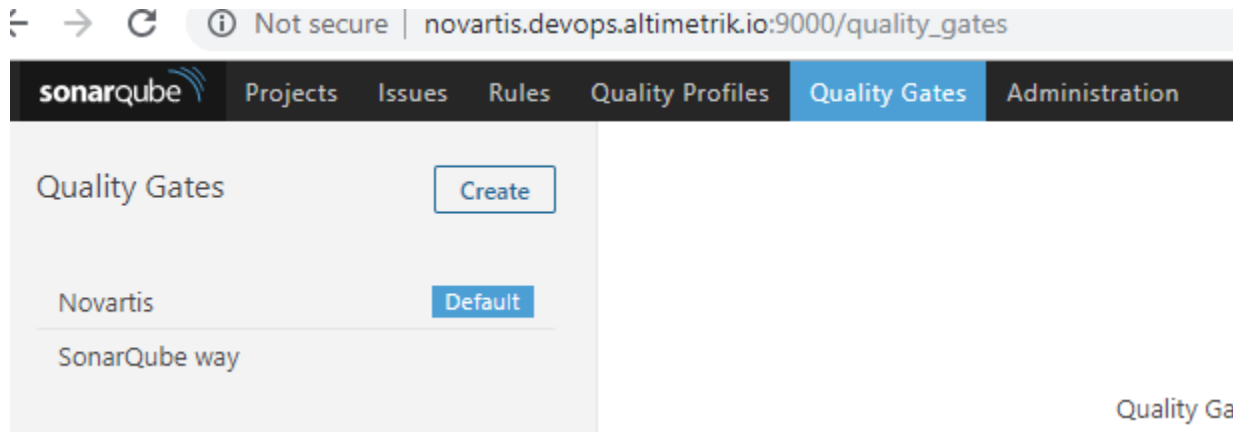


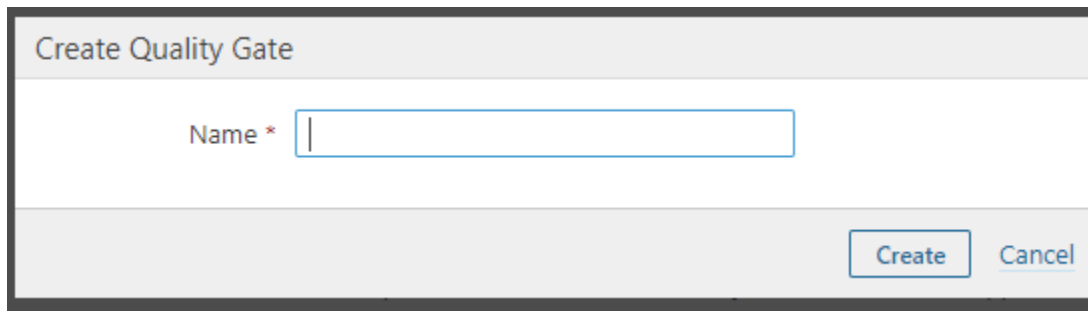
How to - Sonar Configuration for code check gates

Create quality gates

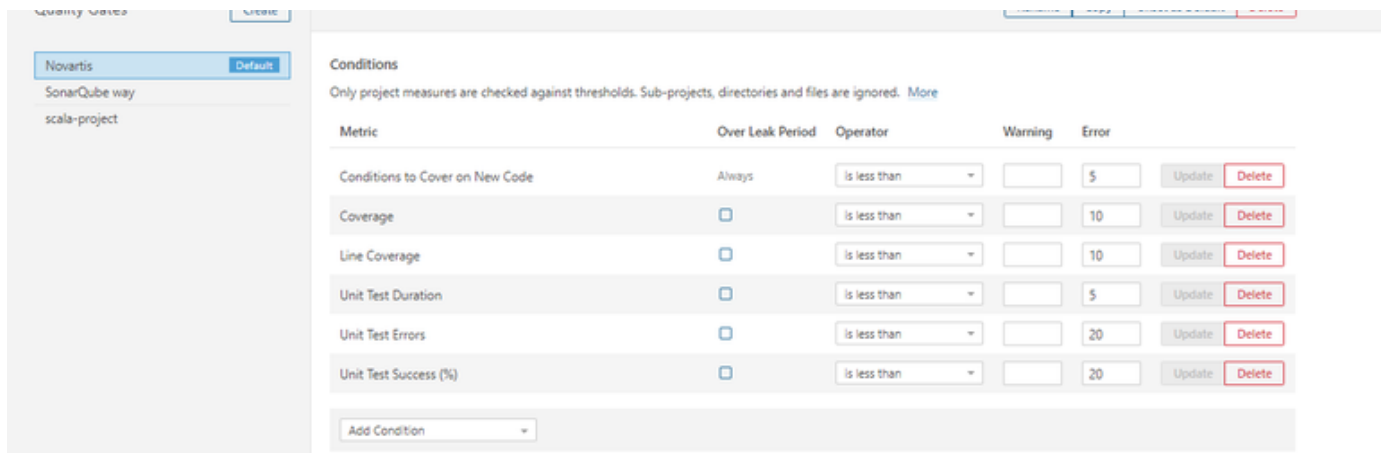
1. Go to sonar dashboard >Quality gates



Click on create option give the name



Add conditions and save it.



Metric	Over Leak Period	Operator	Warning	Error	Update	Delete
Conditions to Cover on New Code	Always	is less than		\$		
Coverage	<input type="checkbox"/>	is less than		10		
Line Coverage	<input type="checkbox"/>	is less than		10		
Unit Test Duration	<input type="checkbox"/>	is less than		\$		
Unit Test Errors	<input type="checkbox"/>	is less than		20		
Unit Test Success (%)	<input type="checkbox"/>	is less than		20		

Quality Profiles:

Quality Profiles are collections of rules to apply during an analysis. For each language there is a default profile.

All projects not explicitly assigned to some other profile will be analyzed with the default.

1. Create new quality profile.

sonarqube Projects Issues Rules **Quality Profiles** Quality Gates Administration

Quality Profiles

Quality Profiles are collections of rules to apply during an analysis.
For each language there is a default profile. All projects not explicitly assigned to some other profile will be analyzed with the default.

[Create](#)

All Profiles ▾

C#, 1 profile(s)	Projects	Rules	Updated	Used
Sonar way Built-in	Default	186	Never	Never ▾

Flex, 1 profile(s)	Projects	Rules	Updated	Used
Sonar way Built-in	Default	60	Never	Never ▾

Recently Added Rules

- Failed unit tests should be fixed
Scala, not yet activated
- Skipped unit tests should be either removed or fixed
Scala, not yet activated
- Source files should not have any duplicated blocks
Scala, not yet activated
- Source files should have a sufficient density of com...
Scala, not yet activated
- Lines should have sufficient coverage by tests
Scala, not yet activated

Click on create option. Give the name and language

Quality Profiles

Quality Profiles are collections of rules to apply during an analysis.
For each language there is a default profile. All projects not explicitly assigned to some other profile will be analyzed with the default.

[Create](#)

All Profiles ▾

C#, 1 profile(s)

[Sonar way](#) [Built-in](#)

New Profile

Name*

Language*

[Create](#) [Cancel](#)

Recently Added Rules

- Failed unit tests should be fixed
Scala, not yet activated
- Skipped unit tests should be either removed or fixed
Scala, not yet activated
- Source files should not have any duplicated blocks
Scala, not yet activated

sonarqube Projects Issues Rules **Quality Profiles** Quality Gates Administration

Quality Profiles / Python

Novartis-test

Updated: Never [Used: Never](#) [Changelog](#) [Actions ▾](#)

Rules	Active	Inactive
Total	0	240
Bugs	0	15
Vulnerabilities	0	1
Code Smells	0	224

[Activate More](#)

Sonar way rules not included ⓘ 32

Permissions

Users with the global "Manage Quality Profile" permission can manage this quality profile.

[Grant permissions to more users](#)

Inheritance

Novartis-test 0 active rules 0 overridden rules

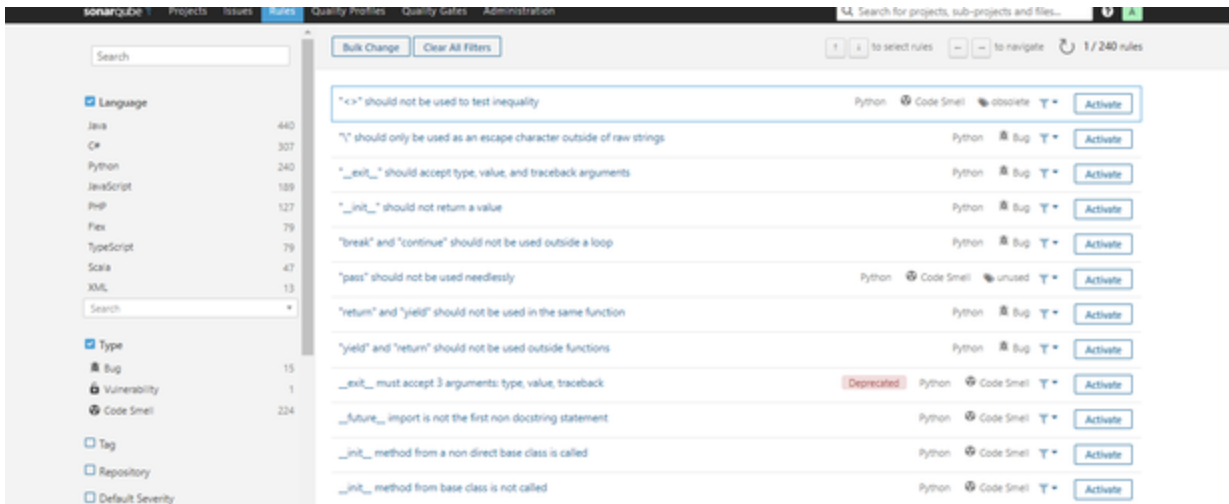
[Change Parent](#)

Projects

No projects are explicitly associated to the profile.

[Change Projects](#)

To activate more rules click on Activate more option.



And then Activate rules for newly created quality profile.

Quality Profiles / Scala

Sonar way Built-in Updated: Never Used: 2 days ago Changelog Actions

This quality profile is provided by a plugin.
It will automatically be updated when a new version of the supplying plugin changes its definition.

Rules	Active	Inactive
Total	29	18
Bugs	5	2
Vulnerabilities	2	0
Code Smells	22	16

Inheritance

Sonar way Built-in 29 active rules 0 overridden rules

Projects

Default Every project not specifically associated with a quality profile will be associated to this one by default.

Python, 1 profile(s)		Projects	Rules	Updated	Used
Sonar way	Built-in	Default	32	Never	Never ▼

Scala, 2 profile(s)		Projects	Rules	Updated	Used
Sonar way	Built-in	Default	29	Never	2 days ago ▼
scala-test		1	0	Never	24 minutes ago ▼

How to activate and deactivate rules for custom rules

By default, when entering the top menu item "Rules", you will see all the available rules brought by the analyzers installed on your SonarQube instance. You have the ability to narrow the selection based on search criteria in the left pane:

Language: the language to which a rule applies.

Type: Bug, Vulnerability, Code Smell or Security Hotspot rules.

Tag: it is possible to add tags to rules in order to classify them and to help discover them more easily.

Repository: the engine/analyzer that contributes rules to SonarQube.

Default Severity: the original severity of the rule - as defined by the analyzer that contributes this rule.

Status: rules can have 3 different statuses:

Deprecated: The rule should no longer be used because a similar, but more powerful and accurate rule exists.

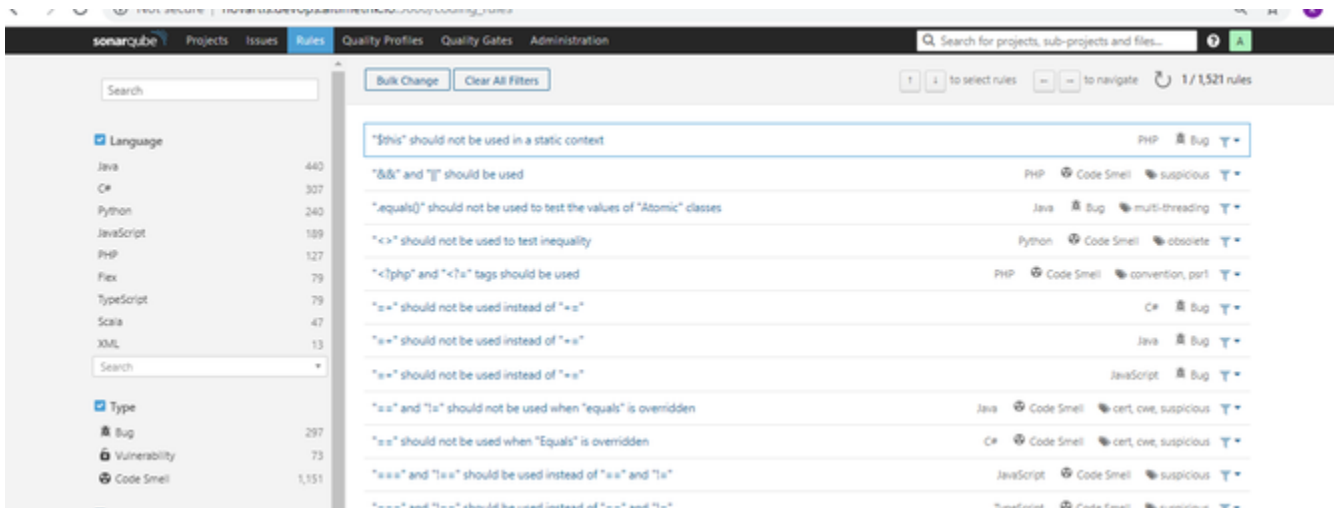
Ready: The rule is ready to be used in production.

Available: date when a rule was first added on SonarQube. This is useful to list all the new rules since the last upgrade of a plugin for instance.

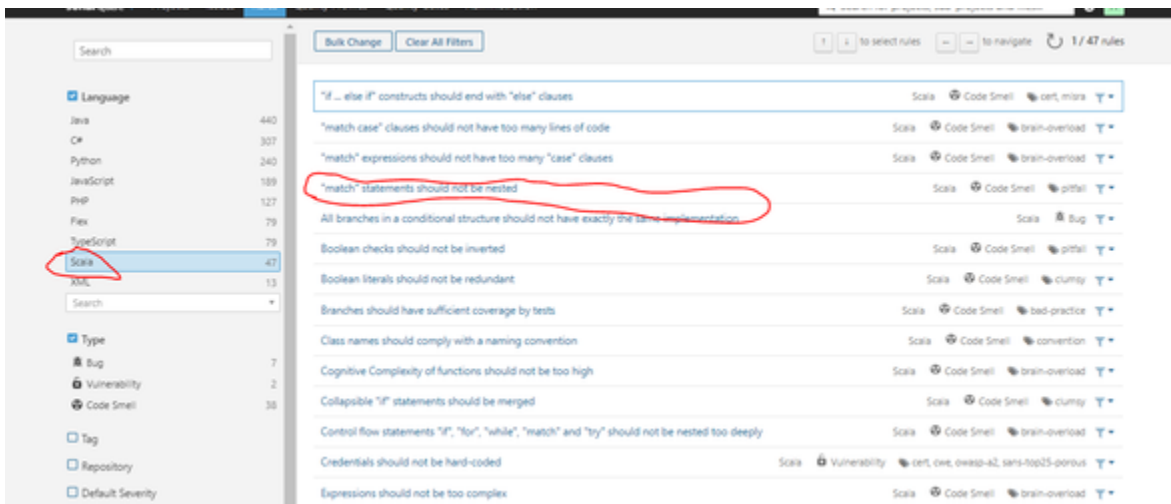
Template: display rule templates that allow to create custom rules (see later on this page).

Quality Profile: inclusion in or exclusion from a specific profile

Go to sonar Dashboard select rules option



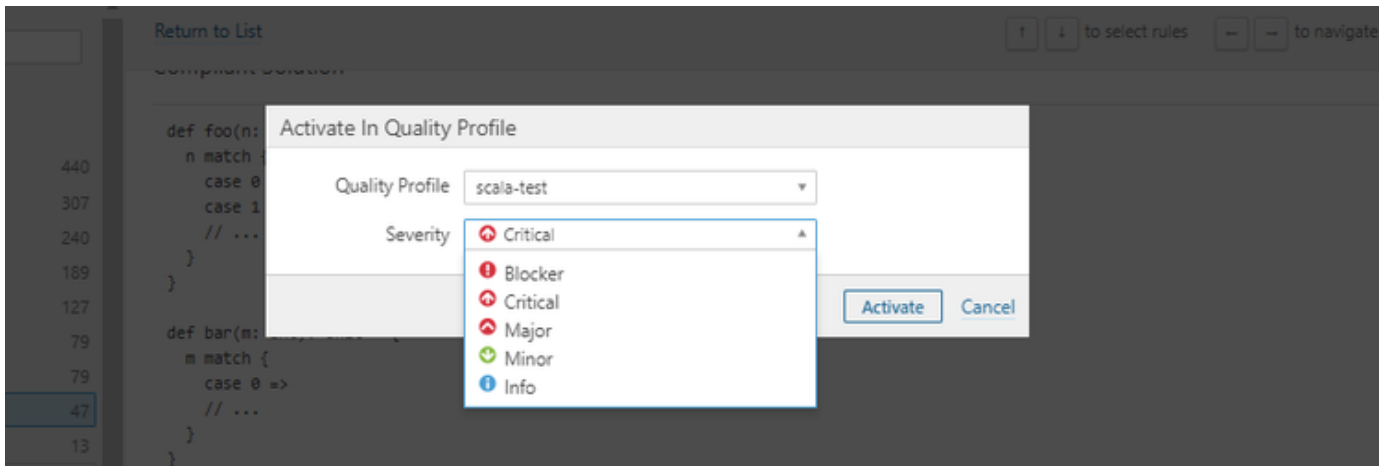
Select Scala and click on any rules



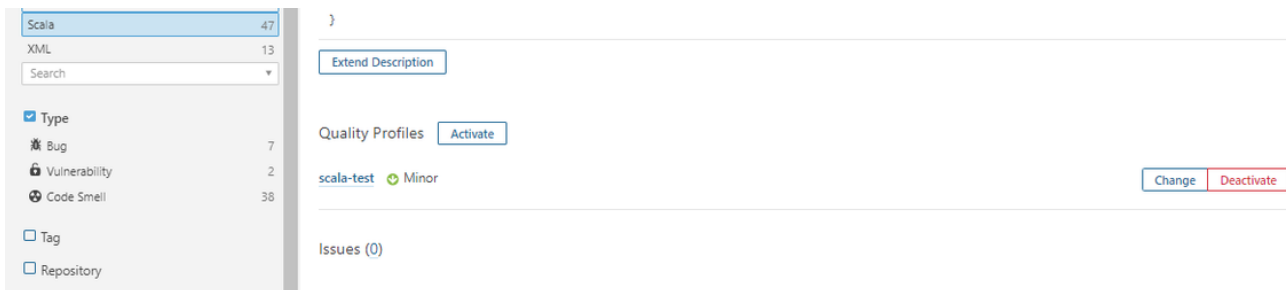
Click on Activate the Rule option



Once you click on Activate select the severity level(Blocker,Critical,Minor,Info)



Click on Activate option



To deactivate the rule click on deactivate option.

Administration Items:

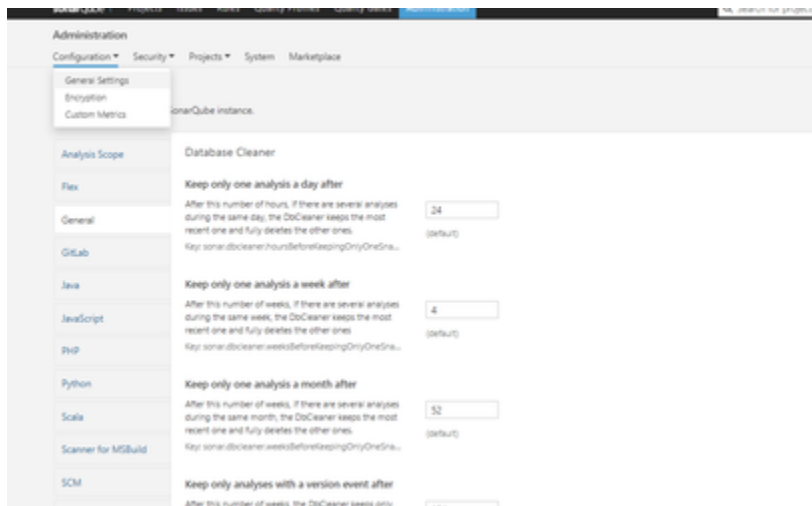
Project administration is accessible through the Administration menu of each project.

Administration tab contains

1. Configuration - Configure the plugin related to java,scala,python,junit etc.
2. security -Configure at a group or user level and giving the permissions to user or group
3. project - It will shows background tasks running for project.
4. system - It showst system related info
5. Marketplace - Download the plugins

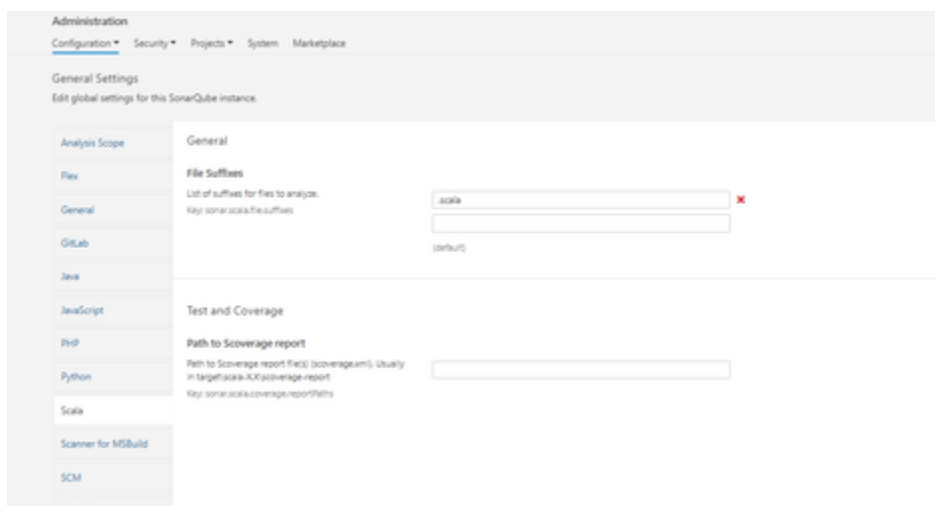
Configuration :

Go to Administration> Configuration > General Setting



Configuration for scala plugin

Administration> Configuration > General Setting > Scala

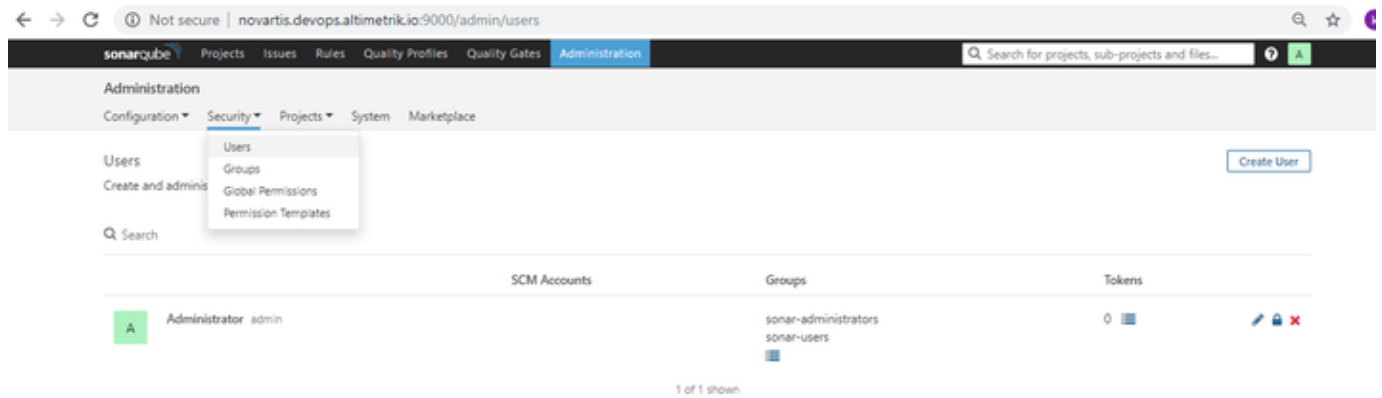


Define coverage path under **Test and Coverage** , shows the coverage report for scala project .

Security :

How to create user :

Administrator > security > Users



Click on create user and give the name and password

The screenshot shows the 'Create User' dialog box. It has a title bar 'Create User'. The form contains the following fields: 'Login*' with the value 'Tester' and a note 'Minimum 3 characters'; 'Name*' with the value 'test'; 'Email' (empty); and 'Password*' with masked characters. There is an 'Add' button next to the 'SCM Accounts' label. A note at the bottom states 'Login and email are automatically considered as SCM accounts'. At the bottom right, there are 'Create' and 'Cancel' buttons.

Project :

Projects Management :its used delete multiple projects at once or to provision projects if you would like to configure them before the first analysis.

Administration

Configuration ▾ Security ▾ Projects ▾ System Marketplace

Projects Management

Use this page to delete multiple projects at once, or to provision projects if you would like to configure them before the first analysis. Note that once a project is provisioned, you have access to perform all project configurations on it.

Default visibility of new projects: Public [Create Project](#)

☐ Last analysis before ☐ Only Provisioned [?](#) Search

[Bulk Apply Permission Template](#) [Delete](#)

Name	Key	Last Analysis	
<input type="checkbox"/> demo-hello	demo-hello	October 1, 2019	Actions ▾
<input type="checkbox"/> SonarDemo-Scala	SonarDemo	September 24, 2019	Actions ▾
<input type="checkbox"/> sonartestproject	samp	October 3, 2019	Actions ▾

3 of 3 shown

To create project click on create project and give the project name and key.

Create Project

Name*

Key*

Visibility ☒ Public ☐ Private

[More](#)

[Create](#) [Cancel](#)

Once you click on create option it will create project successfully.

Background Tasks:Monitoring of the queue of tasks running asynchronously on the server. It also gives access to the history of finished tasks.