

Step 1: Pull the Sonar Docker Image :

```
docker pull sonarqube
```

Step 2: Create a Docker Network

```
docker network create sonar-network
```

Step 3: Start a Database Container

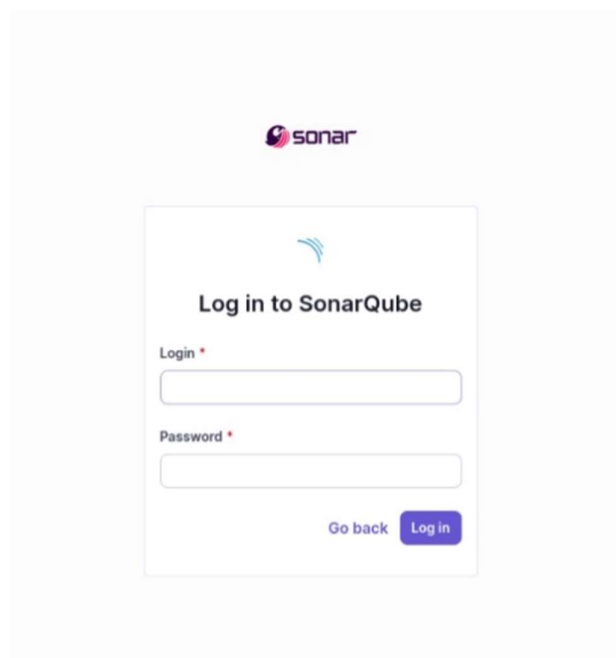
```
docker run -d --name sonar-db --network sonar-network -e POSTGRES_USER=sonar -e POSTGRES_PASSWORD=sonar -e POSTGRES_DB=sonar postgres:9.6
```

Step 4: Start the Sonar Container

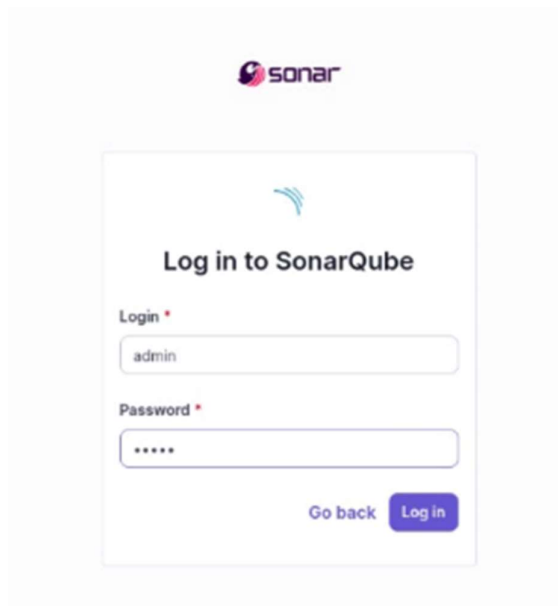
```
docker run -d --name sonar -p 9000:9000 --network sonar-network -e SONARQUBE_JDBC_URL=jdbc:postgresql://sonar-db:5432/sonar -e SONAR_JDBC_USERNAME=sonar -e SONAR_JDBC_PASSWORD=sonar sonarqube
```

Step 5: Access the Sonar Dashboard

Once the Sonar container is running, you can access the Sonar dashboard by opening a web browser and navigating to <http://localhost:9000>

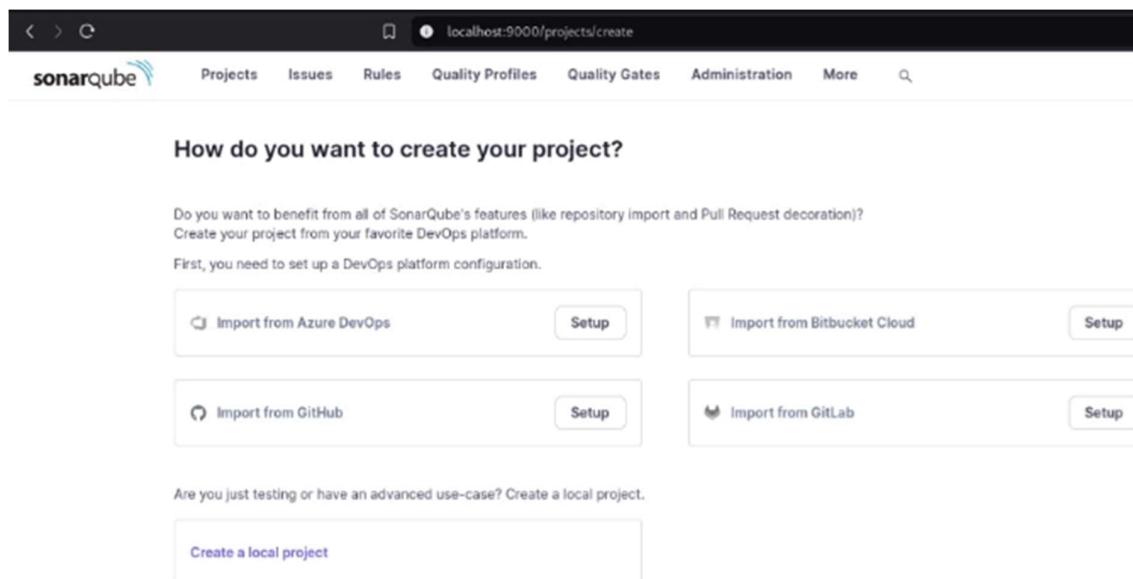


Enter the login and password : as admin admin



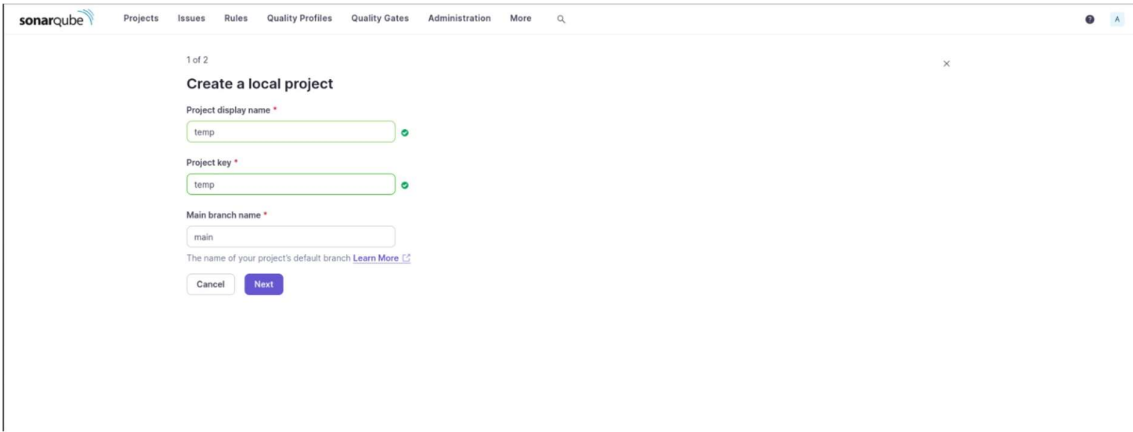
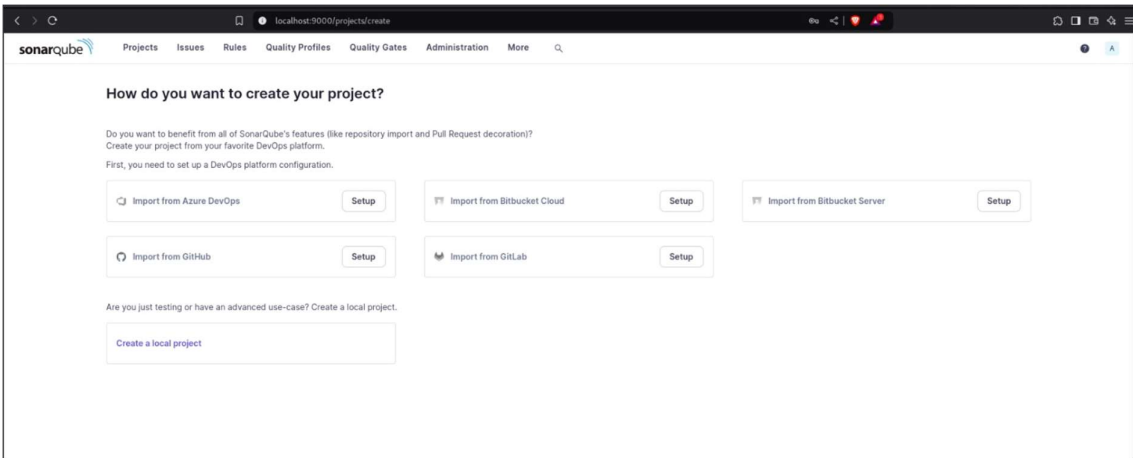
The image shows the SonarQube login page. At the top is the Sonar logo. Below it is a blue wave icon. The title "Log in to SonarQube" is centered. There are two input fields: "Login *" with the text "admin" and "Password *" with masked characters "*****". At the bottom right are two buttons: "Go back" and "Log in".

Update the password :

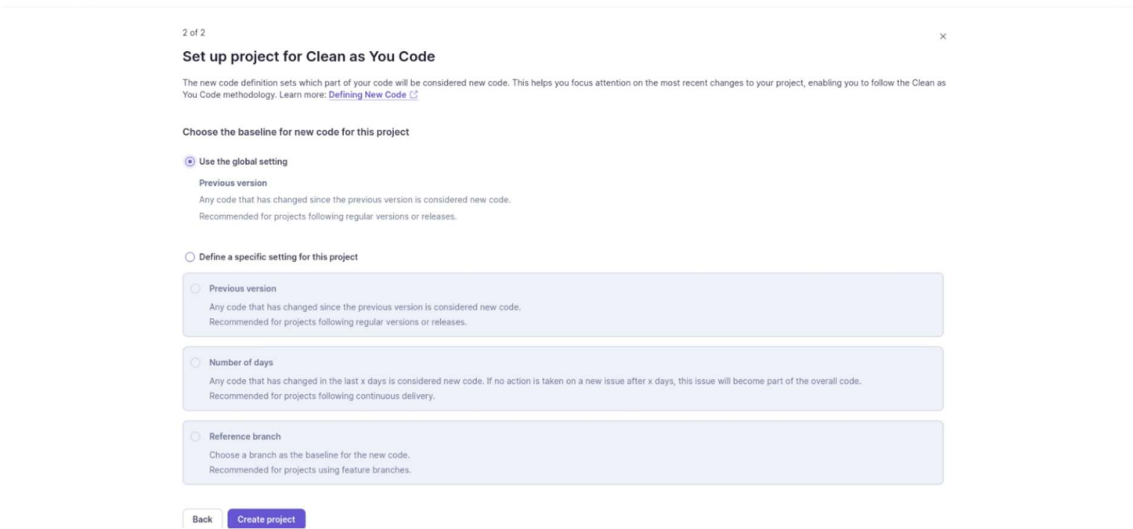


The image shows the SonarQube project creation page. The browser address bar shows "localhost:9000/projects/create". The SonarQube logo is on the left, and a navigation menu with "Projects", "Issues", "Rules", "Quality Profiles", "Quality Gates", "Administration", and "More" is on the right. The main heading is "How do you want to create your project?". Below it is a paragraph: "Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration." There are four buttons arranged in a 2x2 grid: "Import from Azure DevOps", "Import from Bitbucket Cloud", "Import from GitHub", and "Import from GitLab". Each button has a "Setup" link next to it. At the bottom, there is a paragraph: "Are you just testing or have an advanced use-case? Create a local project." and a button labeled "Create a local project".

Click on the **Create a local Project**



Click on **Use global setting** :



Click on **Generate Token** and Select **Other** :

Analysis Method > Locally

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token

Analyze "profile-project": sqp_2d47b52e04b0bb90aef710911fb1fb045a792881

2 Run analysis on your project

What option best describes your project?

MavenGradle.NETOther (for JS, TS, Go, Python, PHP, ...)

Select **other**, Select **Linux** then go to the official url to download the sonar scanner website :

2 Run analysis on your project

What option best describes your project?

MavenGradle.NETOther (for JS, TS, Go, Python, PHP, ...)

What is your OS?

LinuxWindowsmacOS

Download and unzip the Scanner for Linux

Visit the [official documentation of the Scanner](#) to download the latest version, and add the `bin` directory to the `PATH` environment variable

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner \  
-Dsonar.projectKey=profile-project \  
-Dsonar.sources=. \  
-Dsonar.host.url=http://localhost:9000 \  
-Dsonar.token=sqp_2d47b52e04b0bb90aef710911fb1fb045a792881
```

Copy

Please visit the [official documentation of the Scanner](#) for more details.

Download for **Linux x64**

SonarScanner

Issue Tracker

Show more

6.2.1

2024-10-01

FIPS support and improved SSL configuration

Download scanner for: Linux x64 Linux AArch64 Windows x64 macOS x64 macOS AArch64 Docker

Any (Requires a pre-installed JVM)

Release notes

The SonarScanner CLI is the scanner to use when there is no specific scanner for your build system.

The SonarScanner does not yet officially support ARM architecture. Still, early adopters reported it is working fine. If you encounter problems, don't hesitate to share your experience with us on the [SonarQube](#) or [SonarCloud](#) Community Forum but keep in mind that there is no support at this time.

The SonarScanners run on code that is checked out. See [Verifying the code checkout step of your build](#).

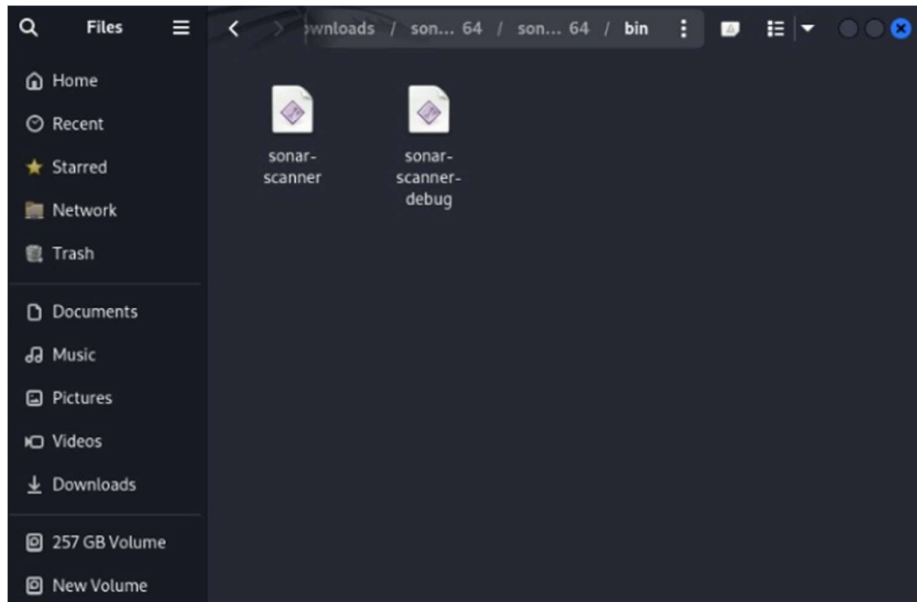
Configuring your project

Create a configuration file in your project's root directory called `sonar-project.properties`.

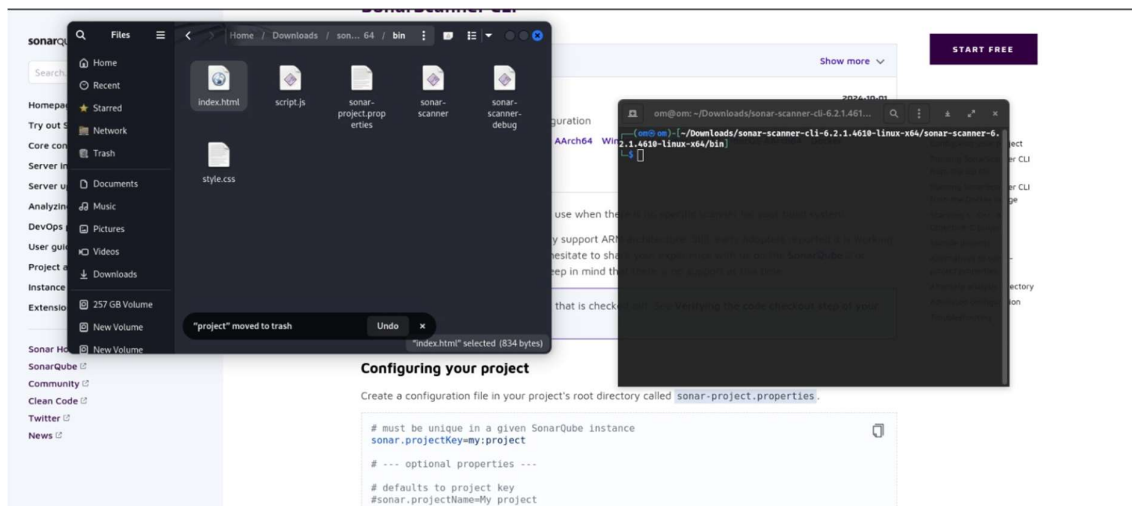
```
# must be unique in a given SonarQube instance  
sonar.projectKey=my:project  
  
# --- optional properties ---  
  
# defaults to project key
```

Extract the zip file in folder :

Then go to the bin directory inside it :



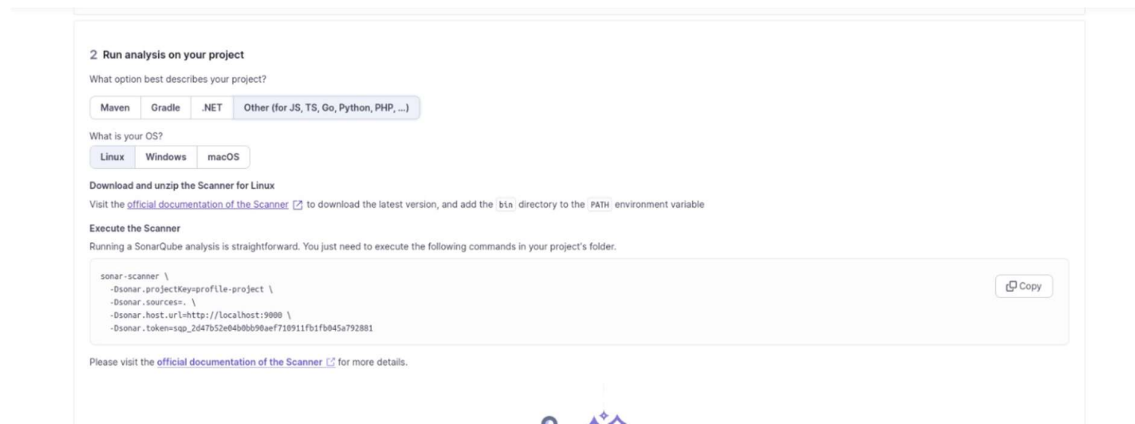
Paste your project files of .html .css .js in same bin folder :



Create the sonar-project.properties file in same folder

```
sonar.projectKey=temp
sonar.host.url=http://localhost:9000
sonar.login=sqp_ce77895c70f118ef6c02ec521398530c6b1e54be
sonar.sourceEncoding=UTF-8
sonar.sources=.
```

and change the project key and sonar login key accordingly from the page



After that open the terminal in the same **bin** directory of sonar scanner

Run command : **./sonar-scanner**

It will check for the issues in the project and generate a report you will get the url for it in terminal :

