





































-  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 only be used with absolute path
 Code Smell
Too long RUN instruction should be split into multiple lines
 Code Smell
Prefer Exec form for ENTRYPOINT and CMD instructions
 Code Smell
"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

WORKDIR instruction should only be used with absolute path

Analyze your code

- Consistency - Conventional
- Reliability 
-  Code Smell
-  Major 

WORKDIR instructions should be used with an absolute path for clarity and reliability.

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

The WORKDIR instruction defines the working directory for any RUN, CMD, ENTRYPOINT, COPY, and ADD instructions that follow in the Dockerfile. Using absolute paths helps to avoid any confusion or unexpected behavior that could arise, for example after changing the current directory in one of the previous instructions.

Available In:

  | 

