

- 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... 🔍

Instructions should be upper case
Code Smell
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 digest to pin versions of base images

Analyze your code

Consistency - Conventional Maintainability ⬆ Reliability ⬆ Security ⬆

Code Smell Major ⓘ

In Dockerfiles, it is recommended to use digests to pin versions of base images. This practice ensures that you are always using the exact version of the base image that you intend to use, making your Docker image builds reproducible.

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

Instead of using tags like FROM <image>:<tag>, you should use the digest like so: FROM image@sha256:<digest>. The digest can be retrieved from Docker CLI or looked up in the Docker registry.

Code examples

Noncompliant code example

```
FROM ubuntu:20.04
ENTRYPOINT ["echo", "Hello, World!"]
```

Compliant solution

```
FROM ubuntu:20.04@sha256:0b897358ff6624825fb50d20ffb605ab0eaea77ced0adb8c6a4b756513dec6fc
ENTRYPOINT ["echo", "Hello, World!"]
```

Available In:

sonarlint | sonarcloud | sonarqube

