**Code Coverage**, also called test coverage, is a measure of how much of the application’s code has been run in testing. Essentially, it's a metric that many teams use to check the quality of their tests because it represents the percentage of the production code that has been tested and run.

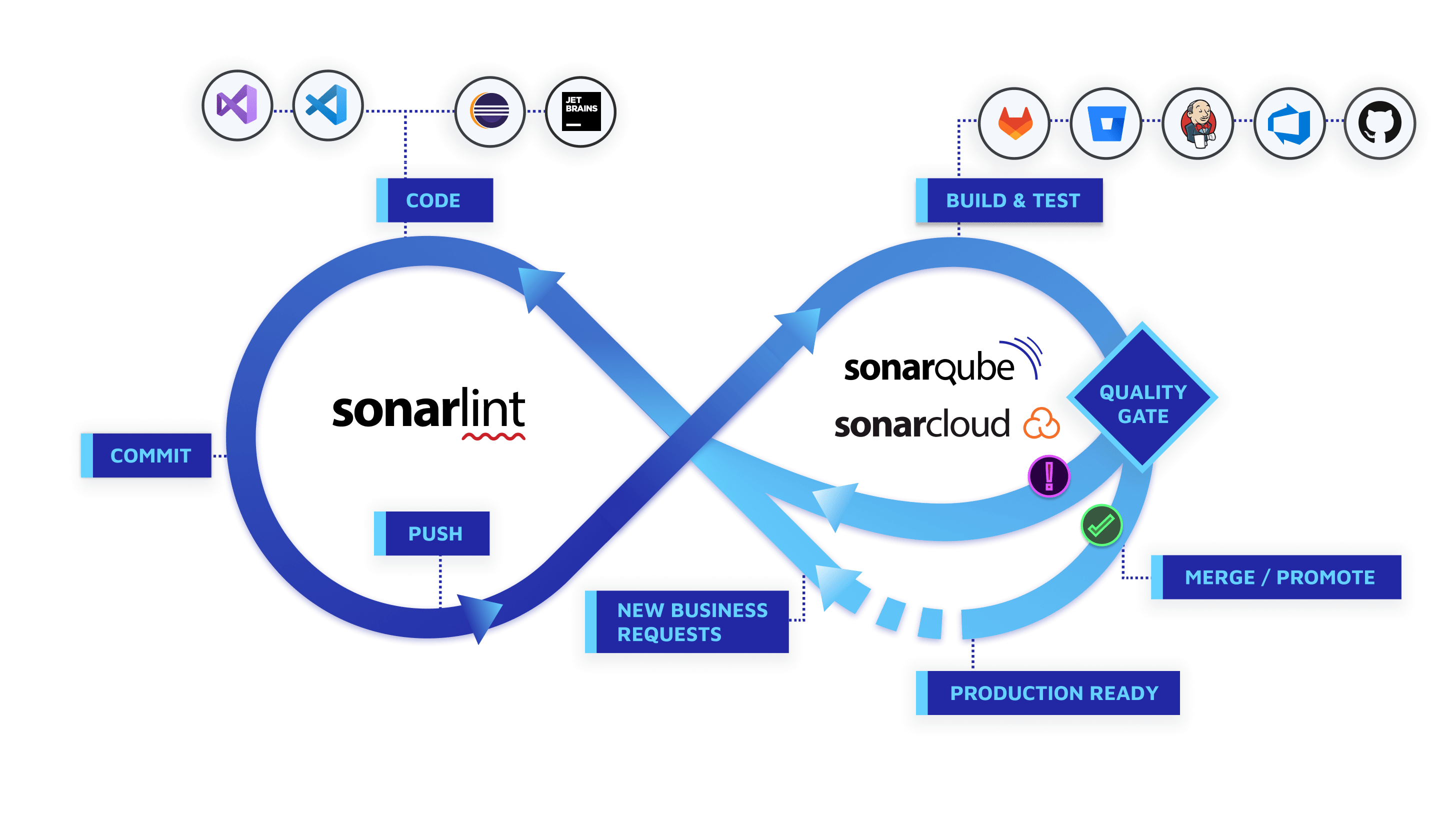
This gives development teams reassurance that their programs have been broadly tested for bugs and should be relatively error-free.

Test coverage reports and test execution reports are important metrics in assessing the quality of your code. Test coverage reports tell you what percentage of your code is covered by your test cases. Test execution reports tell you which tests have been run and their results.

SonarQube itself does not calculate coverage. To include coverage results in your analysis, you must set up a third-party coverage tool and configure SonarQube to import the results produced by that tool.

**SONARQUBE**.

SonarQube is a self-managed, automatic code review tool that systematically helps you deliver clean code. As a core element of our Sonar solution, SonarQube integrates into your existing workflow and detects issues in your code to help you perform continuous code inspections of your projects. The tool analyses 30+ different programming languages and integrates into your CI pipeline and DevOps platform to ensure that your code meets high-quality standards.



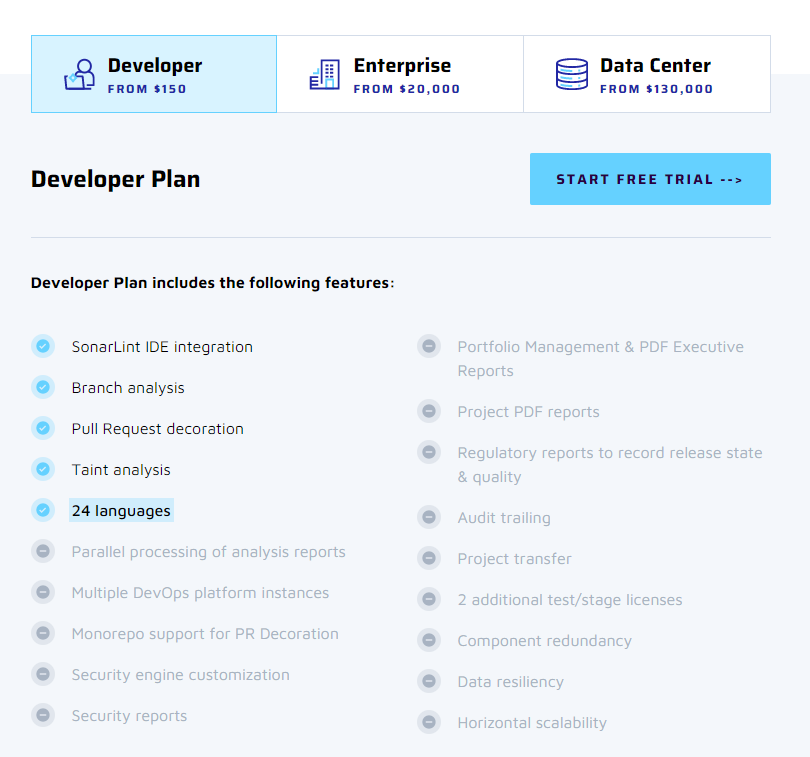
**The Sonar solution performs checks at every stage of the development process:**

* SonarLint provides immediate feedback in your IDE as you write code so you can find and fix issues before a commit.
* SonarQube’s PR analysis fits into your CI/CD workflows with SonarQube’s PR analysis and use of quality gates.
* Quality gates keep code with issues from being released to production, a key tool in helping you incorporate the Clean as You Code methodology.
* The Clean as You Code approach helps you focus on submitting new, clean code for production, knowing that your existing code will be improved over time.

Graphical user interface, application, website

Description automatically generatedSonarQube provides feedback through its UI, email, and in decorations on pull or merge requests (in commercial editions) to notify your team that there are issues to address. Feedback can also be obtained in SonarLint supported IDEs when running in connected mode. SonarQube also provides in-depth guidance on the issues telling you why each issue is a problem and how to fix it, adding a valuable layer of education for developers of all experience levels. Developers can then address issues effectively, so code is only promoted when the code is clean and passes the quality gate.

Graphical user interface, text, application, website

Description automatically generated****

**Graphical user interface, application

Description automatically generated**

**Link : https://www.sonarsource.com/products/sonarqube/downloads/**

**Frequently Asked Questions.**

1. **How are the plans Licensed?**

Commercial Editions (Developer, Enterprise, and Data Center) are priced per instance per year and based on your lines of code (LOC). An instance is an installation of SonarQube. You pay per instance for a maximum number of LOC to be analyzed.

Developer Edition pricing starts at $150/yr for a maximum of 100,000 LOC and can extend to $65K/yr for a maximum of 20M LOC.

Enterprise Edition pricing starts at $20K/yr for a maximum of 1M LOC and can extend to $240K/yr for a maximum of 100M LOC.

1. **How are LOC counted?**

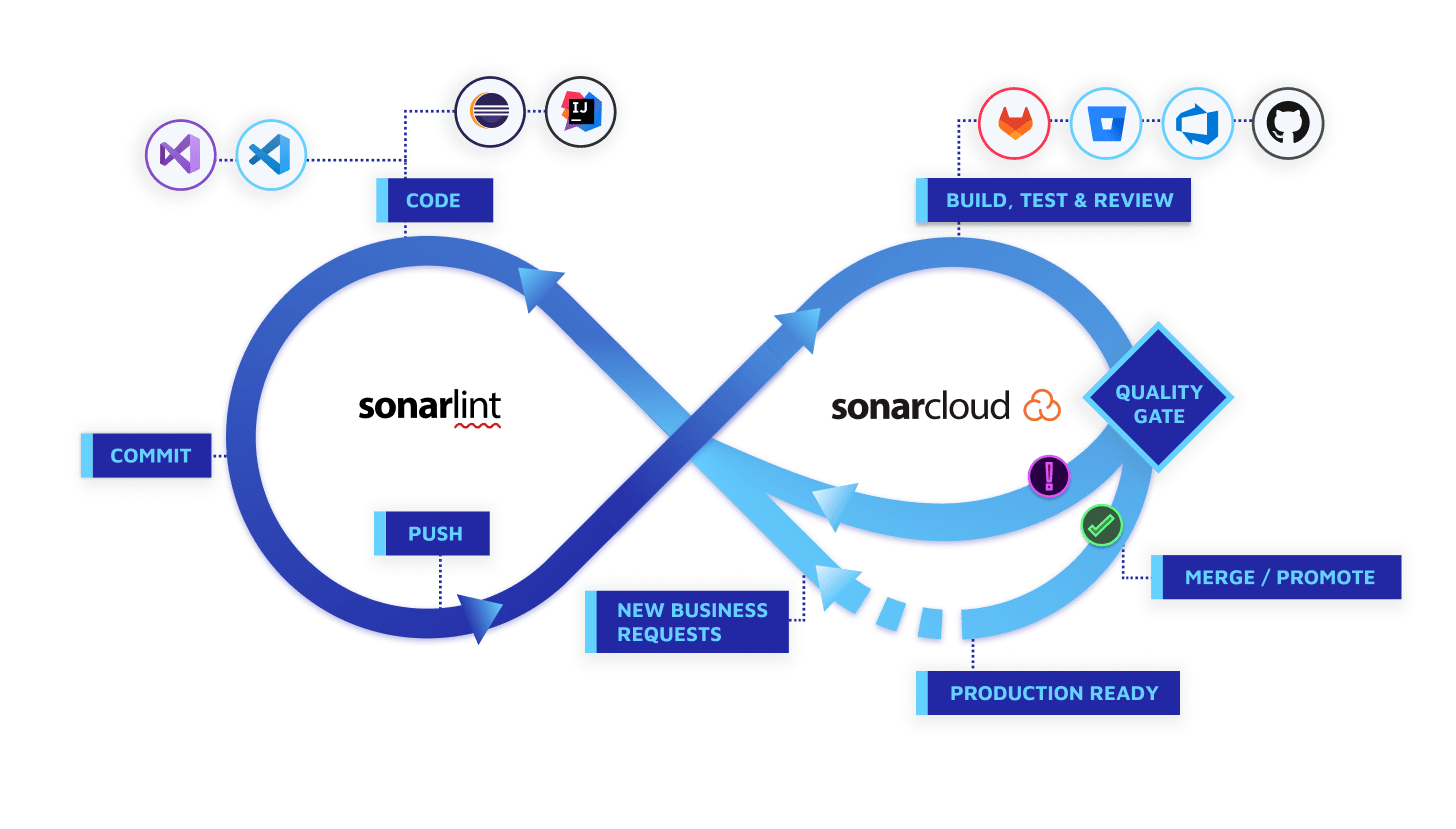
LOCs are computed by summing up the LOCs of each project analyzed. Lines of test code are never included in this number. The LOCs used for a project are the ones found during the most recent analysis of this project. If you start using the branch analysis, then the LOCs of a project will be computed from the project’s largest branch.

1. **What happens if an instance gets close to or reaches the limit of LOCs?**

If you are getting close to the threshold, you will be notified to either upgrade your plan or reduce the number of LOCs in your projects. If you reach the limit, your SonarQube instance will stop accepting new analyses. However, SonarQube will retain basic functionality such as saving configuration changes and allowing project browsing.

**SonarCloud:**

SonarCloud is a cloud-based code analysis service designed to detect coding issues in 26 different programming languages. By integrating directly with your CI pipeline or one of our 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. As a core element of our Sonar solution, SonarCloud completes the analysis loop to help you deliver clean code that meets high-quality standards.



SonarCloud uses state-of-the-art techniques in static code analysis to find problems and potential problems in the code that you and your team write. Static analysis is called static because it does not rely on actually running the code. As a result, SonarCloud offers an additional layer of verification, different from automated testing and manual code review. Its powerful set of language-specific analyzers uses thousands of rules to track down hard-to-find bugs and quality issues - from simple coding mistakes, and tricky bugs, to advanced issues and security vulnerabilities such as injection flaws. Early detection of problems during static analysis ensures that fewer issues get through to the later stages of the process and ultimately helps to increase the overall quality of your production code.

As a core element of our Sonar solution, SonarCloud integrates into your existing workflow and detects issues in your code to help you perform continuous code inspections of your projects. It achieves this by integrating into your CI pipeline or DevOps platform thus, extending your DevOps experience by importing your projects and performing automated code checks within minutes.

**What SonarCloud can do:**

SonarCloud identifies both issues and security hotspots in your code.

**Issues:**

In SonarCloud terminology, an issue is a problem in your code that requires fixing. When scanning for issues, SonarCloud's algorithms are purposely conservative. They are designed to minimize the number of false positives, that is, things wrongly identified as problems. If SonarCloud identifies an issue, you can be quite confident that it really is something that should be fixed. SonarCloud will not overwhelm the developer with false alarms concerning issues.

**Issues are grouped into three types:**

**Bugs:** These are errors in the code that can prevent the program from operating as intended. They affect code reliability.

**Vulnerabilities:** These are problems in the code that could be exploited by a bad actor to compromise the security of the application.

**Code Smells:** These are characteristics of the code that, while not actually preventing the proper functioning of the program, may indicate deeper problems that negatively affect the maintainability of the code. Early identification of these types of issues can help to alleviate technical debt in the application.

**Security hotspots:**

Security hotspots are areas of the code that may cause security issues and therefore need to be reviewed. By design, SonarCloud is more permissive when identifying security hotspots than when identifying vulnerabilities and other issues. An issue is almost always a real problem, while a security hotspot can often be a false alarm (but it is still worth checking). By separating hotspots from issues, SonarCloud maintains the accuracy of its issue detection while still providing developers with useful warnings under the less stringent criteria of the hotspot.

**In the pull request**

Pull requests (on some platforms, called "merge requests") are a mechanism to allow developers to collaborate more effectively. They enable a developer to ask others to review their work (usually their personal feature branch) prior to it being merged into the main body of the code (often the master branch). In the DevOps platform, the pull request is displayed in a dedicated interface that allows the reviewer to see the changes proposed and to either approve or deny the merge.

SonarCloud annotates the pull request interface of the repository service, providing the results of its code analysis on the pull request branch right in the interface and granting or denying approval of the pull request depending on quality gate criteria. In effect, this augments human code review with automatic code review. This feature is often referred to as pull request decoration because it "decorates" the pull request interface with additional information.

**In the codebase**

Code analysis at the IDE and pull request level helps to identify problems before they are merged into the main codebase. However, there are some types of issues and hotspots that can only be found after the code is merged. To find these types of problems, SonarCloud needs to analyze the entire codebase as a single unit and (in the case of some languages) also analyze the results of compiling the code. To do this, SonarCloud offers two approaches: automatic analysis and CI-based analysis.

**Automatic analysis**

With automatic analysis, SonarCloud detects every change to your pull requests or main branch and analyzes the new state of the code in your repository. It uses the same set of analysis methods as CI-based analysis (see below) but it is subject to two restrictions:

* It only works with GitHub (as of today).
* It does not work on repositories that were imported as mono repos into SonarCloud.
* It does not work on all SonarCloud-supported languages. In particular, it does not work with C/C++.

However, if you are using GitHub and the project you imported is in a language that is supported by automatic analysis, then no configuration is needed for analysis to occur so you can start improving your code quality right away. For details, see Automatic Analysis.

As automatic analysis does not work with providers other than GitHub or with certain compiled languages, there are many cases where you will need to configure CI-based analysis instead.

**CI-based analysis**

CI-based analysis refers to the configuration of SonarCloud so that it performs analysis as part of your regular continuous integration (CI) process, in other words, your build process.

To enable CI-based analysis you have to install and configure a piece of software called a scanner. SonarCloud offers scanner extensions and integrations for all of the leading continuous integration (CI) systems used today.

Typically, the scanner is configured to run as part of your continuous integration pipeline so that whenever you push changes to your repository, the scanner is invoked and performs a scan on the code.

The details of how SonarCloud is integrated with your CI/CD process depend on which build tools and the continuous integration system you use. SonarCloud provides custom integrations for the following:

* GitHub Actions
* Bitbucket Pipelines
* Azure Pipelines
* make
* npm
* Maven
* Gradle
* .NET
* Jenkins
* TravisCI
* CircleCI

Additionally, SonarCloud also offers a stand-alone command-line tool (called SonarScanner) that you can install and integrate into your build process manually.

The results of the scan are sent automatically to SonarCloud where they are processed and made available in the dashboard, that is, the SonarCloud interface itself. There you will find all the results of all code analyzed in your repositories. You can sort and filter the results according to a wide range of criteria in order to get a clear picture of the state of your code.

Additionally, the outcome of the SonarCloud analysis (in both automatic and CI-based analyses) can be used to control subsequent build actions such as automatic deployment, etc.

Text

Description automatically generated with medium confidence

|  |  |  |
| --- | --- | --- |
| **LOC** | **Euro/Month** | **INR/Month** |
| 100K | € 10/M | ₹ 897.52/M |
| 250K | € 75/M | ₹ 6731.40/M |
| 500K | € 150/M | ₹ 13462.80/M |
| 1M | € 250/M | ₹ 22438.00/M |
| 2M | € 500/M | ₹ 44876.00/M |
| 5M | € 1500/M | ₹ 134628.00/M |
| 10M | € 3600/M | ₹ 323107.20/M |
| 20M | € 5000/M | ₹ 448760.00/M |

**Frequently Asked Questions.**

1. **How does pricing work for private projects?**

Subscribing to a paid plan on SonarCloud allows you to create a private organization containing private projects. You pay upfront for a maximum number of private lines of code to be analyzed in your organization.

SonarCloud pricing starts at €10/month for a maximum of 100,000 LOC and can extend to €5,000/month for a maximum of 20M LOC.

1. **What payment options are available?**

Payment is done online by credit card and will happen automatically every month, based on the plan you choose. We also accept purchase orders and wire transfer payments for yearly subscriptions of 1M LOCs or more.

1. **What is a Line of Code (LOC) on SonarCloud?**

LOCs are computed by summing up the lines of code of each project analyzed in SonarCloud. The LOCs used for a project are the ones found during the most recent analysis of this project.

1. **How are Lines of Code (LOCs) counted towards billing?**

Only LOCs from your private projects are counted toward your maximum number of LOCs. If your project contains branches, the lines of code computed are the ones from the largest branch. The count is not related to how frequently the source code is analyzed. If your private project has 6K LOCs and you analyze it 100 times in the month, this will be counted as 6K for the billing. If you are getting close to the threshold, you will be notified to either upgrade your plan or reduce the number of LOCs in your projects.

1. **When will I be invoiced?**

You will be invoiced once a month, the day of the month after your trial ends. For example, if you start your free trial on January 1st, it will last till January 14th and you will be first billed on January 15th for your upcoming month, e.g. January 15th to February 15th.

1. **Can I try a private project on SonarCloud for free?**

Your first 14 days are on us. You just have to upgrade your organization to a paid plan and fill your credit card information to get started. After your trial you can continue using SonarCloud and you will be charged for the plan you selected when you first started your free trial. You can cancel anytime.

1. **Can I cancel my subscription?**

Of course! There's no commitment. You can delete your paid organization whenever you wish. Or simply downgrade to the free tier if you wish to keep on analyzing some public projects.

1. **How is the number of lines of code calculated?**

Within your subscription's organization, a line in a file is counted toward the LOC limit only if:

* The file is in a private repository
* The file is not excluded from analysis (see Analysis Scope)
* The line is not a comment or a blank line
* The line is code in a supported language

In short, only code in private projects that is actually analyzed counts toward the LOC limit determined by your subscription plan. You cannot use up your LOC limit by analyzing the same code repeatedly.

1. **Can I still have free public repositories in my private organization?**

Yes. The organization tied to your subscription can contain an unlimited number of public repositories with no size limit, just like in a free public organization. The LOC calculation does not apply to code in public repositories.

Prepare a presentation showing how we are using DevOps to do the process.

Code is prepared in VS and is pushed to azure repo and from their we are building the code in CI using Azure pipelines and the artifacts are deployed to our agents (physical server) and then to various stages like QA & UAT with manual intervention to PROD deployment.

Git repo

Visual studio

CI only builds but no tests are performed.

CD for QA & UAT, prod manually

Test🡪 in other organizations, test scripts are committed in repo, auto testing is happening.

Azure artifacts and their benefits,

What are the other 3 code coverage tools and their pricing, and which is more beneficial.

Sanity testing or acceptance test how can we introduce them into DevOps.

unit testing, integration test with sonar, check can we do it.

Azure monitor 🡪 log analytics , application insights and their charges.

Web apps and their charges and how beneficial are they.

How can we use Azure Key vaults.