

# Dockerlive

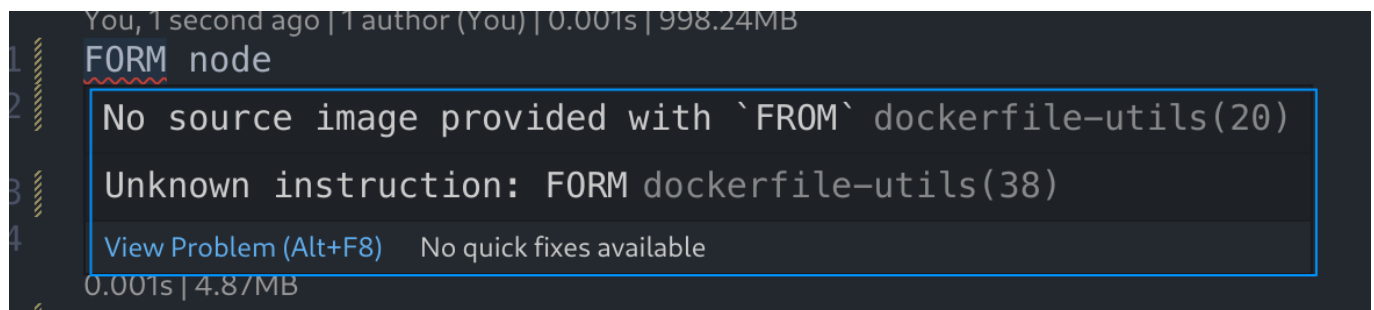
## Introduction

Dockerlive is a Visual Studio Code extension that aims to help developers who need to work with Dockerfiles. The extension continually builds an image using the Dockerfile that is currently being edited and offers several features that make Dockerfile development. You can find a list of the extension's features below.

## Features

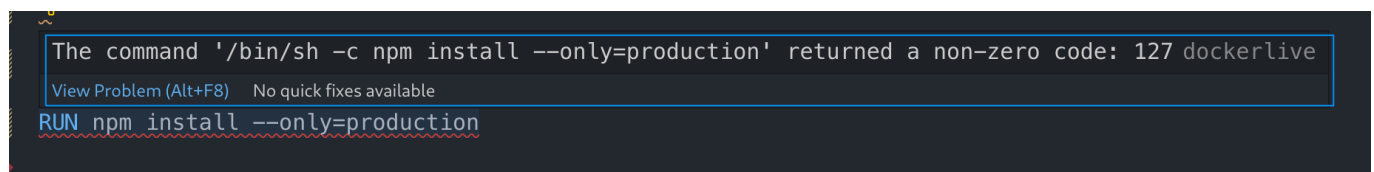
### Static Analysis Errors

The Dockerfile is scanned and any syntax errors are marked with a red underline. Hovering over the underlined region will show additional information.



### Image Build and Container Runtime Errors

If errors occur while building the image, the instruction responsible for the build failure is marked with a red underline. Hovering over the underlined region will show additional information.



### Processes running in the container

By hovering over ENTRYPOINT/CMD instructions, the processes running inside the container are shown.

```
2
3
4 COPY package*.json ./
5
6 RUN
7
8
9 USER
10
11 COPY
12
13 EXPOSE
14
15 CMD node index.js
16
```

PID	PPID	CMD
1	0	/bin/sh -c node index.js
8	1	node index.js

dockerlive

No quick fixes available

## Base image OS information

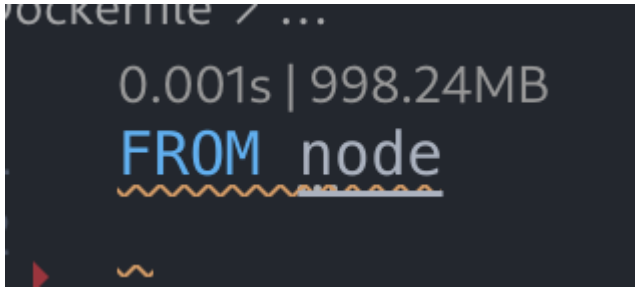
By hovering over the name of the image in the FROM instruction, it's possible to obtain information about the OS running inside the container.

```
You, 47 seconds ago | I author (You)
FROM node
OS Information:
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
dockerlive
CMD node index.js
```

## Layer Size and Build Time

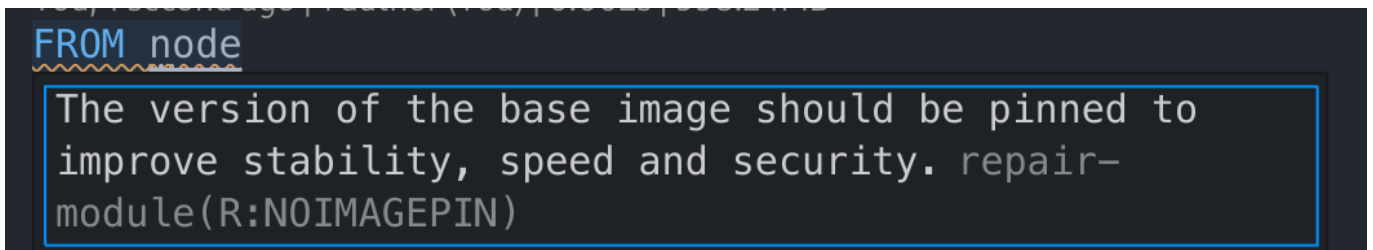
Above every instruction, some information is displayed about the layer that instruction corresponds to:

- The time it took to build that layer
- The size of that layer

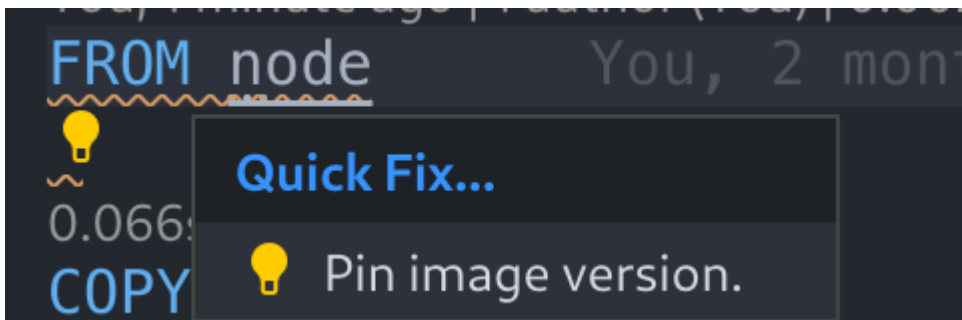


## Repair Opportunities and Quick Fixes

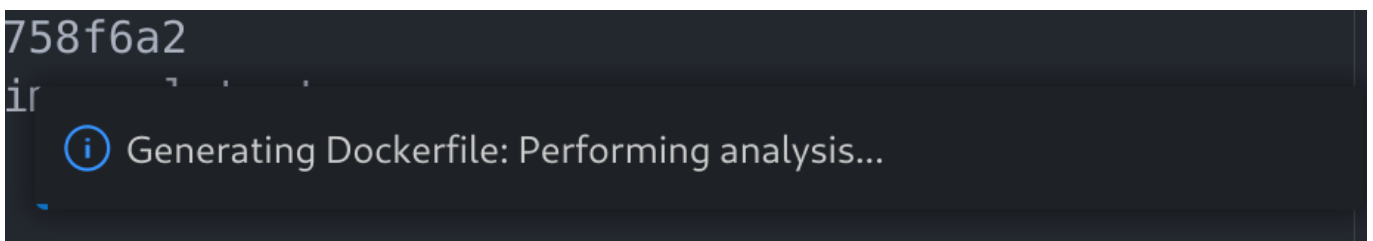
When the extension detects problems that can be repaired, it marks them with a yellow underline (the environment variable problems are processed separately despite using an underline with the same color). Hovering over the underlined region will show additional information.



These problems have corresponding quick fixes that can be applied. Quick fixes can be displayed by clicking the flashlight icon (or the quick fix option in the hover window) or by using the quick fix keyboard shortcut (Ctrl/Cmd + .) while the cursor is in the underlined region. Afterwards, the fix can be applied by clicking it or pressing Enter while the fix is highlighted.



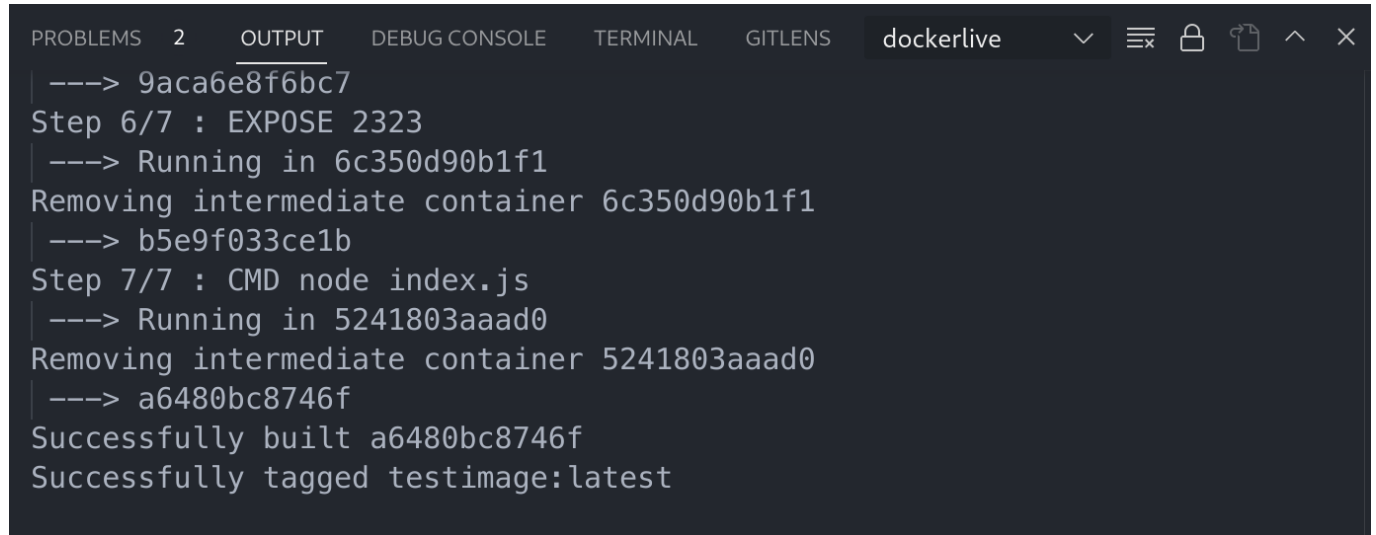
To detect some problems, the extension needs to generate an alternative Dockerfile in the background and compare it with the one being edited. Because of this, the first time a Dockerfile is opened, the extension will automatically start the generation. This process can take some time and the progress can be tracked in the bottom-right corner of the editor. Until this process is finished, no diagnostic information will be shown.



After the generation is complete, all the available repair opportunities will be presented.

## Container log output

The output of the `docker build` command and the output of the container itself are shown in the editor's output pane.

A screenshot of an IDE's output pane showing the logs of a Docker build process. The pane has a dark background with light-colored text. At the top, there is a tab bar with several tabs: 'PROBLEMS', '2', 'OUTPUT' (which is selected), 'DEBUG CONSOLE', 'TERMINAL', 'GITLENS', and 'dockerlive'. To the right of the tabs are several icons: a dropdown arrow, a list icon, a lock icon, a copy icon, an up arrow, and a close icon. The output text shows the following steps:

```
---> 9aca6e8f6bc7
Step 6/7 : EXPOSE 2323
---> Running in 6c350d90b1f1
Removing intermediate container 6c350d90b1f1
---> b5e9f033ce1b
Step 7/7 : CMD node index.js
---> Running in 5241803aaad0
Removing intermediate container 5241803aaad0
---> a6480bc8746f
Successfully built a6480bc8746f
Successfully tagged testimage:latest
```