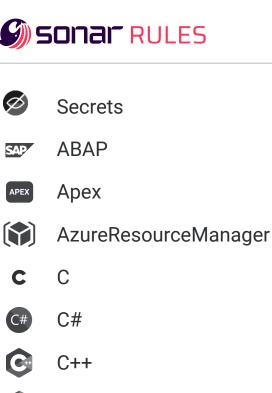
Search by name...





CloudFormation

**Docker** 

**COBOL** 

HTML

Java

**JavaScript** 

Kubernetes

Objective C

PL/SQL

Python

**RPG** 

Terraform

Text

**TypeScript** 

T-SQL

**VB.NET** 

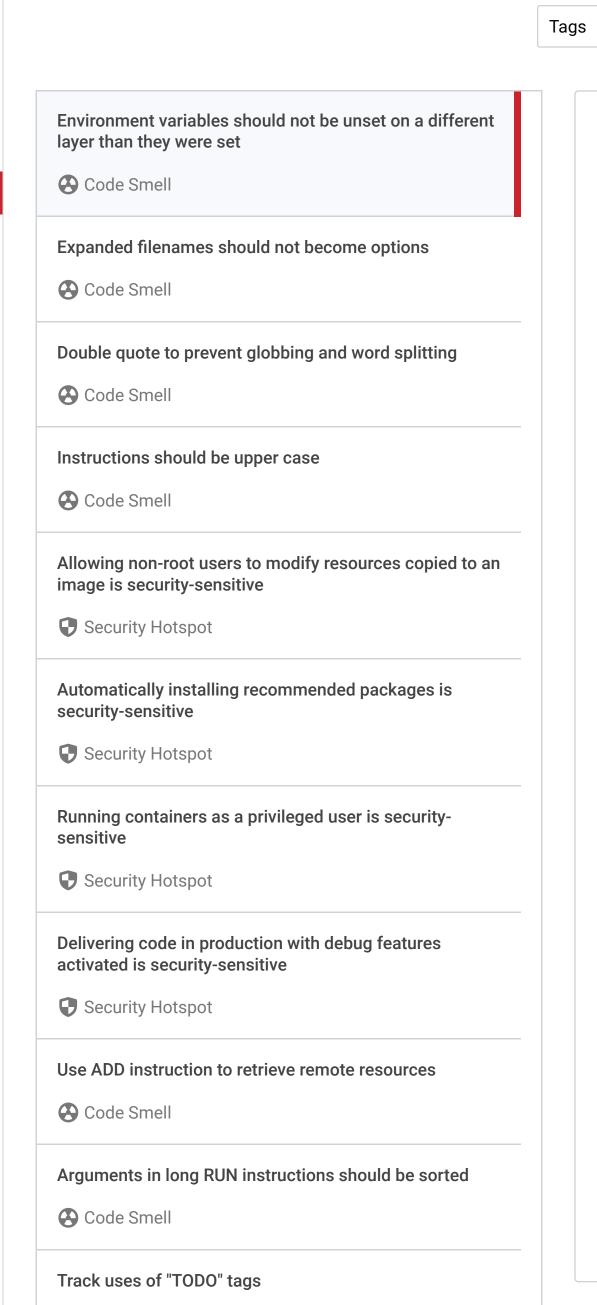


Code Smell

# **Docker static code analysis**

Unique rules to find Vulnerabilities, Security Hotspots, and Code Smells in your DOCKER code







Clean code attribute

**Analyze your code** 

Q

Setting an environment variable using the ENV instruction creates a new layer in the Docker image. The variable is then persisted for all subsequent build stages and is also present in the resulting image. Calling RUN unset <env-variable> unsets the variable only for this particular layer, but it is still possible to dump the environment variable from the previous layer.

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

**Impact** 

Consistency - Conventional Maintainability 🛆

If an environment variable is needed only during build, this variable should be set and unset in a single RUN instruction.

## **Code examples**

#### Noncompliant code example

ENV \$ADMIN USER RUN unset \$ADMIN USER

#### **Compliant solution**

RUN export ADMIN USER="admin" \ / ... && && unset ADMIN\_USER

### How does this work?

In this example, the visibility of ADMIN USER is only limited to the single layer. However, it is still possible to extract the value from the image. The best solution is to use ARG instead of ENV or set and unset the variable in the same RUN instruction.

Available In: 

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

