





































-  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... 🔍

Allowing non-root users to modify resources copied to an image is security-sensitive		Security Hotspot
Automatically installing recommended packages is security-sensitive		Security Hotspot
Running containers as a privileged user is security-sensitive		Security Hotspot
Delivering code in production with debug features activated is security-sensitive		Security Hotspot
Use ADD instruction to retrieve remote resources		Code Smell
Arguments in long RUN instructions should be sorted		Code Smell
Track uses of "TODO" tags		Code Smell
Descriptive labels are mandatory		Code Smell
Use digest to pin versions of base images		Code Smell
Dockerfile parsing failure		Code Smell
Pulling an image based on its digest is security-sensitive		Security Hotspot

Use ADD instruction to retrieve remote resources

Analyze your code

- Consistency - Conventional
- Maintainability ⬆

 Code Smell  Minor ⓘ

In Dockerfiles, a common use case is downloading remote resources to use during the build. This is often done using third-party tools inside the image, like `wget` or `curl`. However, this practice can lead to inefficient use of Docker’s build cache and unnecessary complexity. The `ADD` instruction is a built-in feature of Docker that is specifically designed for this purpose, making it a more efficient and safer choice.

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

Documentation

- Docker Docs - [Building best practices](#)

Available In:

sonarlint  | **sonarcloud**  | **sonarqube** 

© 2008-2024 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE, and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Sonar helps developers write [Clean Code](#).
[Privacy Policy](#) | [Cookie Policy](#)

