Jenkins

> User Documentation Home

User Handbook

- User Handbook Overview
- Installing Jenkins
- Platform Information
- Using Jenkins
- Pipeline
- Blue Ocean
- Managing Jenkins
- Securing Jenkins
- System Administration
- Scaling Jenkins
- Troubleshooting Jenkins
- Glossary

Tutorials

- Guided Tour
- Jenkins Pipeline
- Using Build Tools

Resources

- Pipeline Syntax reference
- Pipeline Steps reference
- LTS Upgrade guides

The following plugin provides functionality available through Pipeline-compatible steps. Read more about how to integrate steps into your Pipeline in the Steps section of the Pipeline Syntax page.

For a list of other such plugins, see the Pipeline Steps Reference page.

Table of Contents

SonarQube Scanner for Jenkins

View this plugin on the Plugins site

`waitForQualityGate`: Wait for SonarQube analysis to be completed and return quality gate status

This step pauses Pipeline execution and wait for previously submitted SonarQube analysis to be completed and returns quality gate status. Setting the parameter `abortPipeline` to true will abort the pipeline if quality gate status is not green.

Note: This step doesn't require an executor.

Requirements:

- SonarQube server 6.2+
- Configure a webhook in your SonarQube server pointing to `<your Jenkins instance>/sonarqube-webhook/`. **The trailing slash is mandatory!**
- Use `withSonarQubeEnv` step to run your analysis prior to use this step

Example using declarative pipeline:

```
pipeline {
  agent none
  stages {
    stage("build & SonarQube analysis") {
      agent any
      steps {
        withSonarQubeEnv('My SonarQube Server') {
          sh 'mvn clean package sonar:sonar'
        }
      }
    }
    stage("Quality Gate") {
      steps {
        timeout(time: 1, unit: 'HOURS') {
          waitForQualityGate abortPipeline: true
        }
      }
    }
  }
}
```

Example using scripted pipeline:

```
stage("build & SonarQube analysis") {
    node {
        withSonarQubeEnv('My SonarQube Server') {
           sh 'mvn clean package sonar:sonar'
        }
    }
}

stage("Quality Gate"){
    timeout(time: 1, unit: 'HOURS') {
        def qg = waitForQualityGate()
        if (qg.status != 'OK') {
            error "Pipeline aborted due to quality gate failure: ${qg.status}"
        }
    }
}
```

- `abortPipeline : boolean`
- `credentialsId : String` (optional)
- `webhookSecretId : String` (optional)

`withSonarQubeEnv`: Prepare SonarQube Scanner environment

- `installationName : String`
- `credentialsId : String` (optional)
- `envOnly : boolean` (optional)

Was this page helpful?

Please submit your feedback about this page through this quick form.

Alternatively, if you don't wish to complete the quick form, you can simply indicate if you found this page helpful?

○ Yes ○ No

Type the answer to 8 plus 1 before clicking "Submit" below.

Submit

See existing feedback here.

⚠ Report page issue

Resources

Downloads
Blog
Documentation
Plugins
Security
Contributing

Project

Structure and governance
Issue tracker
Roadmap
GitHub
Jenkins on Jenkins

Community

Forum
Events
Mailing lists
Chats
Special Interest Groups

𝕏 (formerly Twitter)
Reddit

Other

Code of Conduct
Press information
Merchandise
Artwork
Awards