Sonarqube v7.9.1 End User guide

Introduction

The purpose of this document to analyze/document impact analysis of new feature , existing features, open bugs etc.

Intended Audience

- End Users of Sonarqube

Overview

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities on 20+ programming languages. SonarQube offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security vulnerabilities. SonarQube empowers all developers to write cleaner and safer code.

Supported platforms for Sonarqube v7.9.1:

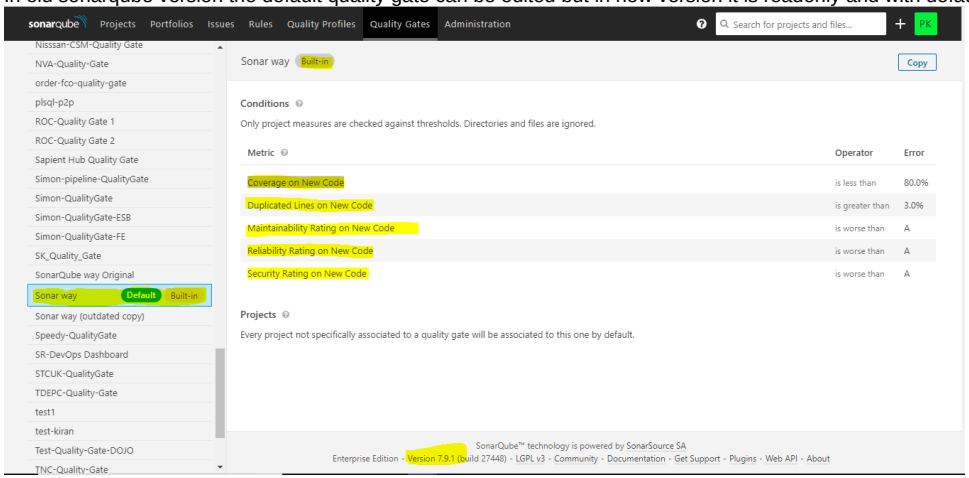
https://docs.sonarqube.org/latest/requirements/requirements/

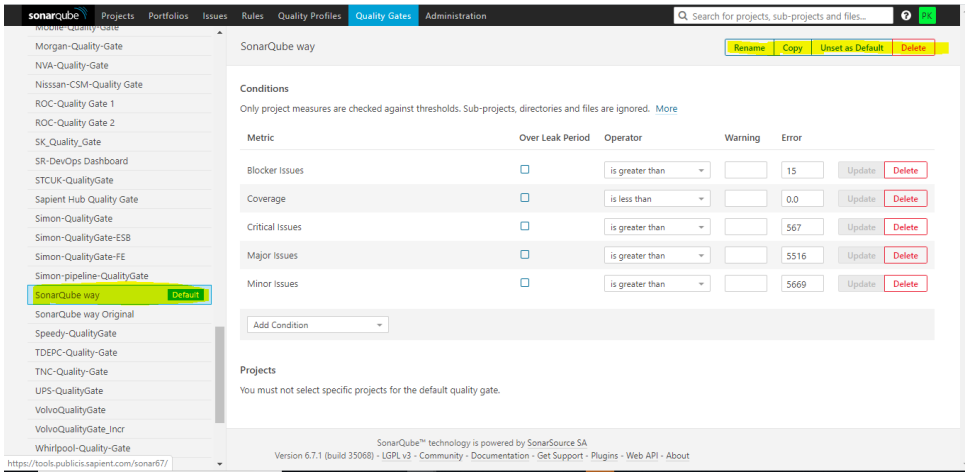
New Features added in the below respective versions:

New Features from 7.0

1. **Built-In Read-Only Quality Gate(for sysadmins also):** In order to make clear the default, minimum and recommended criteria Quality Gates, the "Sonar way" Quality Gate is now read-only, and the default if one is not already set. It may be updated automatically at each upgrade of SonarQube.

   In old sonarqube version the default quality gate can be edited but in new version it is readonly and with default measures added. (Cannot be edited by sys admin). Refer the screenshot for difference.

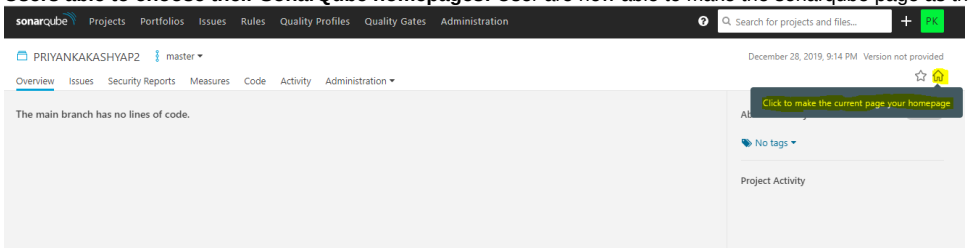




2. **Removed the ability to "unset" the default quality gate:** Currently it's possible to "Unset as Default" the default quality gate, leaving the instance with no default quality gate. But in this version it is not possible to unset as the default QG is made readonly. Can be seen in above screenshots.

New Features from 7.1

1. **Users able to choose their SonarQube homepages:** User are now able to make the sonarqube page as their homepage as per their choice by just clicking on the home icon.



New Features from 7.2

Note: (Try to avoid using this inbuilt feature(PR Decoration) as only one Bitbucket can be used to make this work with our sonarqube instances. Instead use our plugin functionality described below on this page.)

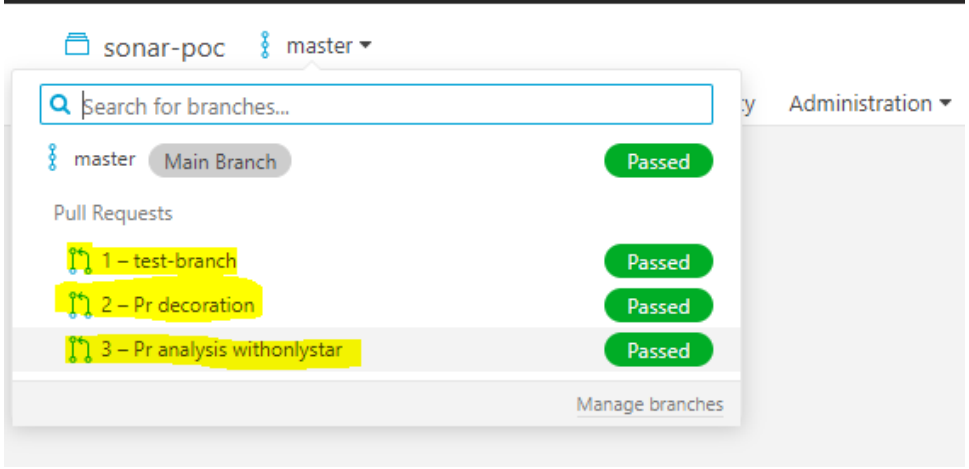
1. **Pull Request Analysis:** SonarQube should now allow to analyze PRS and to vizualize them in the UI. Pull Requests (PRs) are visible in SonarQube from the branches and pull requests dropdown menu of your project.

   PR analysis allows you to:

   - see your PR's Quality Gate status in the SonarQube UI.
   - automatically decorate your PRs with SonarQube issues in your SCM provider's interface.

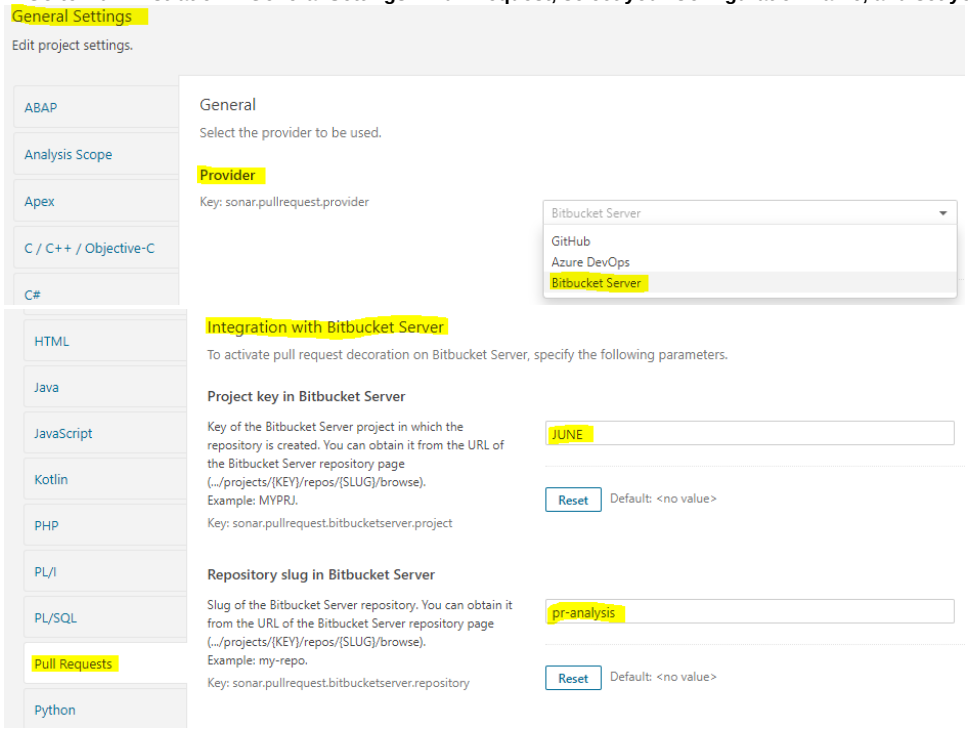
Steps to use PR Analysis: Before using this feature you need to have a repo in Bitbucket and a jenkins job with this repo and a sonar project created in UI. And a Pull Request created in bitbucket.

1. Analyize the master branch initially by running a jenkins job with the respective Bitbucket repository.
2. Add the below parameters in your jenkins job's sonarqube scanner part:
   sonar.pullrequest.key (key of the PR)
   sonar.pullrequest.branch (source branch name)
   sonar.pullrequest.base (target branch name mostly "master")
   sonar.projectKey=projectname (this user will have already specified)
3. Run the jenkins job and you will be able to see the PR analysis of source branch in Sonarqube UI.

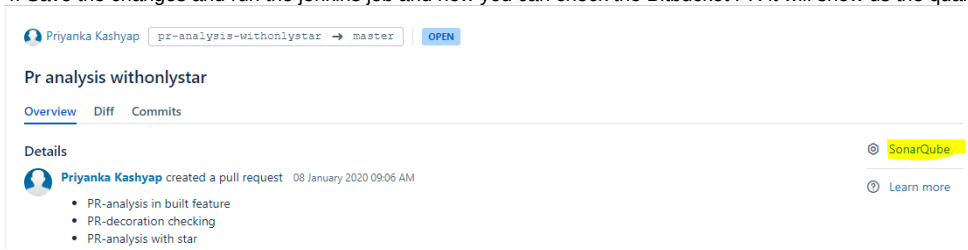


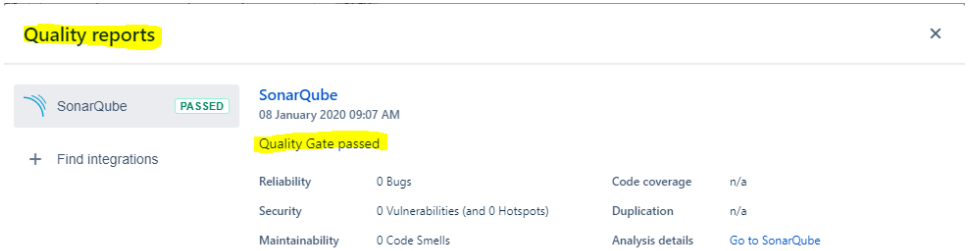
For more infor on PR analysis see the Sonarqube document: https://docs.sonarqube.org/latest/analysis/pull-request/

2. **PR Decoration:** PR Decoration shows the quality gate status of a PR in Bitbucket UI under the PR which is analyized in sonar using PR analysis.
   To Use this feature we need to integrate Bitbucket and sonarqube, which can be done by adding bitbucket's URL and access token in sonarqube. Please see the doc for more info https://docs.sonarqube.org/latest/analysis/pr-decoration/. And this steps has to be done by System admin of sonarqube.
   Steps to use PR Decoration:
   1. Now once a sys admin integrate the Bitbucket and sonar, a project admin of sonar need to configure the below in their project General settings.
   2. Go to Administration > General Settings > Pull Request, select your Configuration name, and set your Project Key and Repo Slug.



3. Here we will be needing the same jenkins job, same repo mentioned above in PR analysis as we are decorating the same PR in Bitbucket for which we did PR analysis.
   There is a field in jenkins freestylejob called as "**Branch Specifier (blank for 'any')**", this filed's value by default is "***/master**" but here to run PR analysis we need to change this value to **"*".**
4. Save the changes and run the jenkins job and now you can check the Bitbucket PR it will show us the quality gate status from sonarqube.

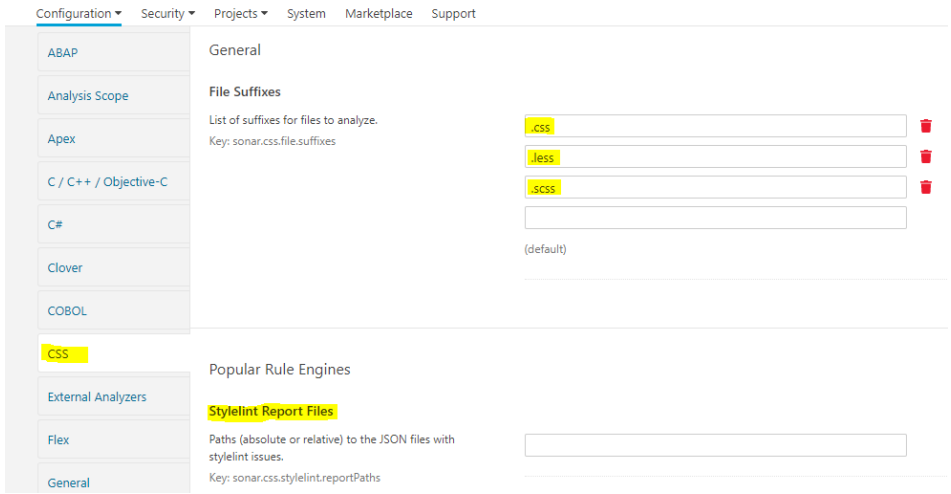




Note: For PR Decoration only one Bitbucket instance can be configured in sonarqube as configuring multiple Bitbucket instances is not yet a feature in new sonarqube version. So we can only integrate our one Bitbucket instance with sonarqube inorder to use this inbuilt PR Decoration feature from new sonar version.

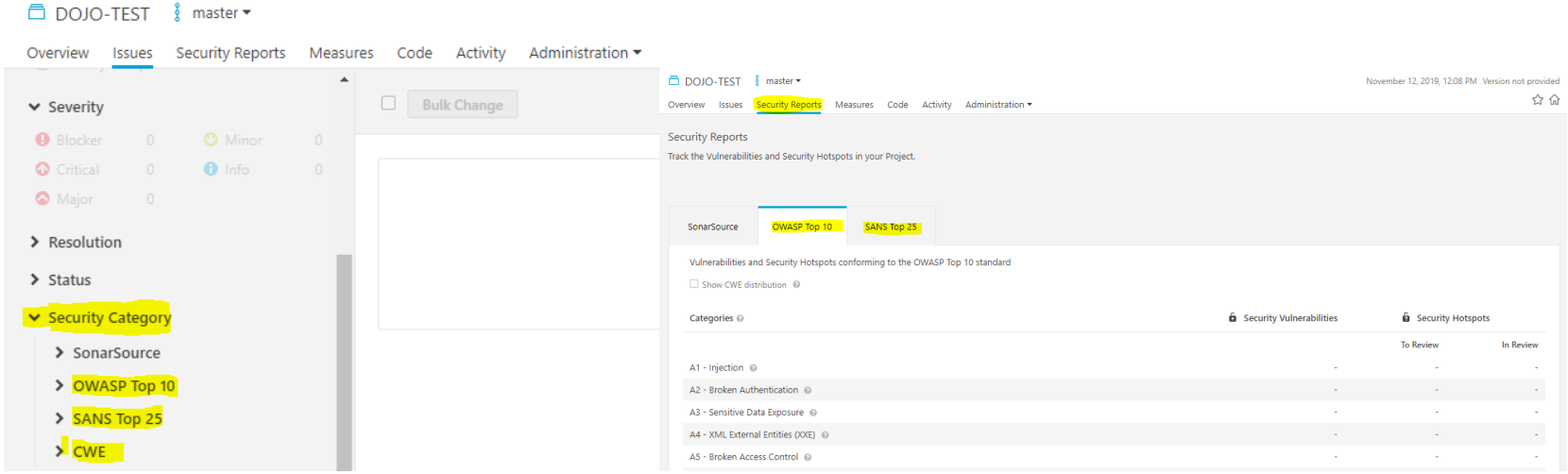
New Features from 7.3

1. **SCSS,Less Language Quality Profiles embedded into SonarCSS:** SonarCSS 1.0 is now released and brings 24 rules and support for StyleLint.io. **Quality Profiles in language SCSS and Less gets removed once new version is upgraded(QP list). So need to take a backup of them if needed.** In new version **SonarCSS** is an inbuilt plugin now.



2. **Embedded SonarKotlin:** SonarKotlin is now an inbuilt plugin in this version. SonarKotlin 1.0 is now released and brings 41 rules and support for AndroidLint and Detekt. **For users a new quality profile will be available now to use Kotlin rules(Sonar Way).**

3. **Added Standards facet on the Issues page:** New "Standards" facet containing 3 sub-facets (**OWASP Top 10**, **SANSTop 25** and **CWE**) has been added on issues page under **"Security Category"**.



4. **New "Administer Security Hotspots" Permission:** During the upgrade, the new "Administer Security Hotspots" permission is granted to all users/groups who already have the "Administer Issues" permission.

New Features from 7.4

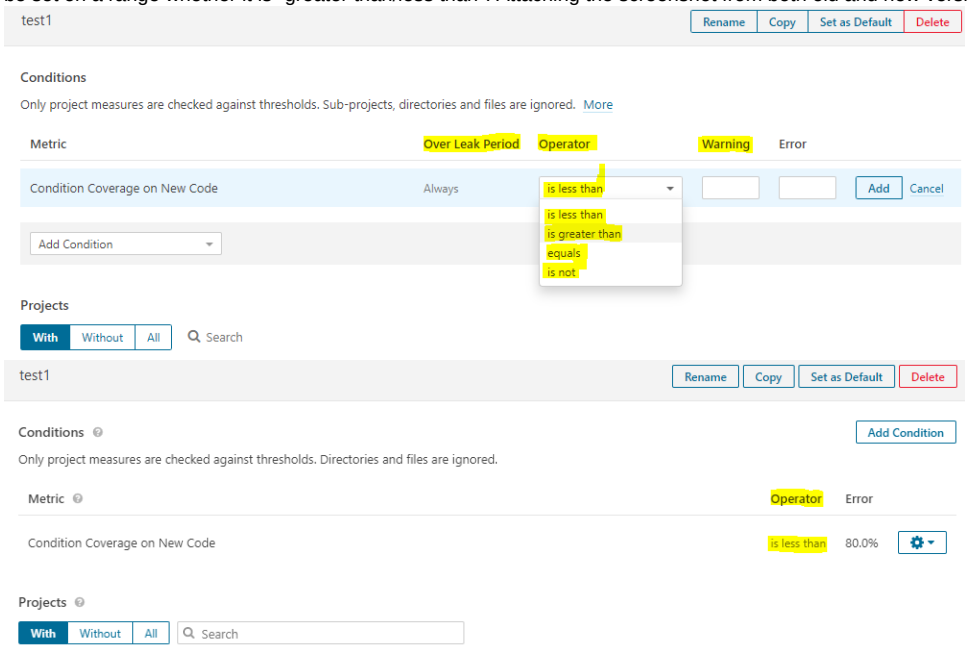
1. **"Leak" replaced with "New Code":** Wording has been updated throughout the interface to replace "Leak" and "Leak Period" with "New Code" and "New Code Period".

New Features from 7.5

1. **Embed SonarApex in Enterprise+ Edition:** Code Analyzer for Apex has been added as a new plugin called "SonarApex".
2. **Embed SonarScala in all SonarQube editions:** Code Analyzer for Scala has been added as a new plugin called "SonarScala".

New Features from 7.6

1. **Quality Gates Simplified:** Quality Gates have been streamlined to remove a number of confusing options. Conditions previously using the "on new code" checkbox will be migrated to On New Code metrics. For example, a condition previously using the overall Coverage metric with the "on new code" checkbox enabled will be migrated to a condition using the Coverage on New Code metric.
   Note: Here "on new code" is referring to "Over Leak Period".
   The ability to set "Warning" and "Over Leak Period" removed from UI: These parameters has been removed from Quality gate settings. Check this for more info (MMF-473).
   Range of Operators on a conditions are hardcoded: The ability to choose from a range of operators on a condition are hardcoded now and cannot be changed by anyone. The condition choosen will automatically be set on a range whether it is "greater than/less than". Attaching the screenshot from both old and new versions to clarify the same for all the above changes for qaulity gate.



New Features from 7.7

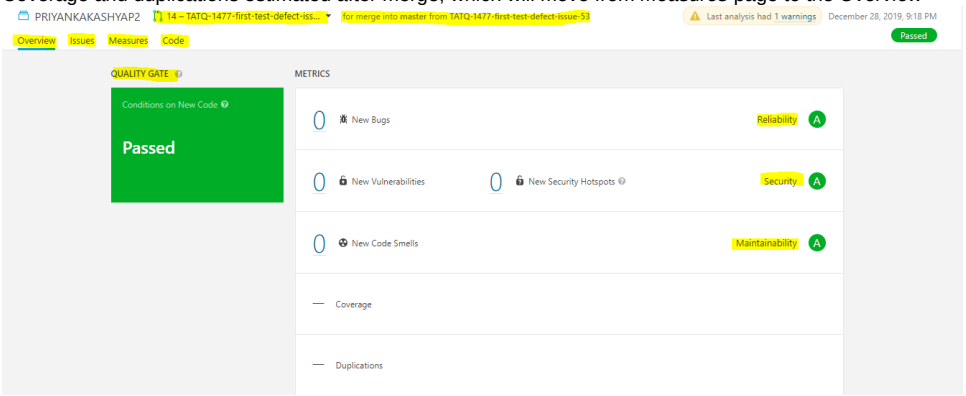
1. Deprecated parameters dropped: sonar.language, and sonar.profile, both deprecated since 4.5, are dropped in this version as is sonar.analysis.mode, which as been deprecated since 6.6. These now-unrecognized parameters will simply be ignored, rather than failing analysis.

2. **Added new Overview tab for PRs and SLBs:**The status for PRs/short-lived branches will now correspond to the Quality Gate status and will be displayed with the same visual indicator than for all the branches.

   A new *Overview* tab will appear for PRs/short-lived branches, with:

   - The Quality Gate status (the information could possibly be repeated in all pages)
   - Conditions in failure
   - Main metrics on new code to help to understand the status of the PR:
     - Reliability, security and maintainability ratings
     - Number of bugs, vulnerabilities and code smells
     - Coverage %
     - Duplications %
     - New lines to cover
     - New lines of code
     - A link to the PR on the ALM (See the PR)
     - Explanations about the Quality Gate that is used, and about the fact that only conditions on new code are evaluated

- Coverage and duplications estimated after merge, which will move from *Measures* page to the *Overview*



New Features from 7.8

1. **Scanner version compatibility:**

   Only the following scanner versions are compatible with SonarQube 7.8:

   - SonarQube Scanner CLI 2.9+
   - SonarQube Scanner Maven 3.3.0.603+
   - SonarQube Scanner Gradle 2.3+
2. **Analysis fails with old branch parameter** `sonar.branch` was deprecated in 6.7. With this version analysis fails when it is used. Where it is still in use, simply remove the `sonar.branch` property and update your `sonar.projectKey` value to `key:branch`.
3. **Notifications changes:** Several changes have been made to notificatons. The notifications algorithm has been replaced with one that offers better performance during background task processing. Issue change notifications spawned by analysis or bulk change now generate only one email per event rather than one email per issue. The ability to subscribe globally to new issues notifications and notifications for issues resolved as False Positive or Won't fix has been dropped, as have all such subscriptions. Issue-related notifications on PRs have also been dropped.

New Features from 7.9

1. **Pylint should be run manually:** Running Pylint automatically during python analysis has been deprecated. If needed, Pylint must be run ahead of time and the resulting report passed in to analysis.

Plugins need to update on Production with below versions:

S. No	Old Sonar version	Plugin Name	Old Plugin version	New Sonar version	New Plugin version(compatible)
1	6.7	Android (Import Android Lint reports)	1.1	7.9.1	Natively build into the product - no need for a plugin in new sonar version. This features comes with

					SonarKotlin which now comes with in built package.
2	6.7	Branch (Branch plugin)	1.0 (build 507)	7.9.1	Support of branches natively build into the product - no need for a plugin.
3	6.7	Checkstyle (Analyze Java code with Checkstyle)	4.7	7.9.1	4.27
4	6.7	Clover (Get code coverage with Atlassian Clover)	3.1	7.9.1	4.1
5	6.7	Crowd (Delegates authentication to Atlassian Crowd)	2.0	7.9.1	2.1.3
6	6.7	Developer (Developer oriented features)	1.0 (build 240)	7.9.1	Natively build into the product - no need for a plugin in new sonar version.
7	6.7	Findbugs (Analyze Java, Scala, Closure and JSP code with SpotBugs. 3.1.12)	3.9.4	7.9.1	3.11.1
8	6.7	Flex (Enables scanning of Flex source files)	2.3	7.9.1	2.5.1 (SonarFlex) new name
9	6.7	GitHub (Provide some integration between GitHub and SonarQube)	1.4.2 (build 1027	7.9.1	Natively build into the product - no need for a plugin in new sonar version.
10	6.7	Governance (Governance plugin)	2.0.2 (build 3011)	7.9.1	Natively build into the product - no need for a plugin in new sonar version.
11	6.7	JaCoCo (JaCoCo XML report importer)	1.0.2 (build 475)	7.9.1	1.0.2
12	6.7	LDAP (Delegates authentication to LDAP)	2.2 (build 608)	7.9.1	2.2
13	6.7	License (SonarSource License Manager)	3.3 (build 1341)	7.9.1	Natively build into the product - no need for a plugin in new sonar version.
14	6.7	PMD (Analyze Java code with PMD)	2.6	7.9.1	3.2.1
15	6.7	SonarABAP (Enable analysis and reporting on ABAP projects)	3.5 (build 1080)	7.9.1	3.8
16	6.7	SonarC# (Code Analyzer for C#)	6.7.1 (build 4347)	7.9.1	8.1
17	6.7	SonarCFamily (Code Analyzer for C, C++, Objective-C)	5.0 (build 9359)	7.9.1	6.6.0
18	6.7	SonarCOBOL (Code Analyzer for COBOL)	4.1.1 (build 2663)	7.9.1	4.4
19	6.7	SonarJS (Code Analyzer for JavaScript)	4.0 (build 5862)	7.9.1	6.1
20	6.7	SonarJava (Code Analyzer for Java)	5.12.1 (build 17771)	7.9.1	6.0
21	6.7	SonarPHP (Code Analyzer for PHP)	2.12.0.2871	7.9.1	3.3.0.5166
22	6.7	SonarPLI (Code Analyzer for PL/I)	1.7 (build 1117)	7.9.1	1.10
23	6.7	SonarPLSQL (Code Analyzer for PL/SQL)	3.2 (build 1753)	7.9.1	3.4.1
24	6.7	SonarPython (Code Analyzer for Python)	1.8 (build 1496)	7.9.1	2.3
25	6.7	SonarQube :: Plugins :: SCM :: Git (Git SCM Provider for SonarQube)	1.3.0.869	7.9.1	Natively build into the product - no need for a plugin in new sonar version.
26	6.7	SonarQube :: Plugins :: SCM :: SVN (Subversion SCM Provider for SonarQube)	1.6.0.860	7.9.1	Natively build into the product - no need for a plugin in new sonar version.
27	6.7	SonarRPG (Code Analyzer for RPG)	2.2 (build 1005)	7.9.1	2.3
28	6.7	SonarSwift (Code Analyzer for Swift)	3.1 (build 2067)	7.9.1	4.2.2

29	6.7	SonarTS (Code Analyzer for TypeScript)	1.1 (build 1079)	7.9.1	2.1
30	6.7	SonarTSQL (Code Analyzer for T-SQL)	1.1 (build 2177)	7.9.1	1.4
31	6.7	SonarVB (Code Analyzer for VB.NET)	4.2 (build 248)	7.9.1	8.1
32	6.7	SonarVB6 (Code Analyzer for Visual Basic)	2.3 (build 992)	7.9.1	2.6
33	6.7	SonarXML (Code Analyzer for XML)	1.4.3 (build 1027)	7.9.1	2.0.1
34	6.7	Web (Enables scanning of HTML, and JSP/JSF files)	2.5.0.476	7.9.1	3.2(SonarHTML) with new name.

PR Analysis using plugins:

Bamboo-Sonar-Bitbucket:

Plugins used: "Sonar for Bamboo" and "Sonar for Bitbucket"

1. Create a project in sonar. (sonar-demo-plugin)
2. Create a repo in bitbucket with some changes.(pranalysis-plugin)
3. Add sonar project in Bitbucket repo settings created above with this enabled:
   Use new Sonar branching and pull request support.
4. Create a plan in bamboo and add the above repo created in it. (sonar-demo-plugin)
   In Branches: **select when PR is created.**
   Add task: sonar-scanner
   select: JDk of bamboo agent
   select: sonar-scanner installed on agent.
   Add parameters:
   -Dsonar.projectKey=sonar-demo-plugin

   Select: stage-new-sonar
   Select: Use pull request analysis if possible (only available in commercial editions of SonarQube and for Bamboo >= 6.9.0)
   save

   Run the bamboo build.
5. Now create one more branch in repo and push changes to UI. Create a PR in Bitbucket with this branch.
6. Go to Bamboo, you will be able to see the new branch created over there. It is created because we have selected "**select when PR is created." in bamboo.**
   select it and run the build.(**run branch**)

   Build successfull and PR is visible on sonar UI.

   And quality gate status is visible on PR in bitbucket.

Jenkins-Sonar-Bitbucket:

1. Create a project in sonar.
2. Create a repo in bitbucket and enable sonar project in it.
3. Add sonar project in Bitbucket repo settings created above with this enabled:
   Use new Sonar branching and pull request support.
4. Create a jenkins job and add the repo in it.

   Add below task in jenkins job:

Execute shell:
echo SONAR_BRANCH=$(printf '%s' $GIT_BRANCH | cut -d'/' -f 2-) > sonar-branch
Inject Environment variables:
sonar-branch

Execute SonarQube Scanner:
Stage-sonar
Analysis properties:
sonar.branch.name=$SONAR_BRANCH
sonar.projectKey=yourprojectkey
sonar.sources=.

5. **Branch Specifier (blank for 'any')**keep: */master for first analysis
   Change it to * when running PR analysis for all branches.
6. Create a branch in repo, create a PR in UI with this branch.
7. Run the jenkins job and check sonar Ui and PR.