsheet ●

cb