Static Code Analysis Setup

For this exercise, you will be setting up SonarQube to analyze your Java project code and SonarScanner to analyze your TypeScript code.

Install SonarQube

We will be using the Docker image to run SonarQube. To install, you must first have Docker running on your computer. You can install Docker Desktop here:

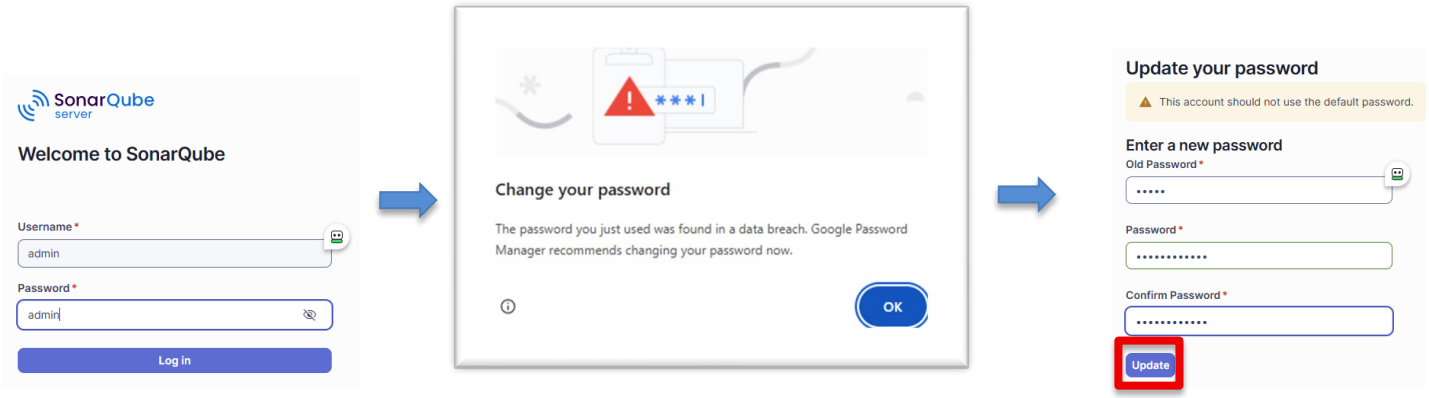
- Windows: https://hub.docker.com/editions/community/docker-ce-desktop-windows
- Mac (manual install): https://docs.docker.com/desktop/mac/install/
- Mac (brew): `brew install --cask docker-desktop`
- Linux - you know what to do (depends on your distribution)

Make sure your Docker Desktop is running. After that open a terminal window and execute (make sure to copy/paste the **entire** command):

```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:26.3.0.120487-community
```

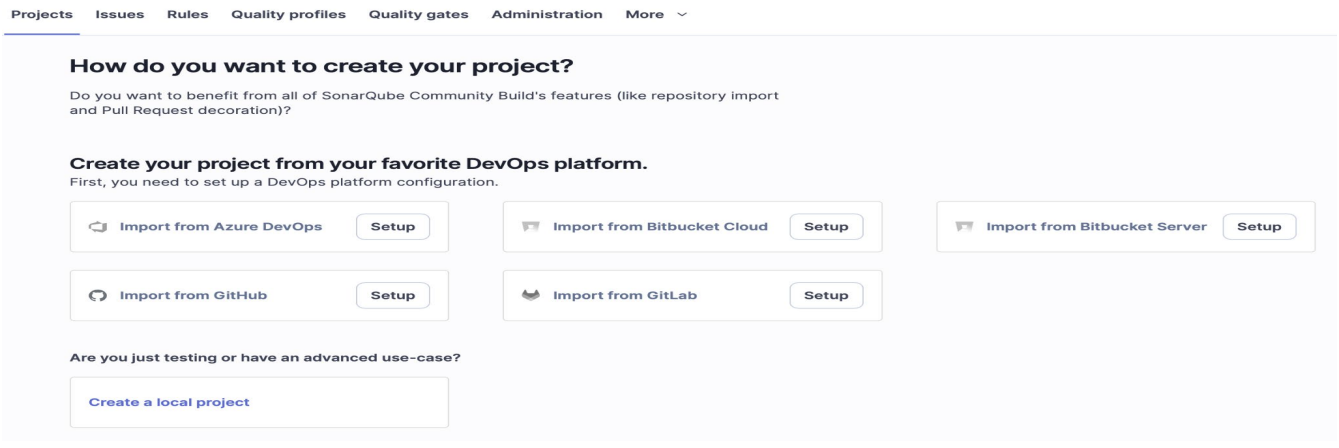
Accessing SonarQube

Once you have SonarQube installed and running, log in to http://localhost:9000 with Administrator credentials (admin/admin) Note: you may get warnings from your browser on the lack of security
Be sure to update your password.

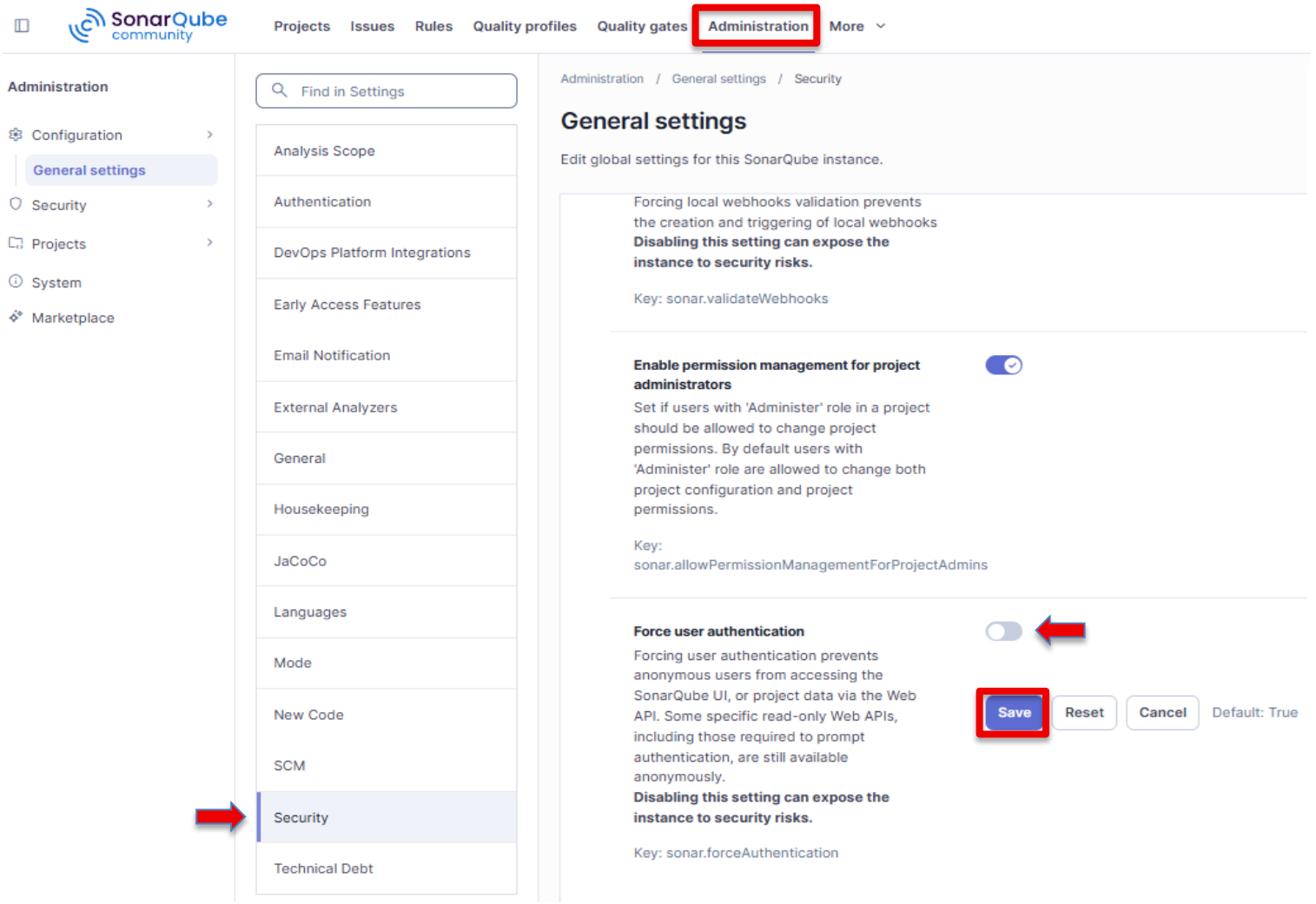


Write down your new password as it may be needed later.

You should see something like this:

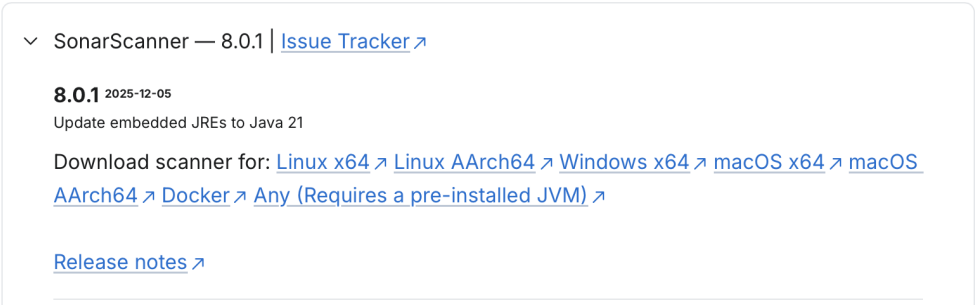


Click **Administration** from the top menu and then scroll down the second menu column to the end and click **Security**. Make sure **Force user authentication** is **unchecked**. This will allow you to freely run SonarQube without requiring any credentials.



At this point, you are done with installing SonarQube.

Install SonarScanner

1. Go to https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/ and download SonarScanner **8.0.1** for your operating system.
2. 

3. Expand the downloaded file into the directory of your choice. We'll refer to it as `[install_directory]` in the next steps.
4. Add the `[install_directory]/bin` directory to your path

- **don't forget to replace `[install_directory]` with the real directory you used**
- for Windows (manual setup): Press Win+S -> search "Environment Variables" -> Click "Edit the system environment variables" -> Click "Environment Variables" -> under "User variables" find "Path" -> Click "Edit" -> Click "New" -> add `[install_directory]\bin` -> click "OK" for all dialogs
- for Windows (PowerShell): `[Environment]::SetEnvironmentVariable("Path", $env:Path + ";[install_directory]\bin", [EnvironmentVariableTarget]::User)`
- for MacOS: `echo 'export PATH="$PATH:[install_directory]/bin"' >> ~/.zshrc && source ~/.zshrc`
- for Linux: you know what to do (depends on the shell you use)

5. Verify your installation by opening a **new shell** and executing the command `sonar-scanner -h`
6. You should get output like this:

```
usage: sonar-scanner [options]

Options:
 -D,--define <arg>     Define property
 -h,--help             Display help information
 -v,--version          Display version information
 -X,--debug            Produce execution debug output
```

Stopping SonarQube

To stop SonarQube on Docker, you must first find the **container id** by typing `docker ps`

```
CONTAINER ID   IMAGE                          COMMAND                  CREATED             STATUS          PORTS                                       NAMES
e6983b096453   sonarqube:26.3.0.120487-community   "/opt/sonarqube/dock…"   About an hour ago   Up 48 minutes   0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp   sonarqube
```

1. Next, practice how to stop the container by typing: **`docker stop`** *`[container id]`*

To start the container again, you need to execute

2. **`docker ps -a`** to recall the container id
   (**`docker ps`** will show only currently **running** containers)
3. **`docker start`** *`[container id]`*

If you execute **`docker rm`** *`[container id]`*, all the changes you introduced will be removed along with the container, and you will need to start from scratch.

Cleaning Up

In the future, when you are done **with the project**, you can clean up the space by executing

1. **`docker ps -a`** (to recall the container id)
2. **`docker stop`** *`[container id]`* (to stop the container)
3. **`docker rm`** *`[container id]`* (to delete the container)
4. **`docker image ls`** (to discover the image id)
5. **`docker image rm`** *`[image id]`* (to delete the image)