






































-  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>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>
<div>"WORKDIR" instruction should be used instead of "cd" commands</div> <div> Code Smell</div>
<div>Specific version tag for image should be used</div> <div> Code Smell</div>
<div>Package update should not be executed without installing it</div> <div> Code Smell</div>
<div>Cache should be cleaned after package installation</div> <div> Code Smell</div>
<div>Deprecated instructions should not be used</div> <div> Code Smell</div>
<div>Consent flag should be set to avoid manual input</div> <div> Code Smell</div>
<div>Environment variables should not be unset on a different layer than they were set</div>

### Reduce the amount of consecutive RUN instructions

Analyze your code

- Consistency - Conventional
- Maintainability 

-  Code Smell
-  Major 

Multiplying the number of RUN instructions increases the final image build time and size.

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

Each time a RUN instruction is added, a new layer is introduced in the final image. This has a direct impact on the build time and image size. Chaining commands in a single RUN instruction using && will use a single layer, thereby reducing the number of layers in the image. This practice can make Docker images more efficient and easier to manage. Each layer in a Docker image is essentially a change to the image, like a version control system. So, fewer layers mean fewer changes to track, which simplifies management and troubleshooting.

Available In:

© 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](#)

