Setup SonarQube with Jenkins Declarative Pipeline

Rosaniline
Aug 22, 2018 · 2 min read

Before start, you should have done…

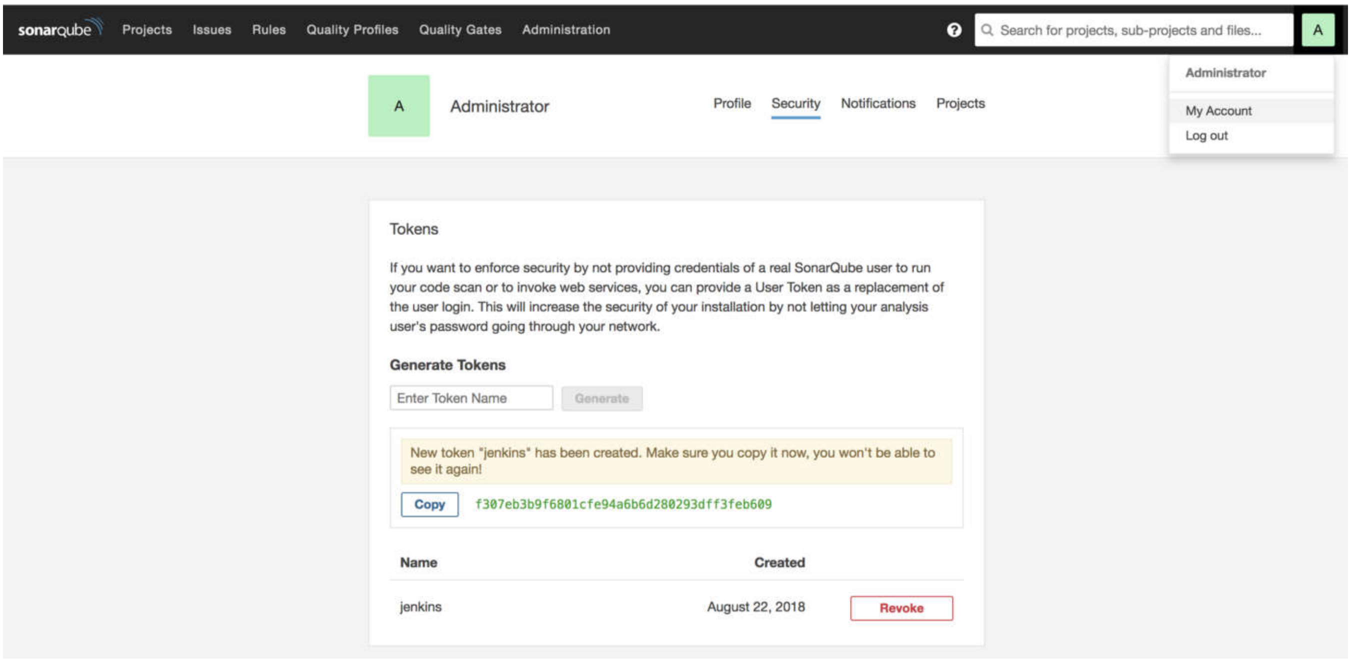
- Jenkins Multibranch Pipeline project setup with declarative pipeline Jenkinsfile. If not, please refer to this tutorial
- `sonar-project.properties` file at your project root

Prerequisite

- **SonarQube Server**
  Installation Guide: Link
  However, it's easier to run a docker version, as always: Link
- Jenskins Plugin: **SonarQube Scanner for Jenkins**

Setup

- Create SonarQube User Token at SonarQube Server
  User > My Account > Security > Generate Tokens



Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Enter Token Name | Generate

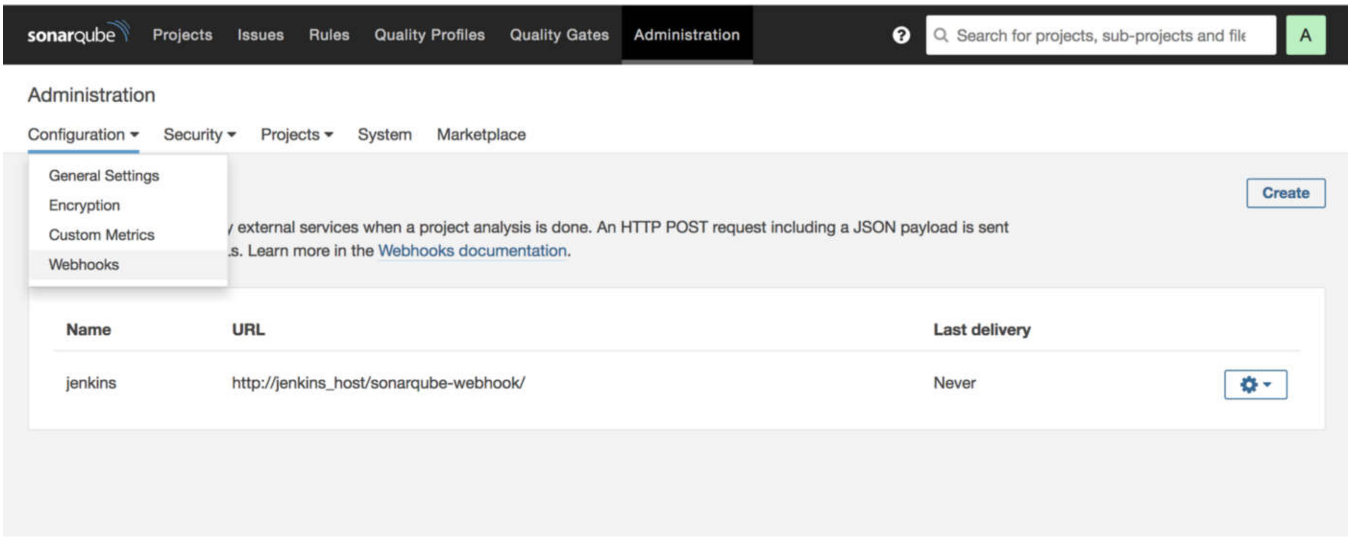
New token "jenkins" has been created. Make sure you copy it now, you won't be able to see it again!

Copy `f307eb3b9f6801cfe94a6b6d280293dff3feb609`

Name	Created	
jenkins	August 22, 2018	Revoke

Click the upper-right button [A] to expand the user dropdown, which took me a while to figure out where it is

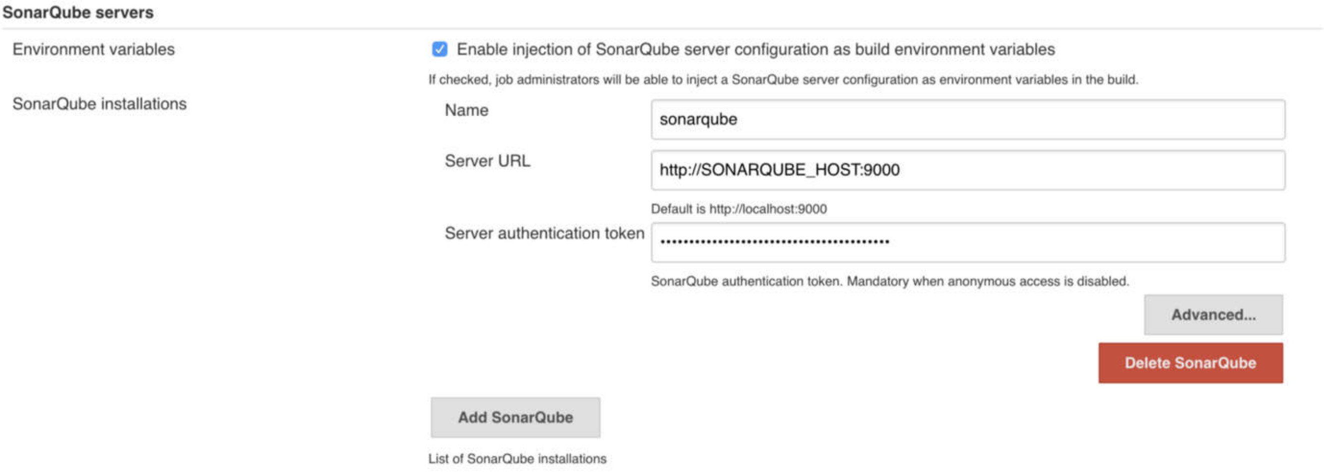
- Configurate SonarQube webhook for quality gate
  Administration > Configuration > Webhooks > Create
  The URL should point to your Jenkins server `http://{JENKINS_HOST}/sonarqube-webhook/`



/ external services when a project analysis is done. An HTTP POST request including a JSON payload is sent
.s. Learn more in the Webhooks documentation.

Name	URL	Last delivery
jenkins	http://jenkins_host/sonarqube-webhook/	Never

- Setup SonarQube server at Jenkins
  Manage Jenkins > Configure System > SonarQube servers
  Enter you SonarQube server host and the token generated at previous step



SonarQube servers

Environment variables ☑ Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name: sonarqube

Server URL: http://SONARQUBE_HOST:9000

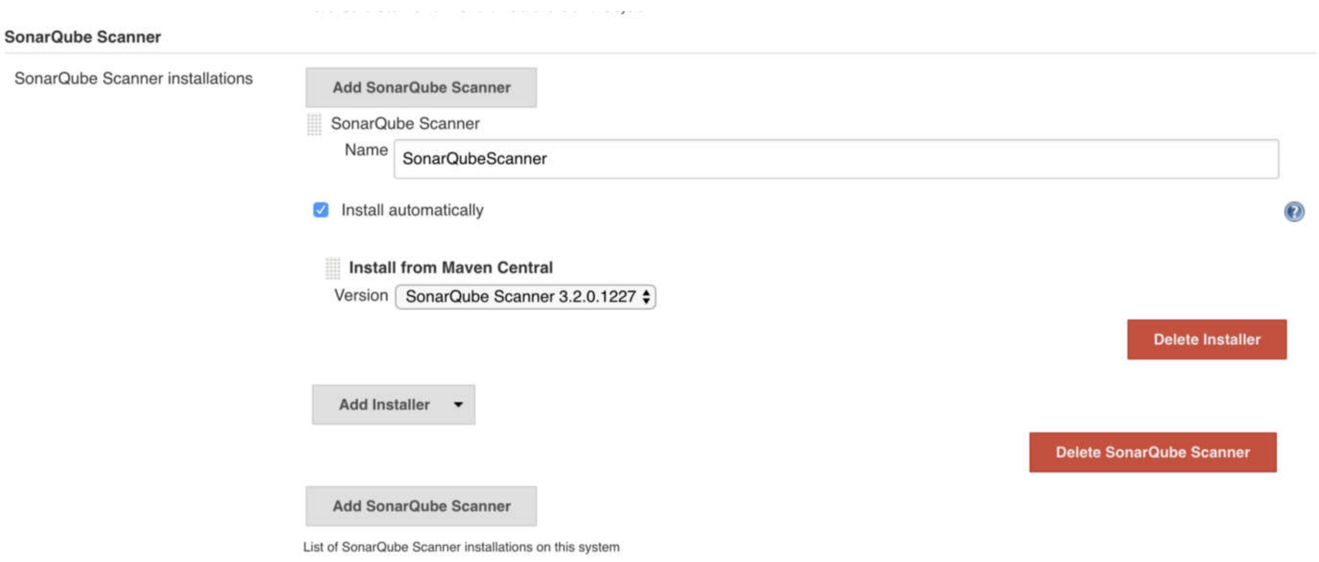
Default is http://localhost:9000

Server authentication token: ....

SonarQube authentication token. Mandatory when anonymous access is disabled.

List of SonarQube installations

- Setup SonarQube scanner at Jenkins
  Manage Jenkins > Global Tool Configuration > SonarQube Scanner
  Remember to give it a name, let's take **SonarQubeScanner** as example here



SonarQube Scanner

SonarQube Scanner installations

SonarQube Scanner

Name: SonarQubeScanner

☑ Install automatically

Install from Maven Central

Version: SonarQube Scanner 3.2.0.1227

List of SonarQube Scanner installations on this system

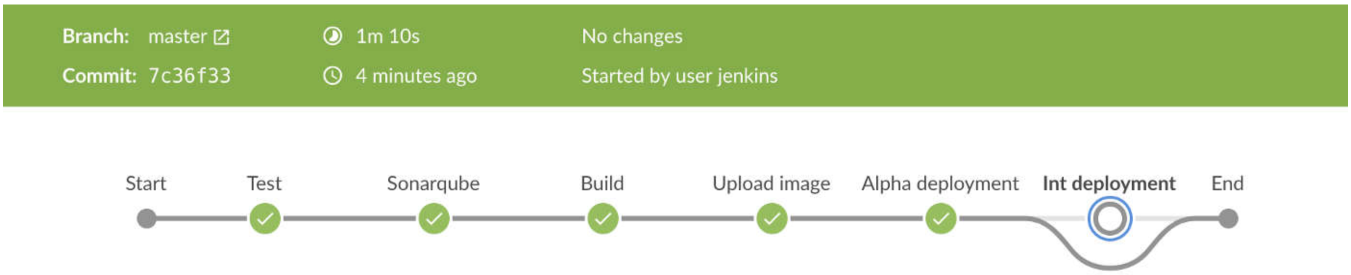
- Setup SonarQube scan stage in declarative pipeline Jenkinsfile

```
stage('Sonarqube') {
    environment {
        scannerHome = tool 'SonarQubeScanner'
    }

    steps {
        withSonarQubeEnv('sonarqube') {
            sh "${scannerHome}/bin/sonar-scanner"
        }

        timeout(time: 10, unit: 'MINUTES') {
            waitForQualityGate abortPipeline: true
        }
    }
}
```

Noted that the tool name is the one you named for your SonarQube scanner at previous step

`waitForQualityGate` will halt the pipeline until SonarQube notifies Jenkins whether quality gate is passed through webhook setup earlier

Now you should see SonarQube in your pipeline, happy coding



Branch: master

Commit: 7c36f33

1m 10s

4 minutes ago

No changes

Started by user jenkins

Here we put SonarQube after testing stage for testing report collection, there are examples running SonarQube in parallel with testing stage. It's all up to your need.

This post is tested with SonarQube 7.1 and Jenkins 2.121.3