dockerhub

# sonarqube

🎗 Docker Official Image · ⬇ 1B+ · ⭐ 2.5K

Official images for SonarQube, code analysis tool for code quality and security

`INTEGRATION & DELIVERY`

```
docker pull sonarqube
```
**Copy**

## Overview     Tags

## Quick reference

- **Maintained by:**
  [SonarSource](#) ↗
- **Where to get help:**
  [the SonarSource Community forum](#) ↗, [the Docker Community Forums](#) ↗, [the Docker Community Slack](#) ↗, or [Stack Overflow](#) ↗

## Supported tags and respective `Dockerfile` links

- [2025.1.0-developer](#), [2025.1-developer](#), [2025-lta-developer](#), [developer](#) ↗

### Recent tags

lts-enterprise    lts-developer

lts-datacenter-search    lts-datacenter-app

lts-community    lts    latest    enterprise

developer    datacenter-search

### About Official Images

- `2025.1.0-enterprise` , `2025.1-enterprise` , `2025-lta-enterprise` , `enterprise` ↗
- `2025.1.0-datacenter-app` , `2025.1-datacenter-app` , `2025-lta-datacenter-app` , `datacenter-app` ↗
- `2025.1.0-datacenter-search` , `2025.1-datacenter-search` , `2025-lta-datacenter-search` , `datacenter-search` ↗
- `25.2.0.102705-community` , `community` , `latest` ↗
- `9.9.8-community` , `9.9-community` , `9-community` , `lts` , `lts-community` ↗
- `9.9.8-developer` , `9.9-developer` , `9-developer` , `lts-developer` ↗
- `9.9.8-enterprise` , `9.9-enterprise` , `9-enterprise` , `lts-enterprise` ↗
- `9.9.8-datacenter-app` , `9.9-datacenter-app` , `9-datacenter-app` , `lts-datacenter-app` ↗
- `9.9.8-datacenter-search` , `9.9-datacenter-search` , `9-datacenter-search` , `lts-datacenter-search` ↗

# Quick reference (cont.)

- **Where to file issues:**
  https://github.com/SonarSource/docker-sonarqube/issues ↗

- **Supported architectures:** (more info ↗)
  `amd64` , `arm64v8`

- **Published image artifact details:**
  repo-info repo's `repos/sonarqube/` directory ↗ (history ↗)
  (image metadata, transfer size, etc)

- **Image updates:**
  official-images repo's `library/sonarqube` label ↗
  official-images repo's `library/sonarqube` file ↗ (history ↗)

- **Source of this description:**
  docs repo's `sonarqube/` directory ↗ (history ↗)

# What is `sonarqube` ?

`sonarqube` Docker repository stores the official Sonar images for SonarQube Server and SonarQube Community Build.

[SonarQube Server](#) ⧉ (formerly SonarQube) is an on-premise analysis tool designed to detect quality and security issues in 30+ languages, frameworks, and IaC platforms. The solution also provides fix recommendations leveraging AI with Sonar's AI CodeFix capability. By integrating directly with your CI pipeline or on one of the supported DevOps platforms, your code is checked against an extensive set of rules that cover many attributes of code, such as maintainability, reliability, and security issues on each merge/pull request.

[SonarQube Community Build](#) ⧉ (formerly SonarQube Community) is Sonar's self-managed free offering, released on a monthly schedule. It includes the latest core capabilities available in open source, providing essential features such as bug detection, identification of code smells, and basic security issue analysis across 21 programming languages and frameworks. For advanced security analysis, enterprise-grade integrations, and scalability features, the commercial version, SonarQube Server, is available.

## How to use this image

Here, you'll find the Docker images for the SonarQube Server (Developer Edition, Enterprise Edition, and Data Center Edition), as well as for SonarQube Community Build.

## Docker Host Requirements

Because SonarQube uses an embedded Elasticsearch, make sure that your Docker host configuration complies with the [Elasticsearch production mode requirements](#) ⧉ and [File Descriptors configuration](#) ⧉.

For example, on Linux, you can set the recommended values for the current session by running the following commands as root on the host:

```
sysctl -w vm.max_map_count=524288
sysctl -w fs.file-max=131072
```

```
ulimit -n 131072
ulimit -u 8192
```

## Demo

To quickly run a demo instance, see Using Docker on the [Try Out SonarQube](#) page. When you are ready to move to a more sustainable setup, take some time to read the **Installation** and **Configuration** sections below.

## Installation

> **Multi-platform support**: Starting from SonarQube `9.9` LTS, the docker images support running both on `amd64` architecture and `arm64` -based Apple Silicon (M1).

For installation instructions, see Installing the Server from the Docker Image on the [Install the Server](#) page.

To run a cluster with the SonarQube Server Data Center Edition, please refer to Installing SonarQube Server from the Docker Image on the [Install the Server as a Cluster](#) page.

> The `lts` tag on Docker images is replaced with every new LTS release. If you want to avoid any automatic major upgrades, we recommend using the corresponding `9.9-<edition>` tag instead of `lts-<edition>` .

## Configuration

### Port binding

By default, the server running within the container will listen on port 9000. You can expose the container port 9000 to the host port 9000 with the `-p 9000:9000` argument to `docker run` , like the command below:

```
docker run --name sonarqube-custom -p 9000:9000 sonarqube:community
```

You can then browse to `http://localhost:9000` or `http://host-ip:9000` in your web browser to access the web interface.

## Database

By default, the image will use an embedded H2 database that is not suited for production.

> **Warning:** Only a single instance of SonarQube Server or SonarQube Community Build can connect to a database schema. If you're using a Docker Swarm or Kubernetes, make sure that multiple instances are never running on the same database schema simultaneously. This will cause the SonarQube to behave unpredictably, and data will be corrupted. There is no safeguard, as described on SONAR-10362 ↗. The SonarQube Server Data Center Edition has the same limitation in that only one cluster can connect to one database schema at the same time.

Set up a database by following the "Installing the Database" ↗ section.

## Use volumes

We recommend creating volumes for the following directories:

- `/opt/sonarqube/data` : data files, such as the embedded H2 database and Elasticsearch indexes
- `/opt/sonarqube/logs` : contains SonarQube logs about access, web process, CE process, Elasticsearch logs
- `/opt/sonarqube/extensions` : for 3rd party plugins

> **Warning:** You cannot use the same volumes on multiple instances of SonarQube.

# Upgrading

For upgrade instructions, see Upgrading from the Docker Image on the [Upgrade the Server](#) ↗ page.

## Advanced configuration

### Customized image

In some environments, it may make more sense to prepare a custom image containing your configuration. A `Dockerfile` to achieve this may be as simple as:

```
FROM sonarqube:community
COPY sonar-custom-plugin-1.0.jar /opt/sonarqube/extensions/
```

You could then build and try the image with something like:

```
$ docker build --tag=sonarqube-custom .
$ docker run -ti sonarqube-custom
```

### Avoid hard termination

The instance will stop gracefully, waiting for any tasks in progress to finish. Waiting for in-progress tasks to finish can take a large amount of time, which the docker does not expect by default when stopping. To avoid having the instance killed by the Docker daemon after 10 seconds, it is best to configure a timeout to stop the container with `--stop-timeout`. For example:

```
docker run --stop-timeout 3600 sonarqube
```

## Administration

The administration guide can be found [here](#) ↗.

# License

SonarQube Community Build is licensed under [GNU Lesser General Public License, Version 3.0](#) ⧉. SonarQube Server Developer, Enterprise, and Data Center Editions are licensed under [SonarSource Terms and Condition](#) ⧉.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).

Some additional license information which was able to be auto-detected might be found in [the `repo-info` repository's `sonarqube/` directory](#) ⧉.

As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.

🐳 **docker**

**Why**
Overview
What is a Container

**Products**
Product Overview

**Product Offerings**
Docker Desktop
Docker Hub

**Features**
Container Runtime
Developer Tools
Docker App
Kubernetes

**Developers**
Getting Started
Play with Docker
Community
Open Source
Documentation

**Company**
About Us
Resources
Blog
Customers
Partners
Newsroom
Events and Webinars
Careers
Contact Us
System Status ⧉

Cookies Settings