Secrets
ABAP
Apex
AzureResourceManager
C
C#
C++
CloudFormation
COBOL
CSS
Dart
Docker
Flex
Go
HTML
Java
JavaScript
JCL
Kotlin
Kubernetes
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript
T-SQL
VB.NET
VB6
XML

# Docker static code analysis

Unique rules to find Vulnerabilities, Security Hotspots, and Code Smells in your DOCKER code

All rules 44 | Vulnerability 4 | Bug 4 | Security Hotspot 15 | Code Smell 21

Tags | Impact | Clean code attribute | Search by name...

"WORKDIR" instruction should be used instead of "cd" commands
Code Smell

Specific version tag for image should be used
Code Smell

Package update should not be executed without installing it
Code Smell

Cache should be cleaned after package installation
Code Smell

Deprecated instructions should not be used
Code Smell

Consent flag should be set to avoid manual input
Code Smell

Environment variables should not be unset on a different layer than they were set
Code Smell

Expanded filenames should not become options
Code Smell

Double quote to prevent globbing and word splitting
Code Smell

Instructions should be upper case
Code Smell

Allowing non-root users to modify resources copied to an image is security-sensitive
Security Hotspot

## "WORKDIR" instruction should be used instead of "cd" commands

Analyze your code

Intentionality - Clear | Maintainability

Code Smell | Major

Why is this an issue? | How can I fix it? | More Info

In Dockerfile, instructions `RUN`, `CMD`, and `ENTRYPOINT` can contain long shell scripts chaining multiple commands, including the `cd` command for changing directories. Using `WORKDIR` instruction instead reduces the complexity of the above instructions and makes them easier to read, understand, troubleshoot, and maintain.

### What is the potential impact?

The Dockerfile instructions like `RUN` allow users to execute longer scripts. See the following example:

```
RUN cd /tmp && \
  git clone myrepository.com/MyOrganization/project && \
  cd project && \
  make && \
  cd /app/bin
```

In this example, the first `cd /tmp` command can be replaced by `WORKDIR /tmp` before `RUN`. The last `cd /app/bin` command can be replaced with `WORKDIR /app/bin` after the `RUN` instruction. The result will be the following:

```
WORKDIR /tmp
RUN git clone myrepository.com/MyOrganization/project && \
  cd project && \
  make
WORKDIR /app/bin
```

Those actions will reduce the length of the `RUN` instruction, which makes it easier to read and understand. Sometimes, it is hard to avoid the usage of `cd` command, especially in the middle of a long script. Removing them from the beginning and end of a multi-line is an easy improvement. Additionally, many commands work well with absolute paths, so changing directories can be avoided in most cases.

The `WORKDIR` instruction can be used multiple times in a Dockerfile. It changes the current directory for the next instructions and until there is a following change. This approach simplifies understanding of what is a current directory.

The same principles apply to `CMD` and `ENTRYPOINT` instructions.

This recommendation provides a clear structure for Dockerfiles, making it easier to maintain.

Available In:
sonarlint | sonarcloud | sonarqube