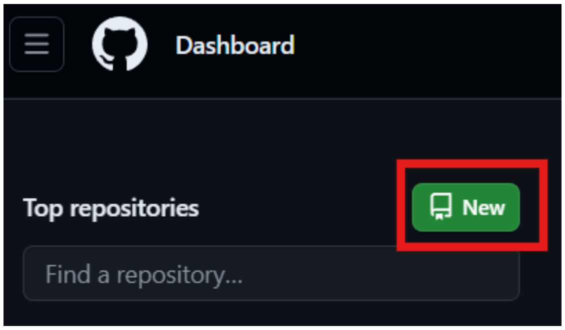
SETUP CONTINUES INTEGRATION

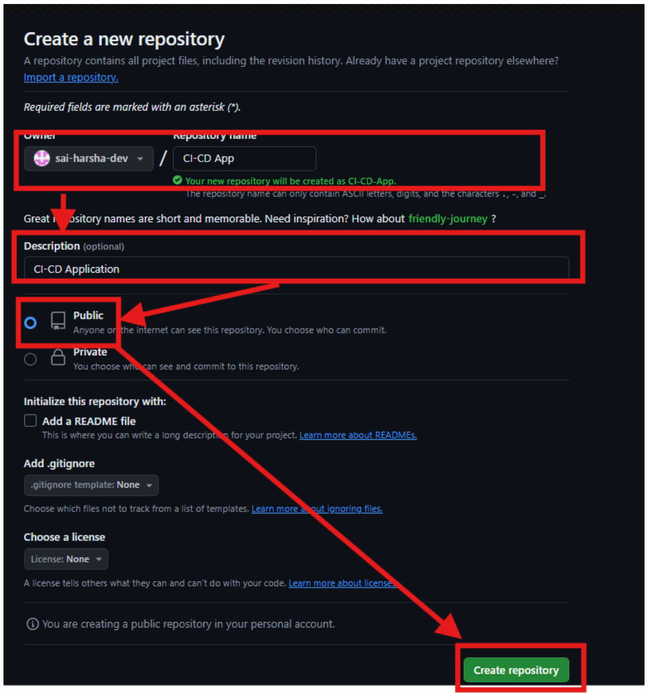
SETUP SOURCE REPOSITORY

✓ Create GITHUB repository (*for local files)

- Login to https://github.com/ and Sign in to your account.
- Add a new repository by clicking on **NEW ICON.**



- Give the repository a Name, Description and keep it as Public and Create the repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner: sai-harsha-dev / Repository name: CI-CD App

✓ Your new repository will be created as CI-CD-App.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about friendly-journey ?

Description (optional)

CI-CD Application

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

□ **Add a README file**
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: **None**

A license tells others what they can and can't do with your code. Learn more about licenses.

ⓘ You are creating a public repository in your personal account.

✓ Commit Source code files (*for local files)

- Initialize and add the source files using the following commands.

```
$ git init
$ git add .
$ git commit -m "<Commit message>"
```

- Setup the remote URL

```
$ git remote add <remote-name> <remote-url>
```

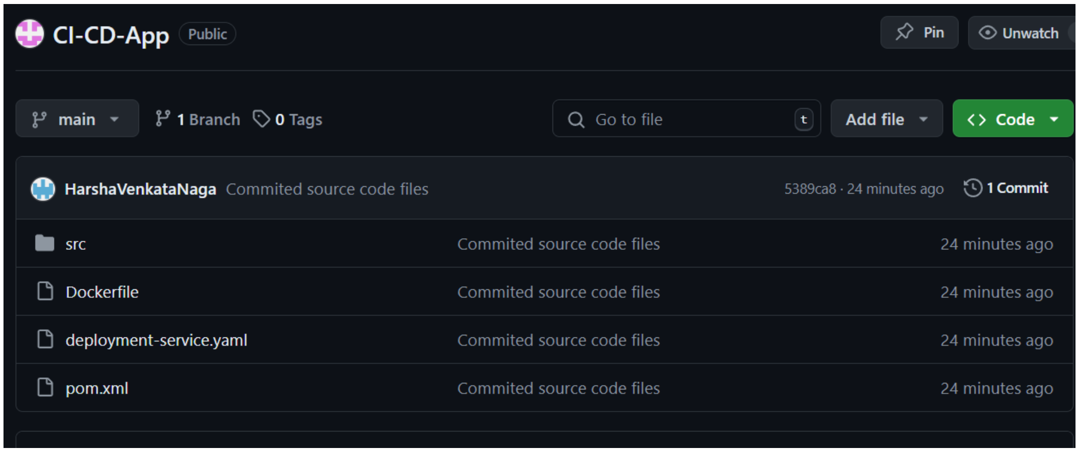
- Update branch name as per GITHUB conventions

```
git branch -M main
```

- Setup up upstream and push.

```
$ git push -u <remote-name> main
```

- Check if the files are reflected in the GITHUB repo.

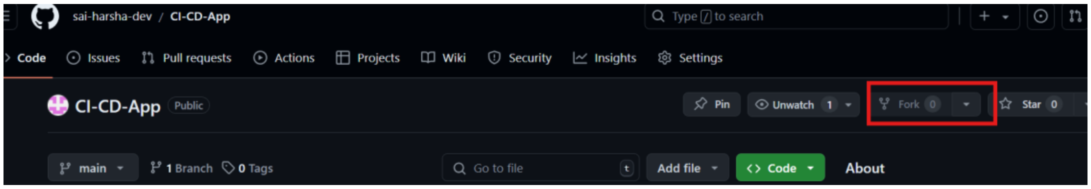


HarshaVenkataNaga Commited source code files 5389ca8 · 24 minutes ago 1 Commit

src	Commited source code files	24 minutes ago
Dockerfile	Commited source code files	24 minutes ago
deployment-service.yaml	Commited source code files	24 minutes ago
pom.xml	Commited source code files	24 minutes ago

✓ Fork the existing repository (*for using existing repo)

- Login to https://github.com/ and Sign in to your account.
- Search for the repository CICD-APP
- Fork the repository, by clicking on the **fork icon.**



CREATE PIPELINE SCRIPT

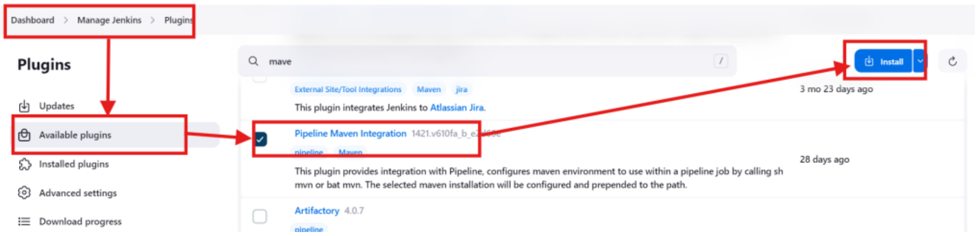
✓ Checkout Repository

- The first stage involves checking out source code files from GITHUB repo.
- Include the following pipeline code in Jenkinsfile.

```
stage ("GITHUB CHECKOUT"){
            steps{
                git branch: '<branch-name>', url: '<git-url>'
            }
        }
```

✓ Run source code compilation

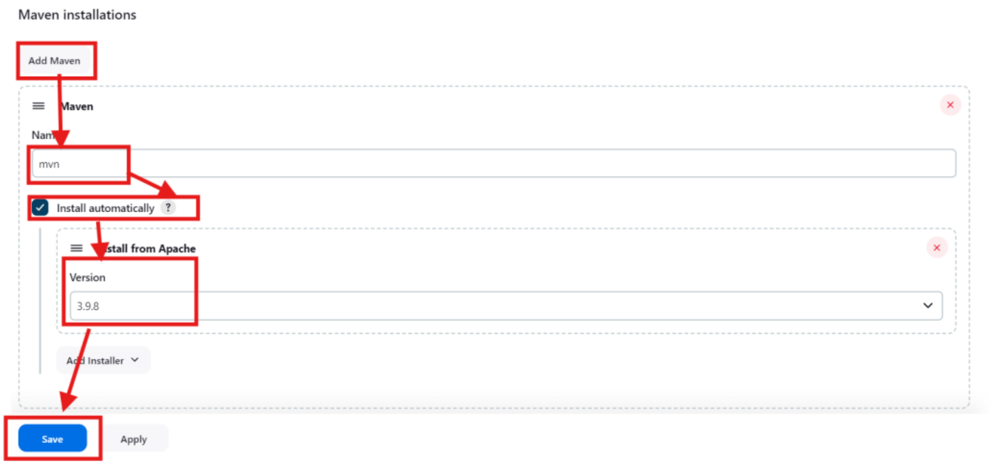
- Install the Maven plugin.
  - Login to Jenkins
  - Move **Mange Jenkins > Plugins > Available Plugins.**
  - Search for **Pipeline Maven Integration > Select it > Install.**
  - **Restart Jenkins** if needed.



This plugin integrates Jenkins to Atlassian Jira.

This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.

- Setup the Maven.
  - Head over to **Manage Jenkins > Tools > Maven Installations.**
  - **Select Install Automatically > Select version** (*not mandatory)
  - **Save** changes.



Maven installations

Maven

Name: mvn

☑ Install automatically

Install from Apache

Version: 3.9.8

- ➢ Include the following pipeline code in Jenkinsfile.

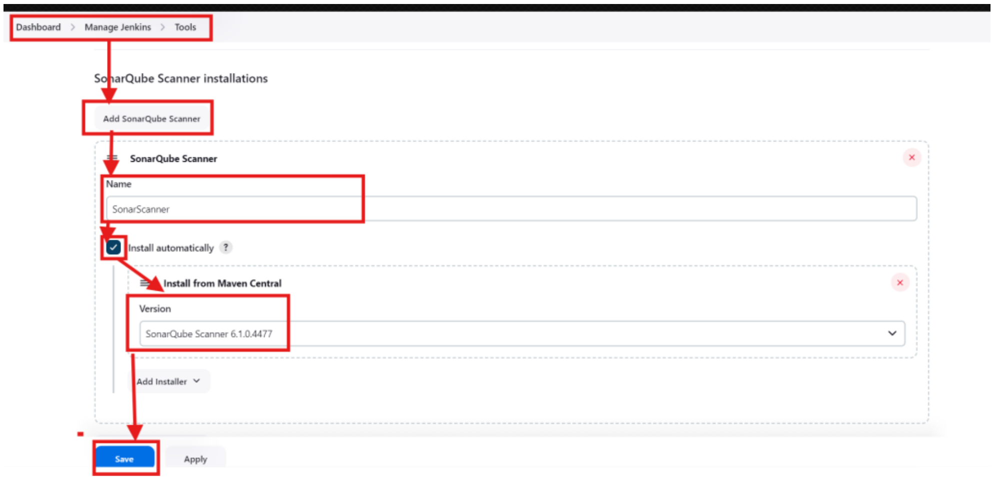
```
tools{

        maven 'mvn'
    }

stage ("COMPILE SOURCE CODE"){
            steps{
                sh "mvn compile"
            }
        }
```

✓ Run Code Quality Check

- ➢ Setup the SonarScanner. (*Make sure **Sonar Server is setup in Manage Jenkins > System,** refer DAY-1 document)
  - Head over to **Manage Jenkins > Tools > SonarScanner.**
  - **Select Install Automatically > Select version** (*not mandatory)
  - **Apply** changes.


Dashboard > Manage Jenkins > Tools
SonarQube Scanner installations
Add SonarQube Scanner
SonarQube Scanner
Install automatically ?
Install from Maven Central
Add Installer
Save
Apply

Name: SonarScanner

☑ Install automatically

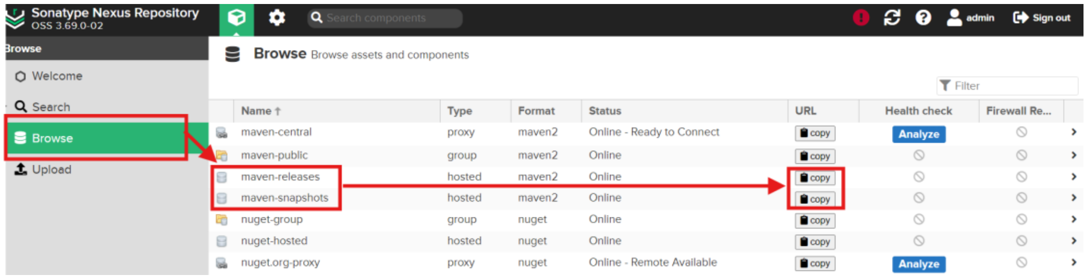
Version: SonarQube Scanner 6.1.0.4477

- ➢ Include the following pipeline code in Jenkinsfile.

```
stage ("CODE QUALITY CHECK"){
            steps{
                withSonarQubeEnv('SonarQube') {
                sh "mvn sonar:sonar"
                }
            }
       }
```

✓ Generate and Store Artifacts

- ➢ Add the repository links in **pom.xml** file, in the respective section.
- ➢ Login to **Nexus server > Select Browse > Click Copy besides the respective URL > Copy the URL**.



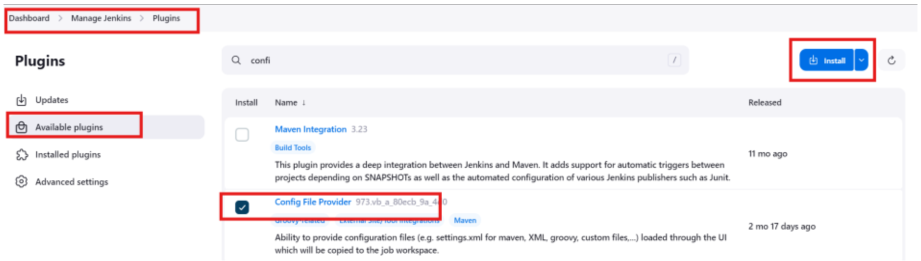
Name ↑	Type	Format	Status	URL	Health check	Firewall Re...
maven-central	proxy	maven2	Online - Ready to Connect	copy	Analyze	⃠
maven-public	group	maven2	Online	copy	⃠	⃠
maven-releases	hosted	maven2	Online	copy	⃠	⃠
maven-snapshots	hosted	maven2	Online	copy	⃠	⃠
nuget-group	group	nuget	Online	copy	⃠	⃠
nuget-hosted	hosted	nuget	Online	copy	⃠	⃠
nuget.org-proxy	proxy	nuget	Online - Remote Available	copy	Analyze	⃠

- ➢ Repeat the same for both **maven-releases, maven snapshots** and add them to **pom.xml file**

```
<distributionManagement>
        <repository>
            <id>maven-releases</id>
            <url><maven-releases repo url></url>
        </repository>
        <snapshotRepository>
            <id>maven-snapshots</id>
            <url><maven-snapshots repo url></url>
        </snapshotRepository>
</distributionManagement>
```

➢ Install the Config File Provider plugin.

- Login to Jenkins
- Move **Mange Jenkins > Plugins > Available Plugins.**
- Search for **Config File Provider > Select it > Install.**
- **Restart Jenkins** if needed.



□ Maven Integration 3.23 — Build Tools — 11 mo ago

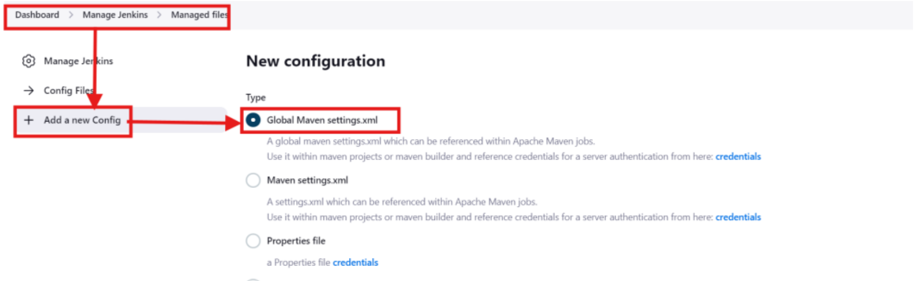
This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.

☑ Config File Provider 973.vb_a_80ecb_9a_4d0 — Maven — 2 mo 17 days ago

Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.

➢ Configure the credentials for Nexus in **setting.xml** file.

- Under **Manage Jenkins > Managed Files > Select Add new Config > Global Maven setting.xml.**



● Global Maven settings.xml

A global maven settings.xml which can be referenced within Apache Maven jobs.
Use it within maven projects or maven builder and reference credentials for a server authentication from here: credentials

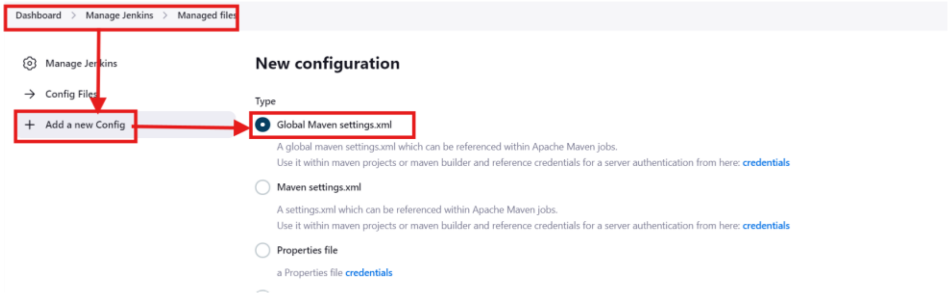
○ Maven settings.xml

A settings.xml which can be referenced within Apache Maven jobs.
Use it within maven projects or maven builder and reference credentials for a server authentication from here: credentials

○ Properties file

a Properties file credentials

- Select **Add new Config > Global Maven setting.xml > Give an ID > Next**



● Global Maven settings.xml

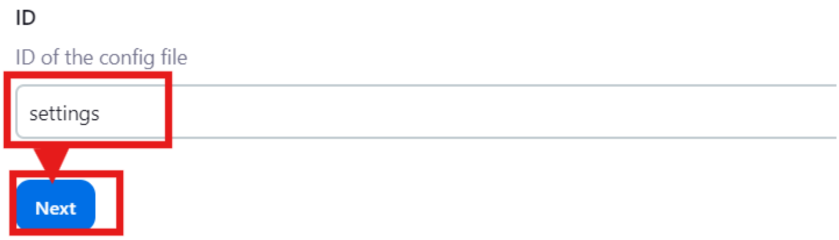
A global maven settings.xml which can be referenced within Apache Maven jobs.
Use it within maven projects or maven builder and reference credentials for a server authentication from here: credentials

○ Maven settings.xml

A settings.xml which can be referenced within Apache Maven jobs.
Use it within maven projects or maven builder and reference credentials for a server authentication from here: credentials

○ Properties file

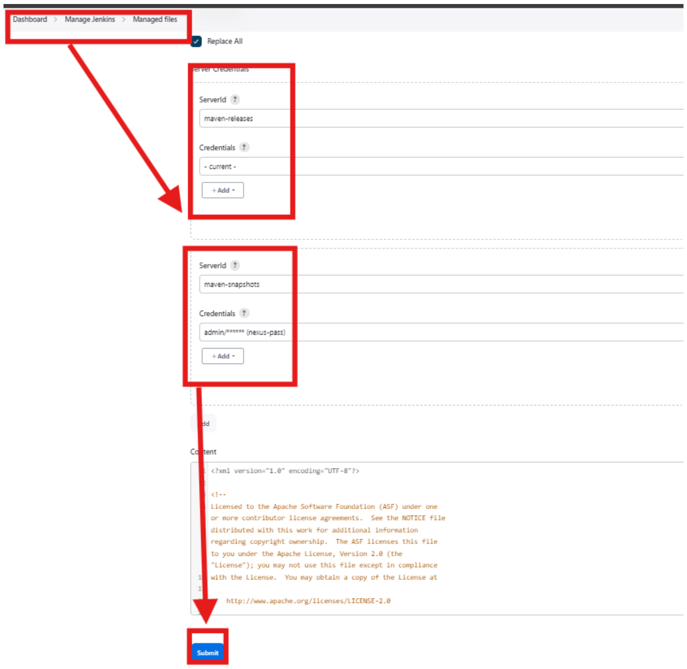
a Properties file credentials


ID

ID of the config file

settings

- **Add Server Credentials > Give ID (maven-releases) > Add > Global Credentials > User Name and Password** add user name and password
- Repeat the same for Maven snapshots
- Submit



erver Credentials

ServerId: maven-releases

Credentials: - current -

ServerId: maven-snapshots

Credentials: admin/****** (nexus-pass)

Co tent

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements.  See the NOTICE file
distributed with this work for additional information
regarding copyright ownership.  The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0
```

➢ Include the following pipeline code in Jenkinsfile.

```
stage("DEPLOY ARTIFACTS TO NEXUS"){
    steps{
       withMaven(globalMavenSettingsConfig: 'settings', jdk: '',
maven: '', mavenSettingsConfig: '', traceability: true) {
                    sh "mvn package && mvn deploy"
                }
            }
        }
```