





































-  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

Arguments in long RUN instructions should be sorted

Analyze your code

- Consistency - Conventional
- Maintainability ⬆
- Reliability ⬆
- Security ⬆

 Code Smell  Minor 

In Dockerfiles, when commands within a `RUN` instruction have a lot of arguments, especially those that install system packages, it is important to ensure that the arguments are sorted alphabetically (if the order is not enforced by a command). This practice enhances the readability and maintainability of the code. It allows for easier tracking of modifications and can help prevent potential errors.

- 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](#)

