





































-  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... 🔍

<div>Access variable which is not available in the current scope</div> <div> Bug</div>
<div>A space before the equal sign in key-value pair may lead to unintended behavior</div> <div> Bug</div>
<div>Allowing downgrades to a clear-text protocol is security-sensitive</div> <div> Security Hotspot</div>
<div>Allowing shell scripts execution during package installation is security-sensitive</div> <div> Security Hotspot</div>
<div>Using host operating system namespaces is security-sensitive</div> <div> Security Hotspot</div>
<div>Setting loose POSIX file permissions is security-sensitive</div> <div> Security Hotspot</div>
<div>Reduce the amount of consecutive RUN instructions</div> <div> Code Smell</div>
<div>Prefer COPY over ADD for copying local resources</div> <div> Code Smell</div>
<div>WORKDIR instruction should only be used with absolute path</div> <div> Code Smell</div>
<div>Too long RUN instruction should be split into multiple lines</div> <div> Code Smell</div>
<div>Prefer Exec form for ENTRYPOINT and CMD instructions</div> <div> Code Smell</div>

Access variable which is not available in the current scope

Analyze your code

- Intentionality - Logical
- Reliability ⬆

 Bug  Major ⓘ

The variable is not available in the current scope. It will be evaluated to an empty value.

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

The variables defined by ARG instruction have a scope from the definition to the end of the build stage where it was defined. If it was defined in the beginning of the Dockerfile (outside of any build stage), then its scope is restricted to only FROM instructions. Outside of their scope, variables will be resolved to empty string which may lead to unintended behaviour.

Available In:

sonarlint  | sonarcloud  | sonarqube 

