Configure Jenkins TO build and deploy talend Project items

Contents

[1.Jenkins 2](#_Toc107918170)

[2.Configuring Jenkins to build and deploy project items 3](#_Toc107918171)

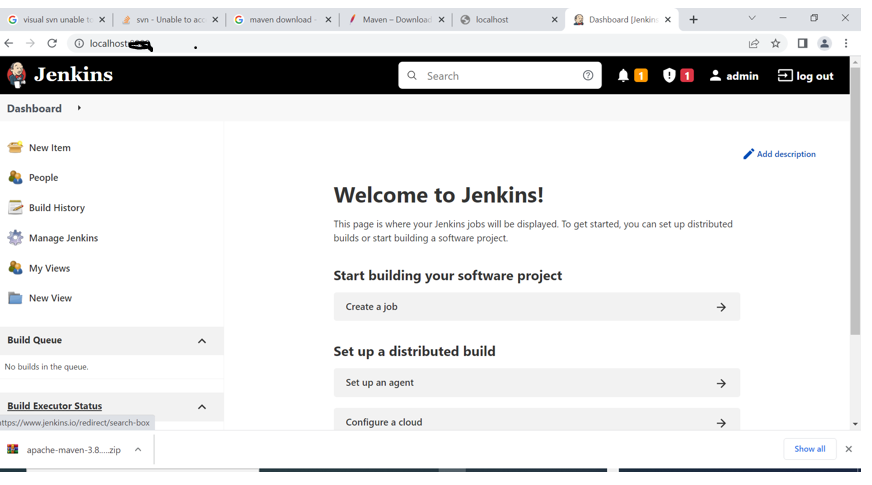
[2.1. Installing Jenkins plugins 3](#_Toc107918172)

[2.2. Storing login credentials in Jenkins 4](#_Toc107918173)

[2.3. Storing your custom Maven settings file in Jenkins 5](#_Toc107918174)

[2.4. Creating the Jenkins pipeline 11](#_Toc107918175)

# 1. Jenkins

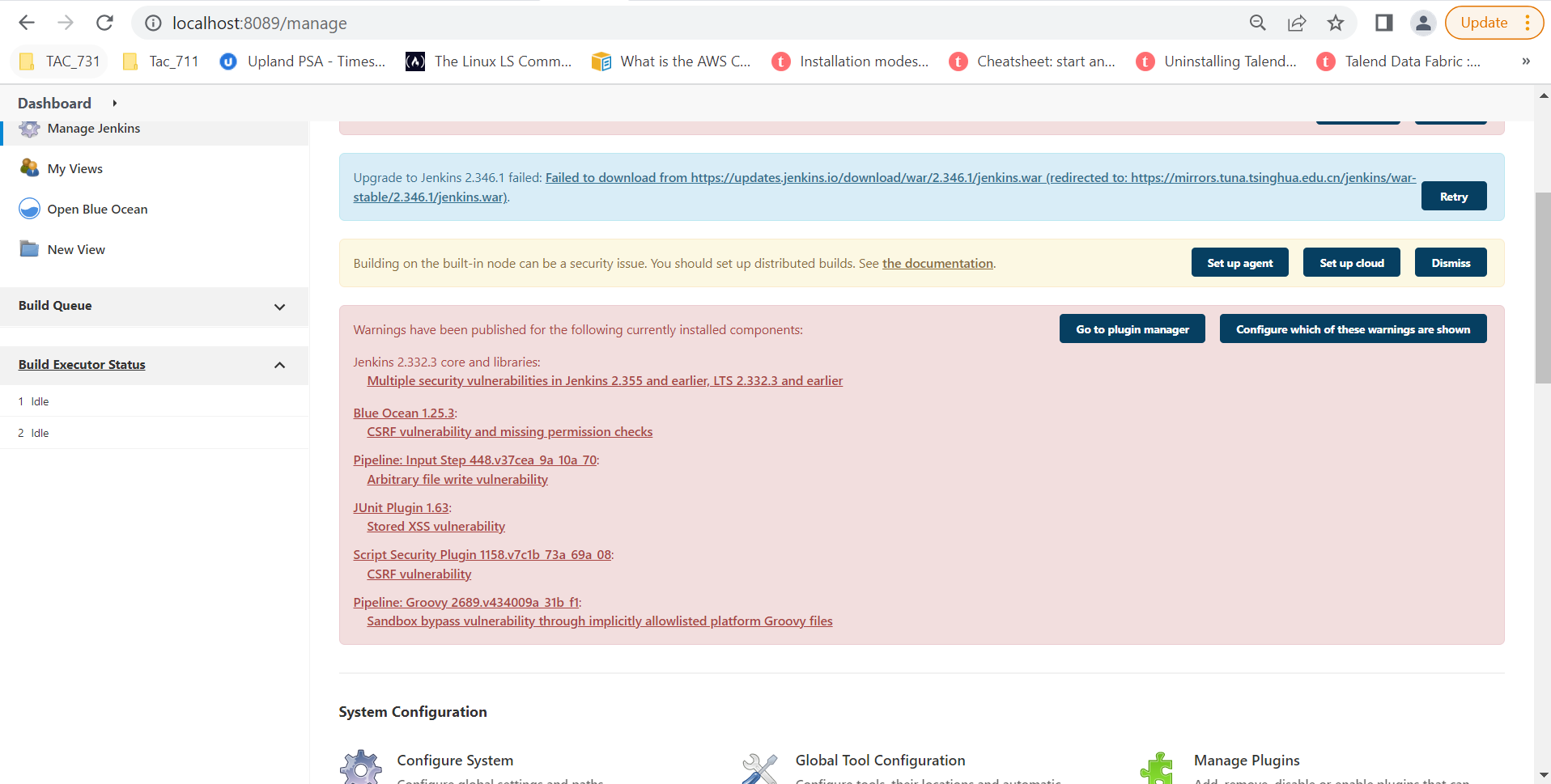


# 2. Configuring Jenkins to build and deploy project items

How to configure Jenkins to generate the sources of your artifacts (Jobs, Routes, Services, etc.) and

their related Test cases, to run all these artifacts as well as to deploy them on an artifact repository.

## 2.1. Installing Jenkins plugins



How to add the plugins necessary to execute your Pipelines in the Jenkins server.

**Procedure:**

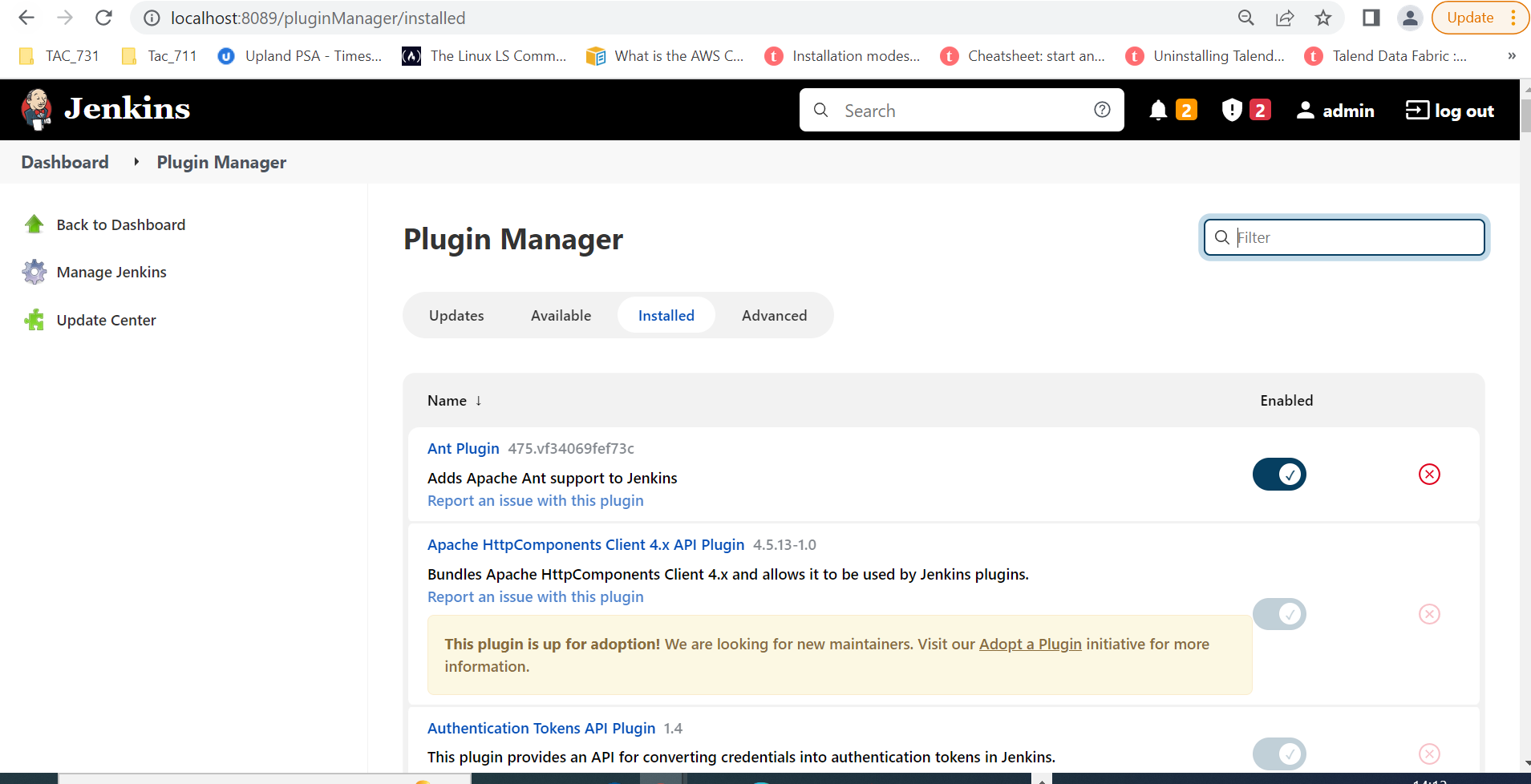
**1.** In Jenkins, navigate to **Manage Jenkins** > **Manage Plugins**.

**2.** Click the **Available** tab to see the available plugins.

**3.** Type in the following plugins in the **Filter** search bar and select them:

* **Blue Ocean** (to visualize in details the Pipeline workflow)
* **Configure File Provider** (to add your customized maven settings file to Jenkins)
* **Docker**
* **Git**
* **Maven Integration**
* **Pipeline** (set of tools for modelling scripted Continuous Deployment Pipelines in Jenkins)
* **Pipeline Maven Integration**

**4.** Click **Install without restart** to install all plugins at once without restarting the Jenkins server.



## 2.2. Storing login credentials in Jenkins

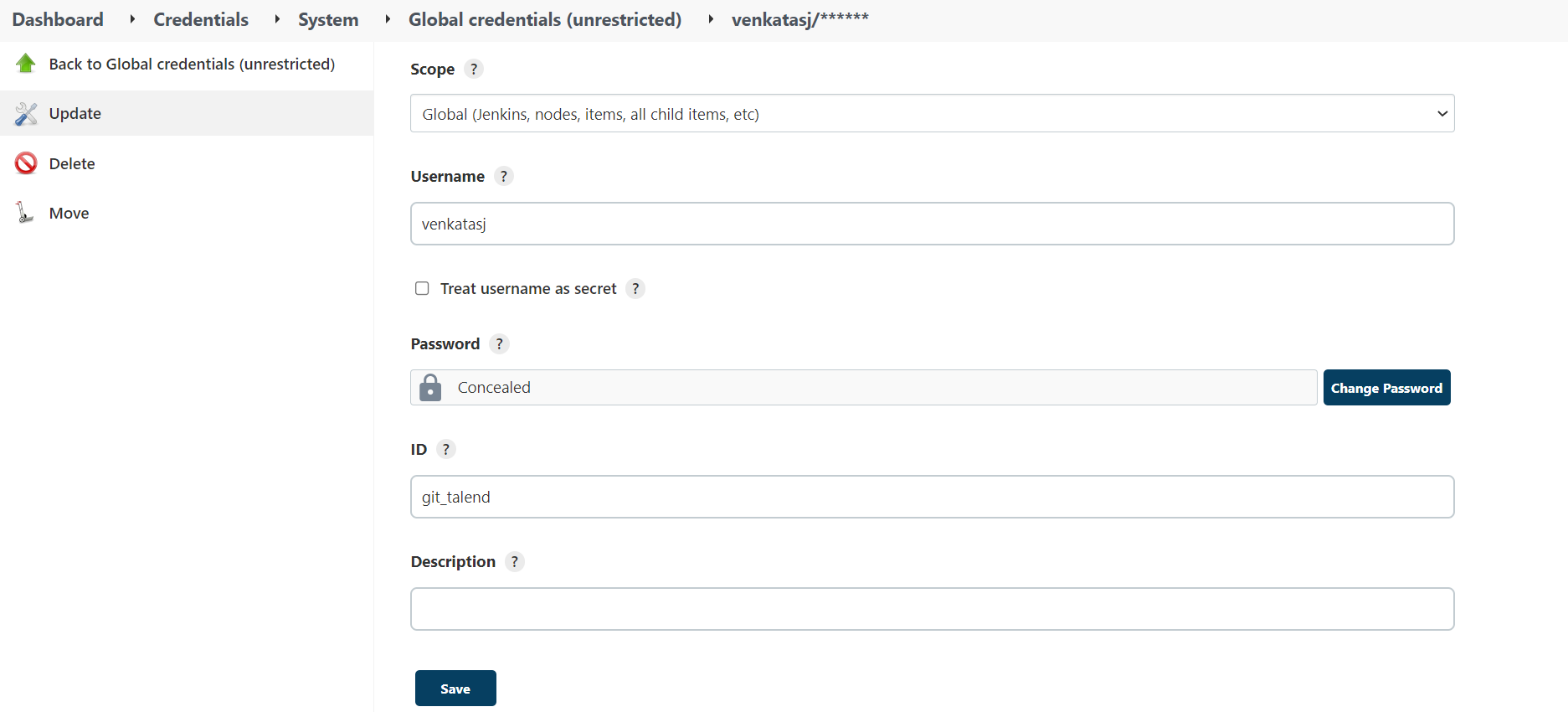
You need to store the login credentials of your source control management tool and deployment containers in Jenkins.

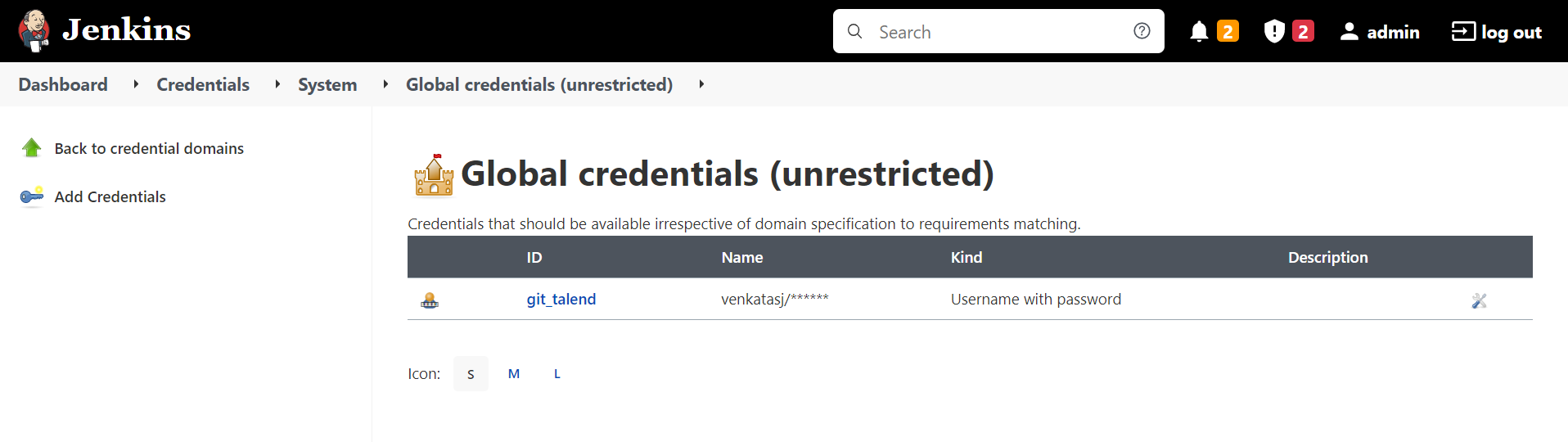
Steps for storing credentials

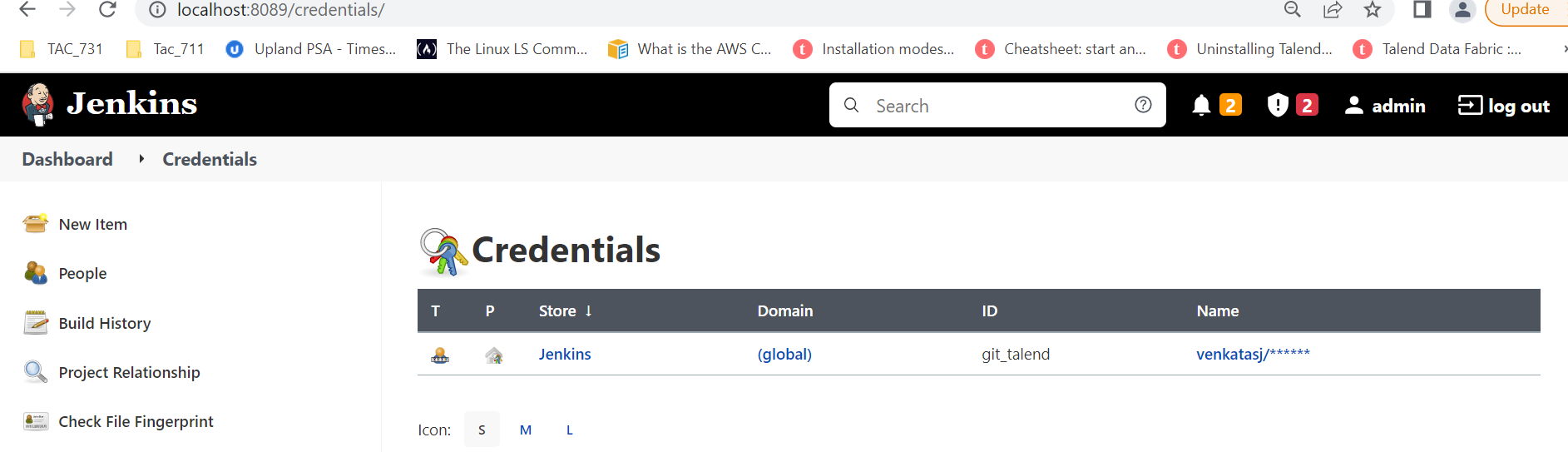
1. Navigate to **Manage Jenkins** > **Manage Credentials** > **Credentials**.

Click **System** > **Global credentials** > **Add Credentials** and add the Git credentials to access the repository where your project items (Jobs, Routes, Tests, etc.) are stored

3.Name its ID git\_talend, this name will be used in the pipeline script.

4.Click **Save** to take the changes into account.



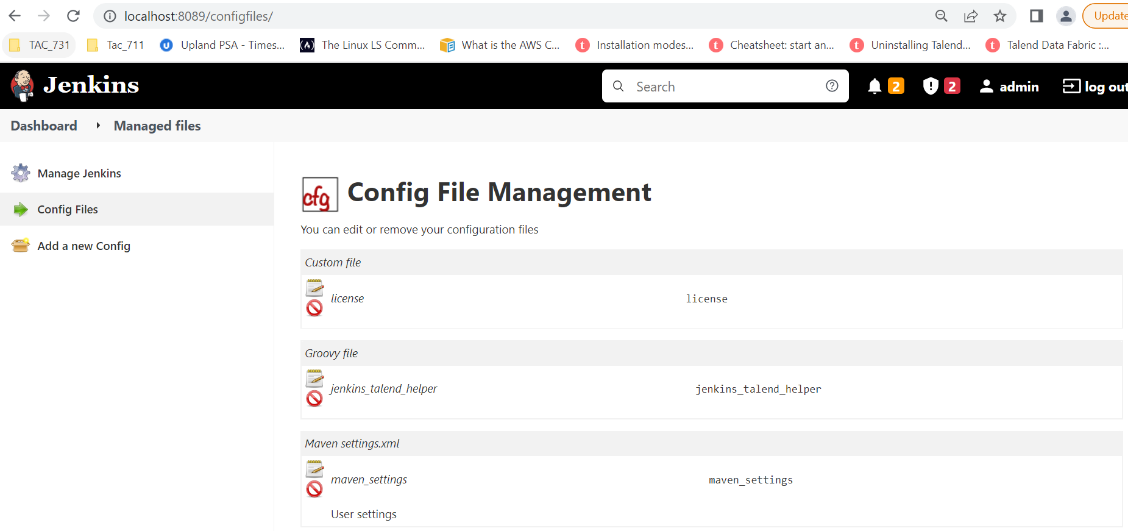


## 2.3. Storing your custom Maven settings file in Jenkins

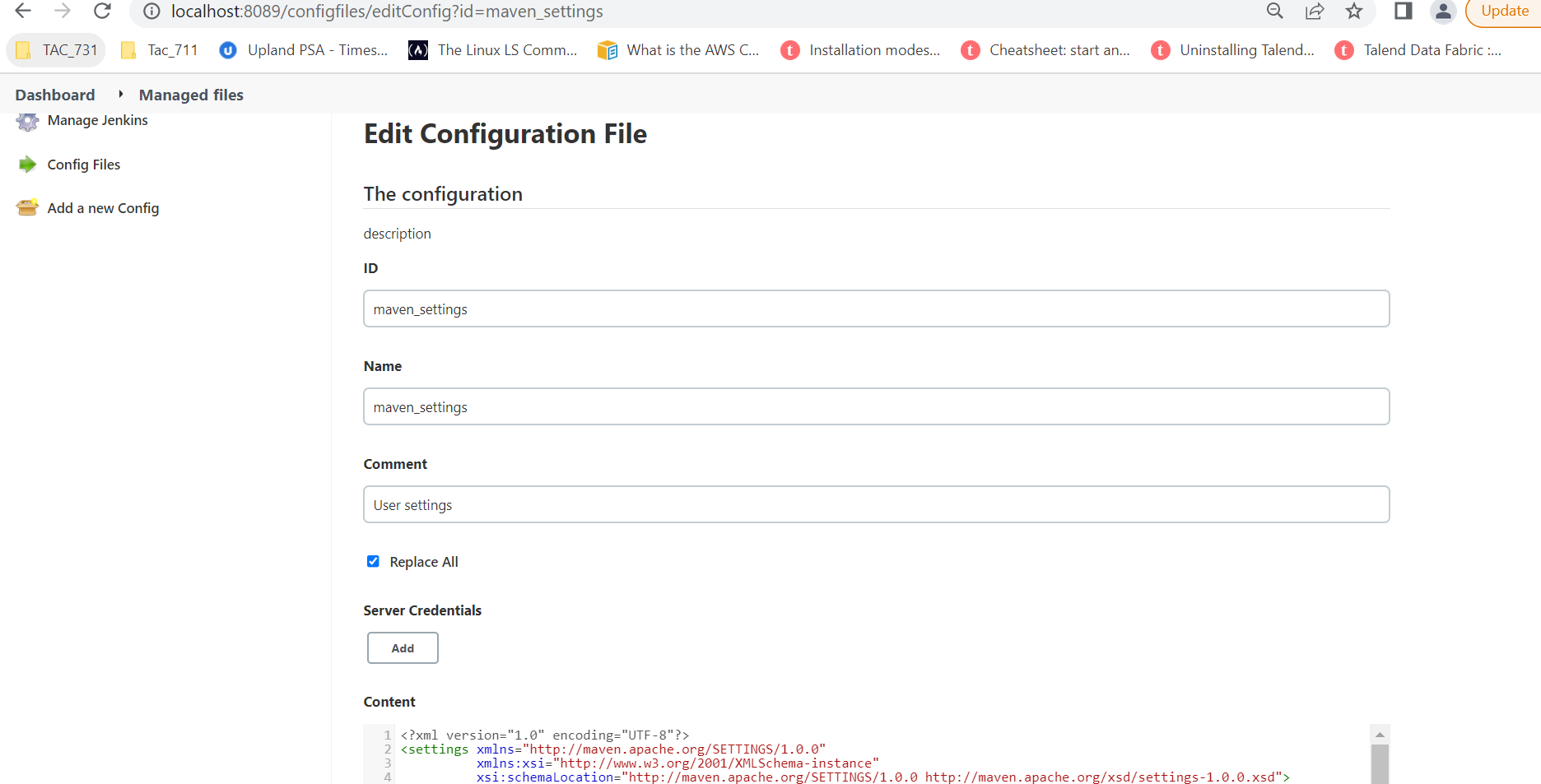
The maven\_settings.xml and jenkins\_talend\_helper.groovy files have been retrieved from the Downloads tab of this page.

**Procedure**

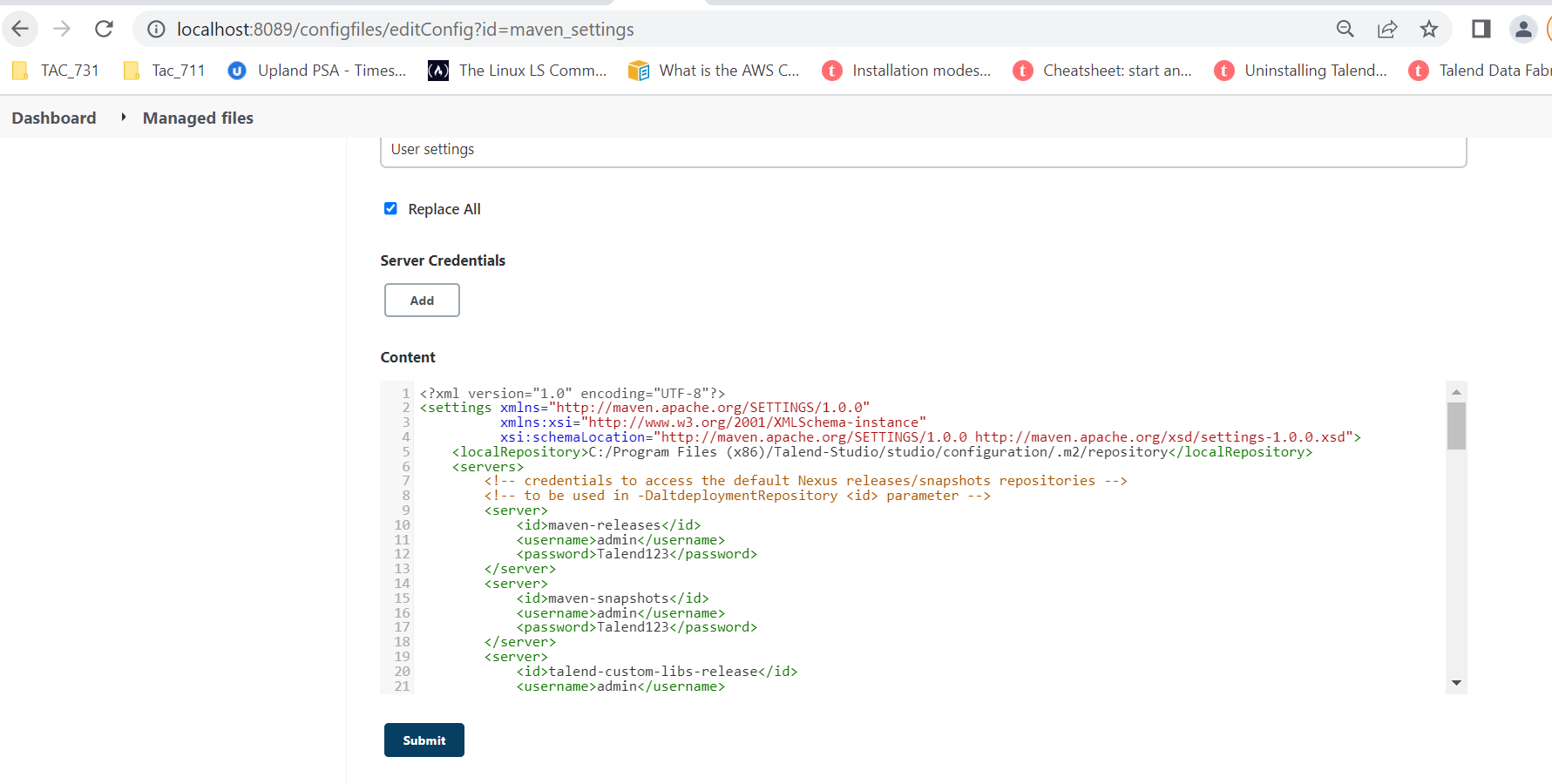
1. In Jenkins, navigate to **Manage Jenkins** > **Managed files**.
2. Click **Add a new config** and select **Maven settings** to add the Maven configuration file.
3. Give it a name and an ID, **maven\_settings**. This ID will later be used in the Jenkins [the pipeline script](https://help.talend.com/r/EJHl~PuPCuHT1YrPk7BfZg/FVE8gOngTQC4Y2NQqEvnEg).



1. In the **Content** editor, paste the content of the maven\_settings.xml file you have previously downloaded.



* 1. In maven settings.xml file local repository given as talend m2 repository path

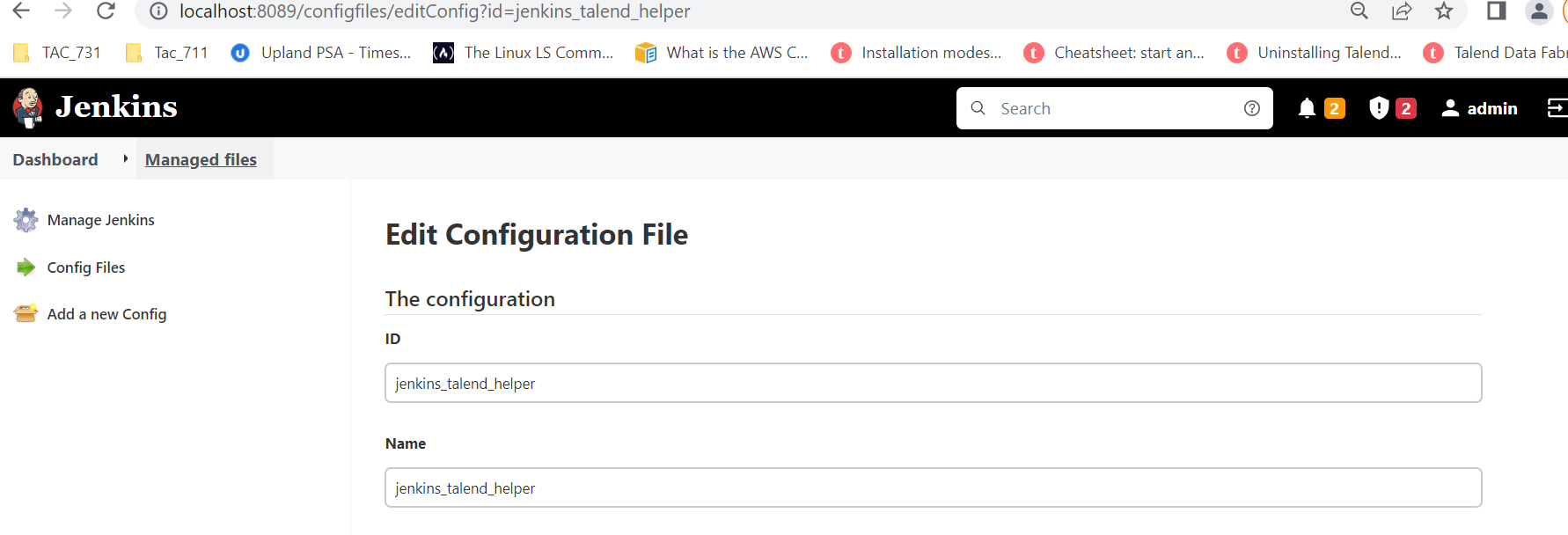




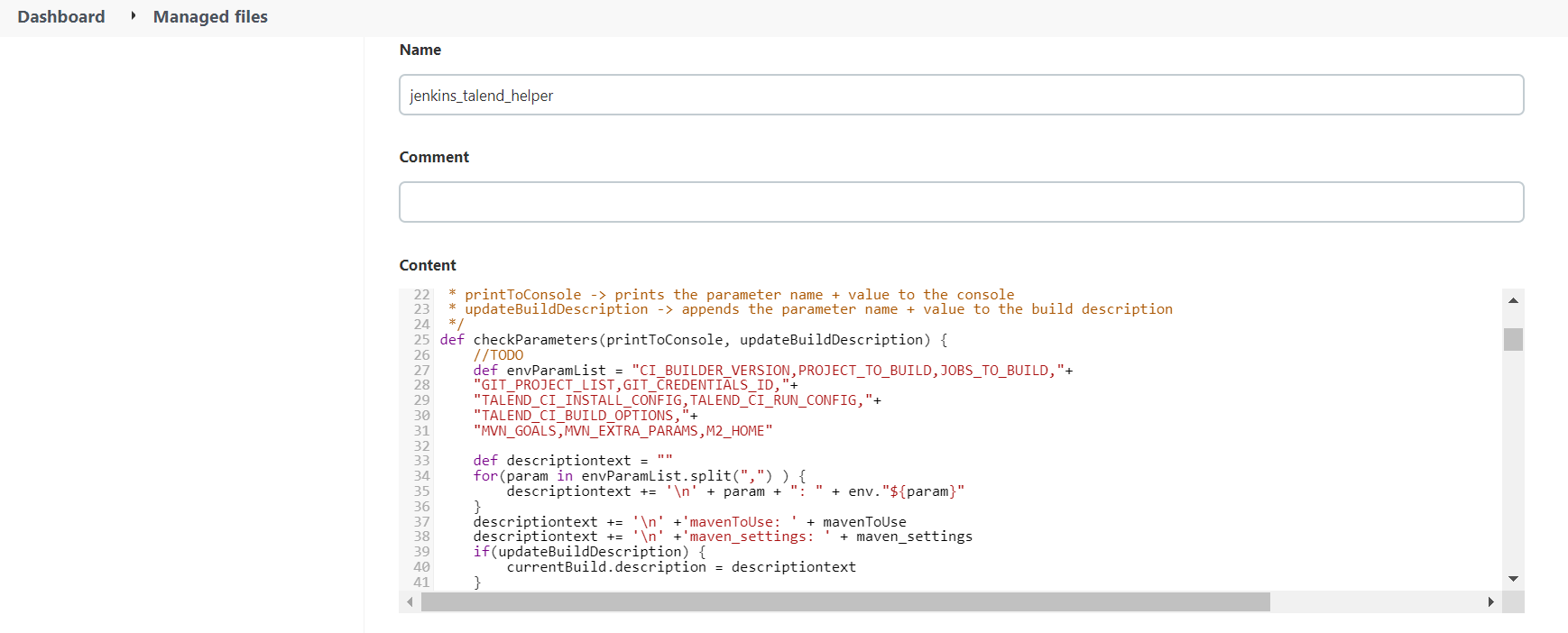
Click submit

Click **Add new config** and select **Groovy** to add Jenkins helper configuration file.

1. Give it a name and an ID, **jenkins\_talend\_helper**. This ID will later be used in the Jenkins pipeline script.



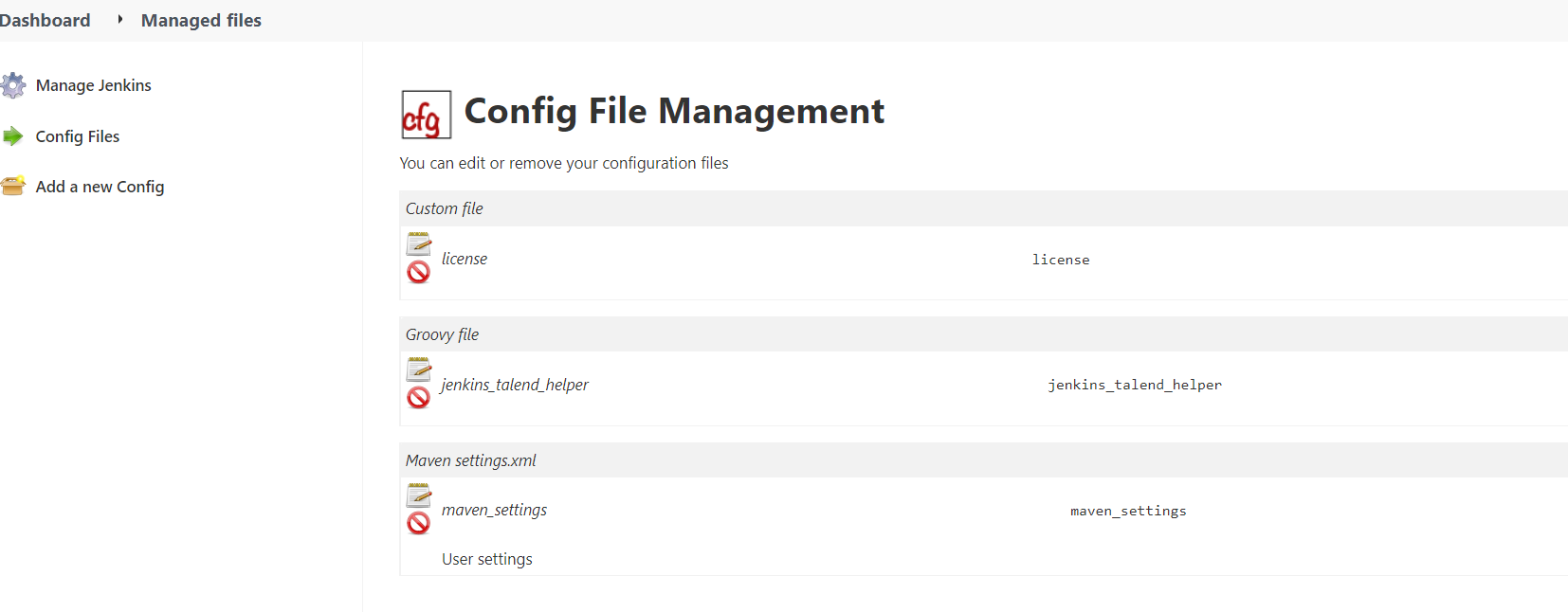
6.In the **Content** editor, paste the content of the Groovy file and click **Submit**.



Click submit

1. Click **Add a new config** and select **Custom** to add your Talend license file.
2. Give it a name and an ID, **license**. This ID will later be used in the Jenkins [pipeline script](https://help.talend.com/r/EJHl~PuPCuHT1YrPk7BfZg/FVE8gOngTQC4Y2NQqEvnEg)
3. In the **Content** editor, paste the content of the license file and click **Submit**.

The Result



These files contain the connection information to your artifact repositories, the definition of the pipeline steps as well as the license information. These files will be are referenced in the configuration of your Jenkins pipeline.

How to store the environment variables and paths to the third-party tools needed to run your Jenkins pipeline (JDK/OpenJDK, Maven).

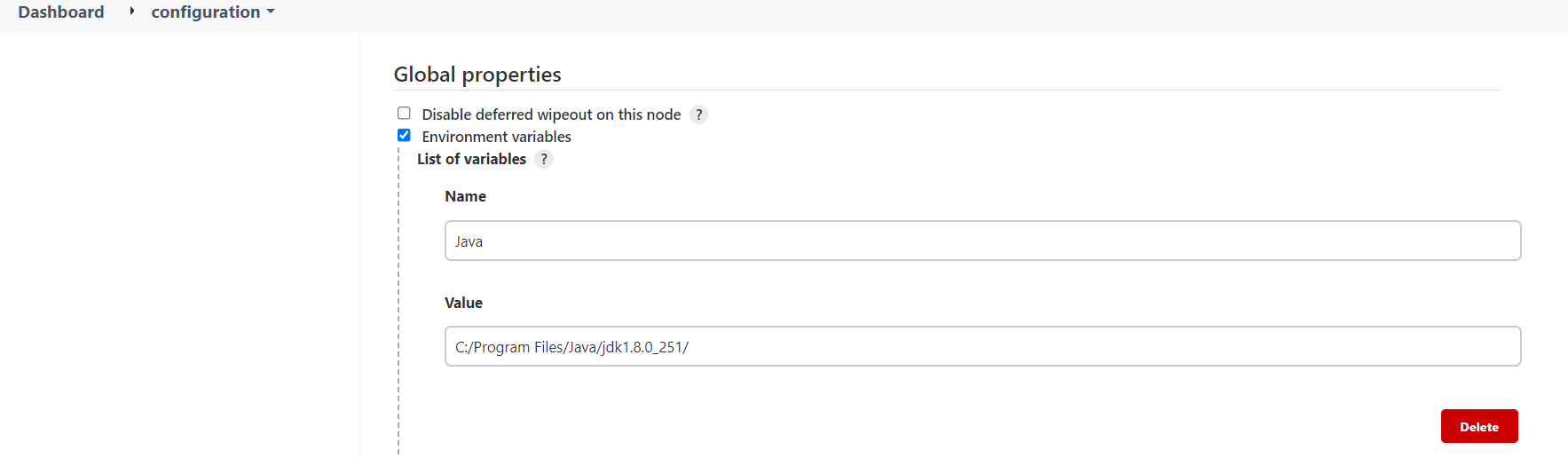
**Procedure**

1.Navigate to **Manage Jenkins** > **Configure System**.

**2.** In the **Global properties** > **Environment variables** area, add the environment variables

corresponding to the following tools:

• Java or OpenJDK

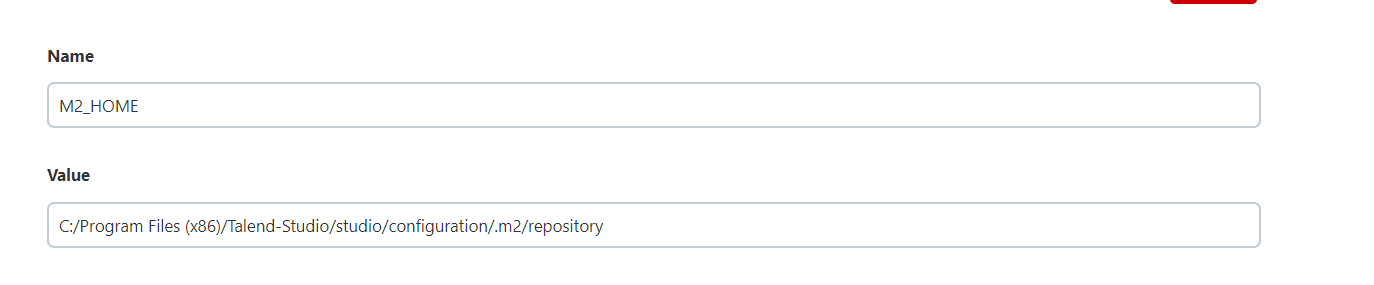


• Maven: name the variable MAVEN\_HOME, this name will be used in the pipeline script

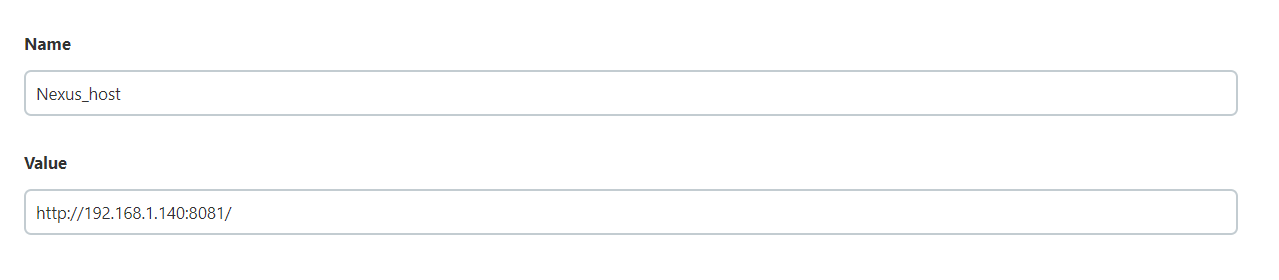


• M2 Maven repository: name the variable M2\_HOME, this name will be used in the pipeline

Script



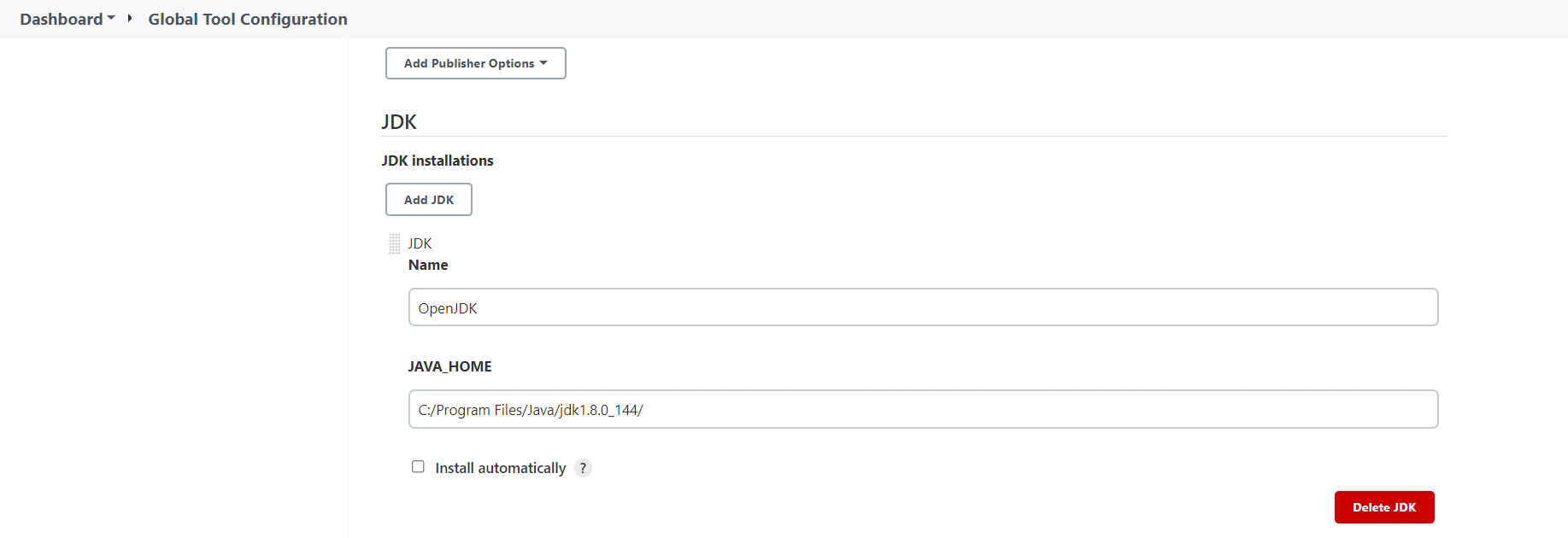
Name the variable Nexus\_host and given url of the nexus



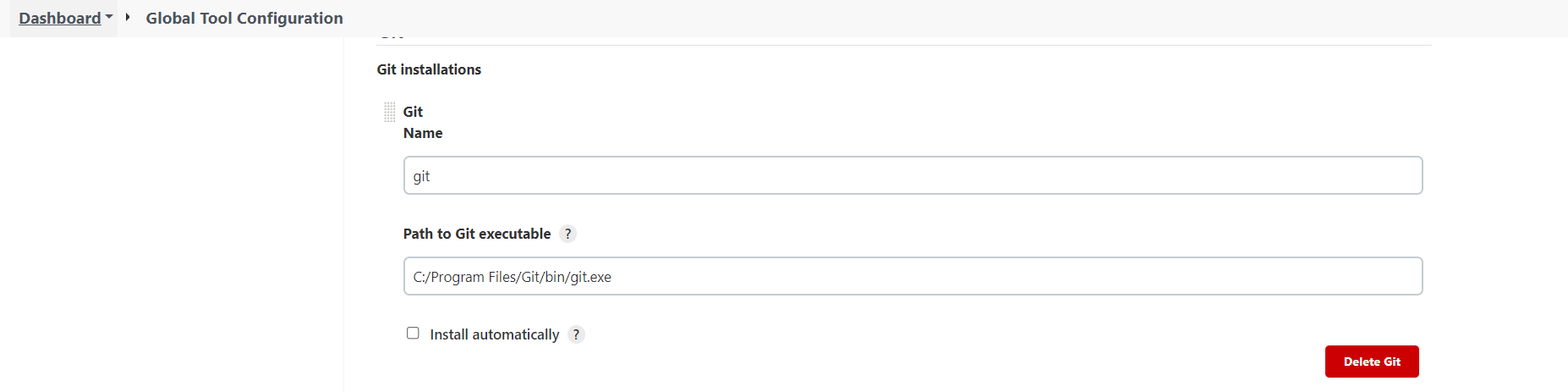
**Tip:** This variable value corresponds to the <studio\_path>/configuration/.m2/

repository folder.

In the **Global Tool Configuration** > JDK area, Enter the path to your JDK installation directory in the **JAVA\_HOME** field.



In the **Global Tool Configuration** >GIT area, add Git executable path

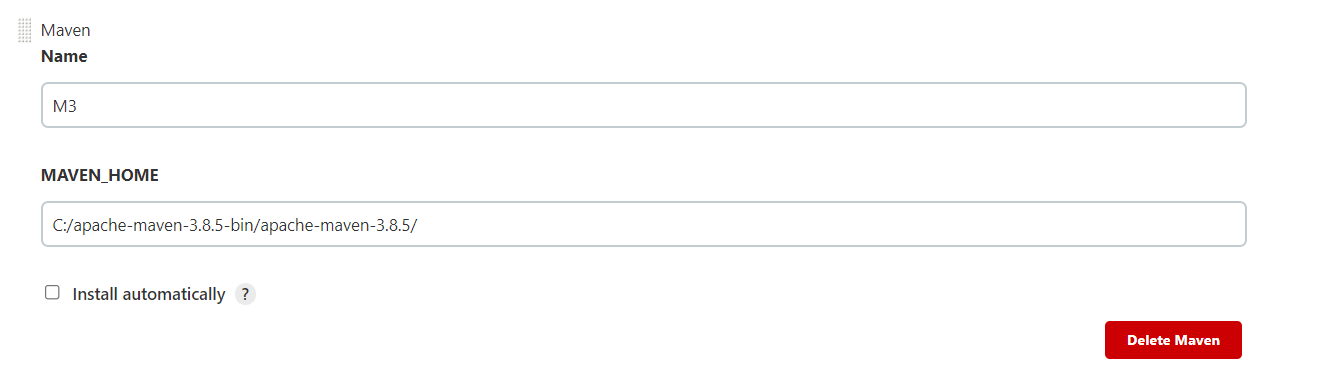


**3.** Click **Save** to take the changes into account.

**4.** In the **Global Tool Configuration** > **Maven** area, add the M3 Maven tool:

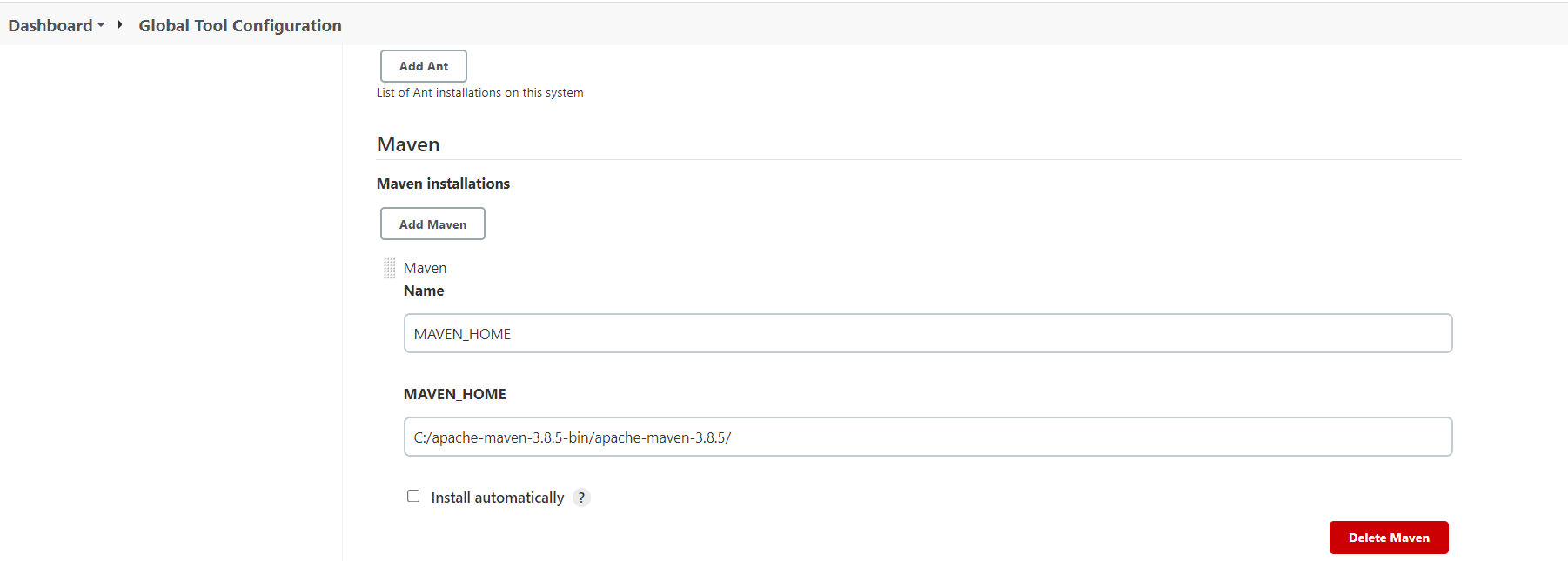
a) Click **Add Maven**.

b) Name the tool M3, this name will be used in the pipeline script.



c) Enter the path to your Maven installation directory in the **MAVEN\_HOME** field.

**5.** Click **Save** to take the changes into account.



2.4. Creating the Jenkins pipeline:

How to automatically create the build pipeline on the Jenkins server following the configuration defined in an XML file.

Before you begin:

1.The Jenkins\_pipeline\_sample.xml file has be downloaded from downloads tab of this url:

<https://help.talend.com/r/en-US/7.3/software-dev-lifecycle-best-practices-guide/storing-config-files-in-jenkins>

2.Through the <https://talend-update.talend.com/nexus/> we can download patch of the studio file and apply to patch to the studio

 The path Patch\_20220408\_R2022-04\_v1-7.3.1 archive has been downloaded and this path to the patch is used to configure the -Dpatch.path parameter in the jenkins\_pipeline\_simple.xml file.

3.The Talend\_Full\_Studio\_p2\_repository-20200219\_1130-V7 archive has been downloaded and this path of url <https://www.opensourceetl.net/tis/tpdsbdrt_731/Talend_Full_Studio_p2_repository-20200219_1130-V7.3.1.zip> is used to configure the -Dupdatesite.path parameter in the jenkins\_pipeline\_simple.xml file.

4.the jenkins\_pipeline\_simple.xml file and replace them with the values corresponding to your environment.

Note that you must use the Talend CI Builder version which is compatible to the Talend Studio version in your CI commands or pipeline scripts.

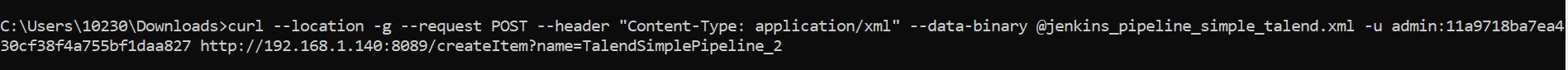
5. Open a command prompt window, browse to the folder where the jenkins\_pipeline\_simple.xml is located and execute the following Curl command to automatically create your pipeline:

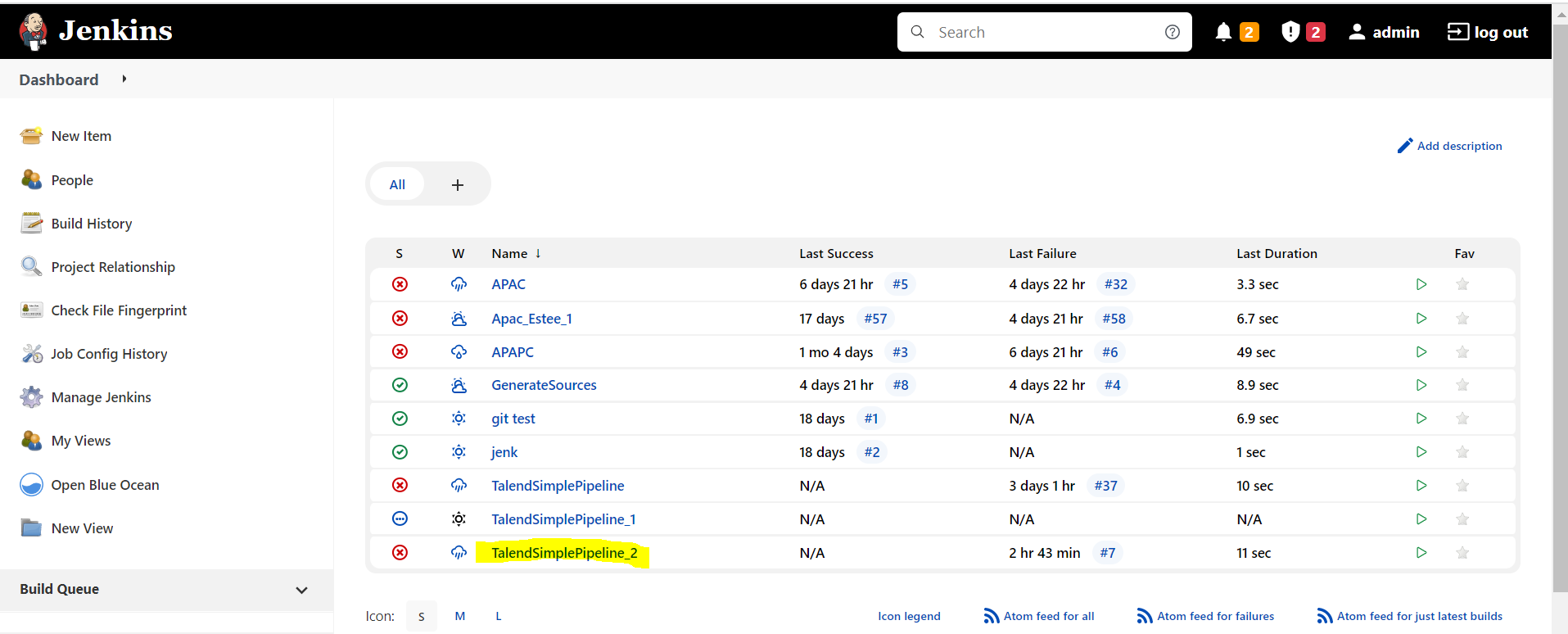
curl --location -g --request POST --header "Content-Type: application/xml" --data-binary @jenkins\_pipeline\_simple.xml -u <jenkins\_username>:<jenkins\_token> <jenkins\_url>/createItem?name=TalendSimplePipeline

 Make sure you replace the <jenkins\_username>, <jenkins\_token> and <jenkins\_url> with the values corresponding to your environment (username and API token needed to authenticate to Jenkins, and Jenkins server URL).

D:\EsteeLauder\CICD\CICD\_NEW>curl --location -g --request POST --header "Content-Type: application/xml" --data-binary @jenkins\_pipeline\_simple.xml -u admin:11a9718ba7ea430cf38f4a755bf1daa827 http://192.168.1.140:8089/createItem?name=TalendSimplePipeline

After executing of curl command in cmd..The talendSimplePipeline\_2 is created and displayed on our Jenkins dashboard





Running the Jenkins pipeline using parameters adapted to your environment

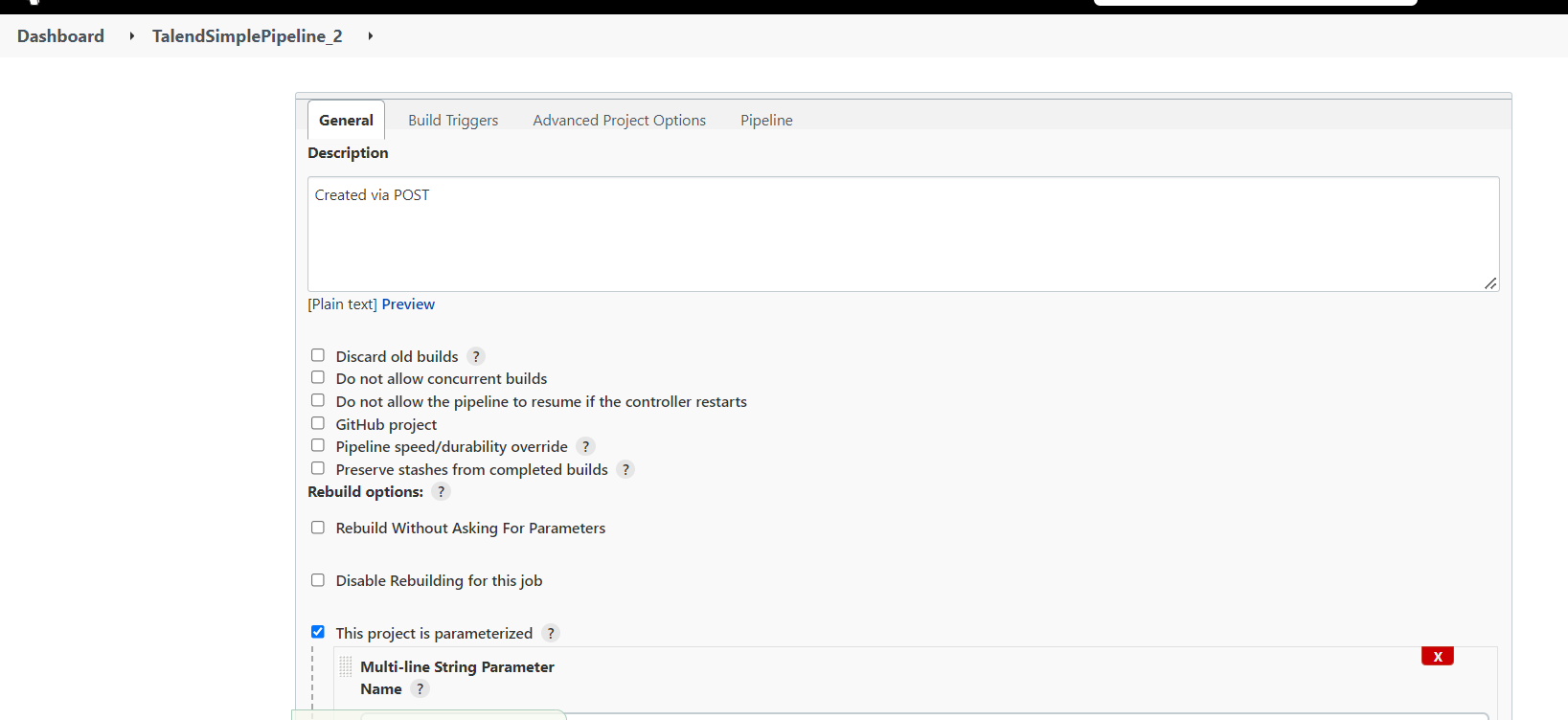
Configure the pipeline script provided by Talend to generate the project items according to the Maven phase you want to achieve and the repository you want to deploy your artifacts to

Some parameters come from the **maven\_settings file** you have previously defined and the **credentials** you have stored in Jenkins.

Procedure:

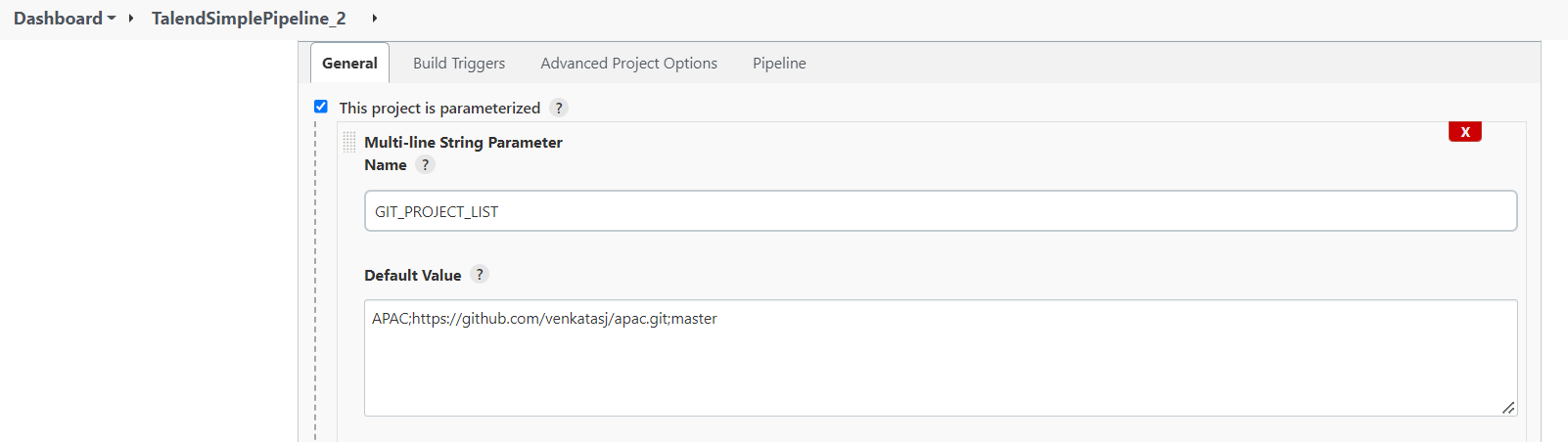
1.From the Jenkins home page ,Select **TalendSimplePipeline\_2**.

2.Select **Configure**



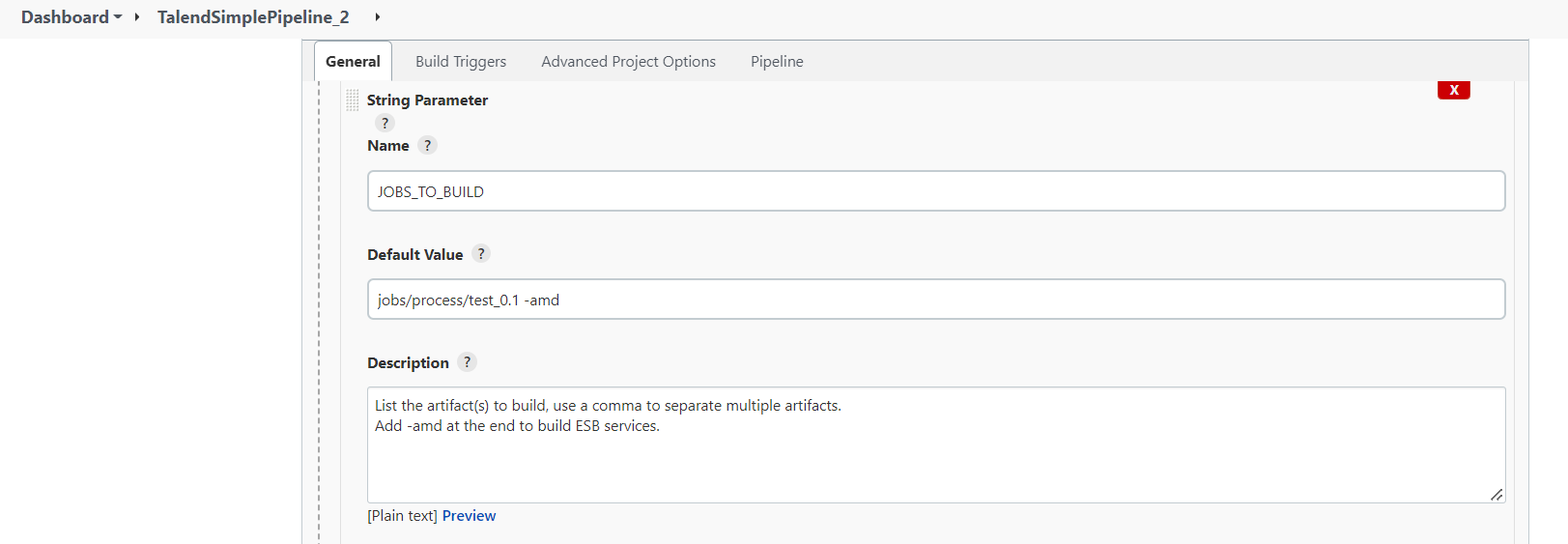
In the GIT\_PROJECT\_LIST .mention the git url ,project\_name and branch name

| **Parameter** | **Example of default value** |
| --- | --- |
| **GIT\_PROJECT\_LIST** | APAC;https://github.com/venkatasj/apac.git;master |



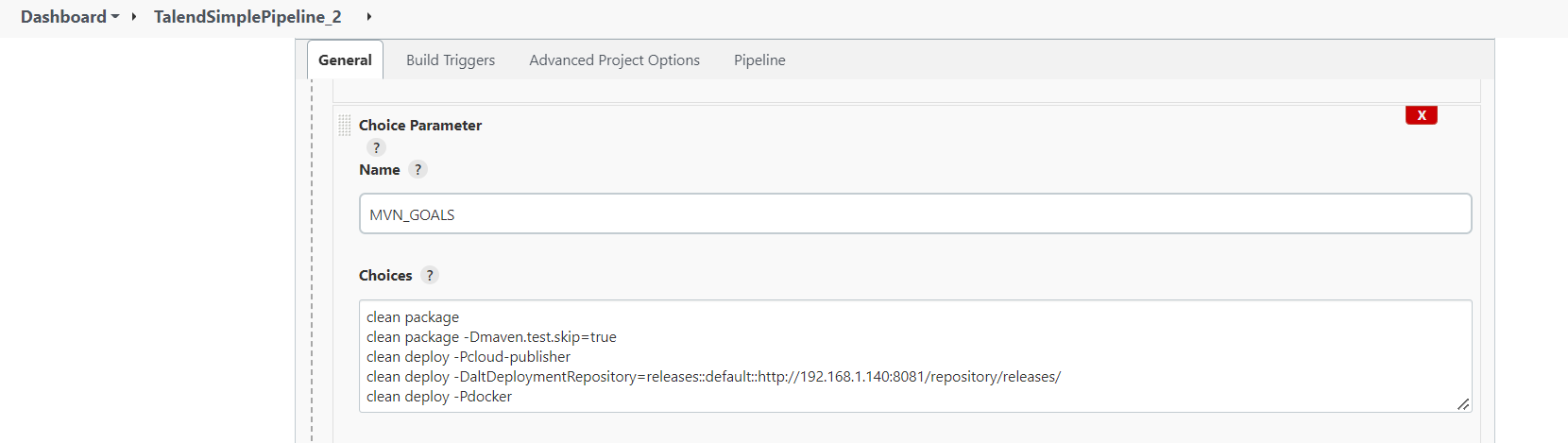
In the **JOBS\_TO\_BUILD** area, edit the commands and parameters that allow you to filter the Jobs according to our needs:

|  |  |
| --- | --- |
| **JOBS\_TO\_BUILD** | jobs/process/test\_0.1 -amd |



In The MVN\_GOALS and enter the name as MVN\_GOALS

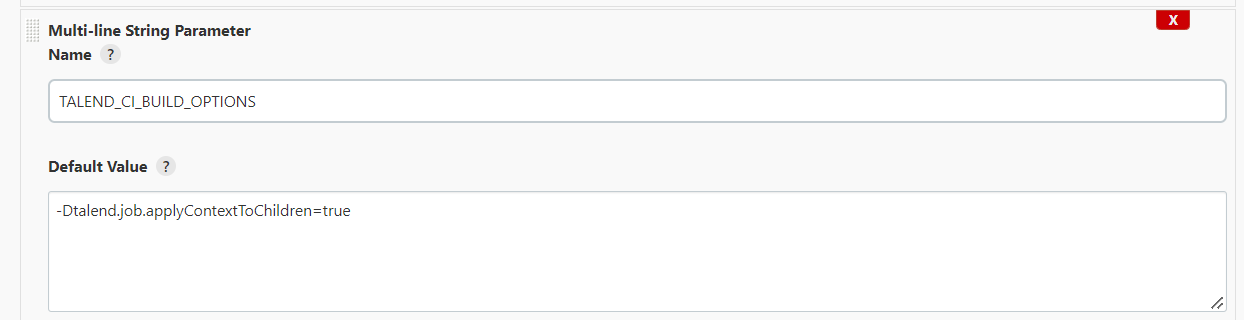
To define the artifact repository in which you want to deploy the project artifacts, edit the repository URL stated in the **MVN\_GOALS** parameters.



In TALEND\_CI\_RUN\_CONFIG mentioned name as TALEND\_CI\_RUN\_CONFIG and Make sure you are pointing to the latest Patch file in the -DPatch=path

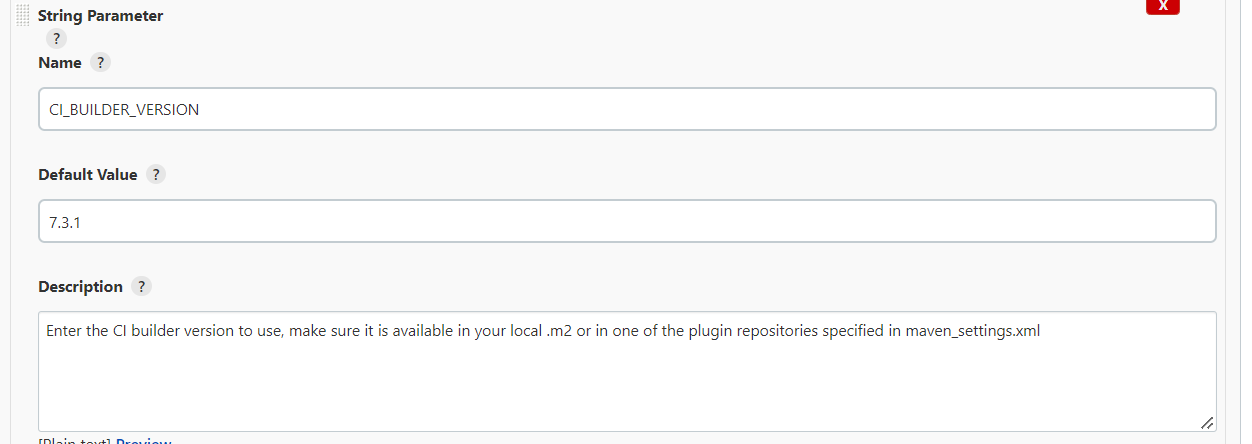


TALEND\_CI\_BUILD\_OPTIONS

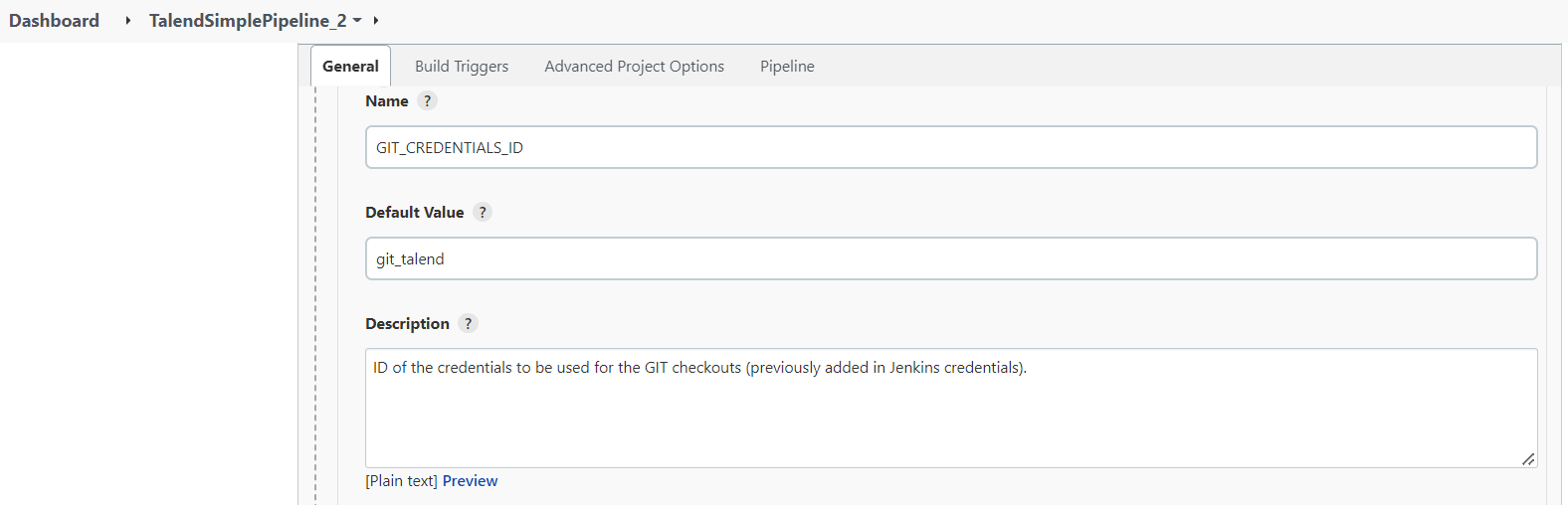


CI\_BUILDER\_VERSION

Make sure you have the latest version of the CI builder synced up in the repository and set the version accordingly

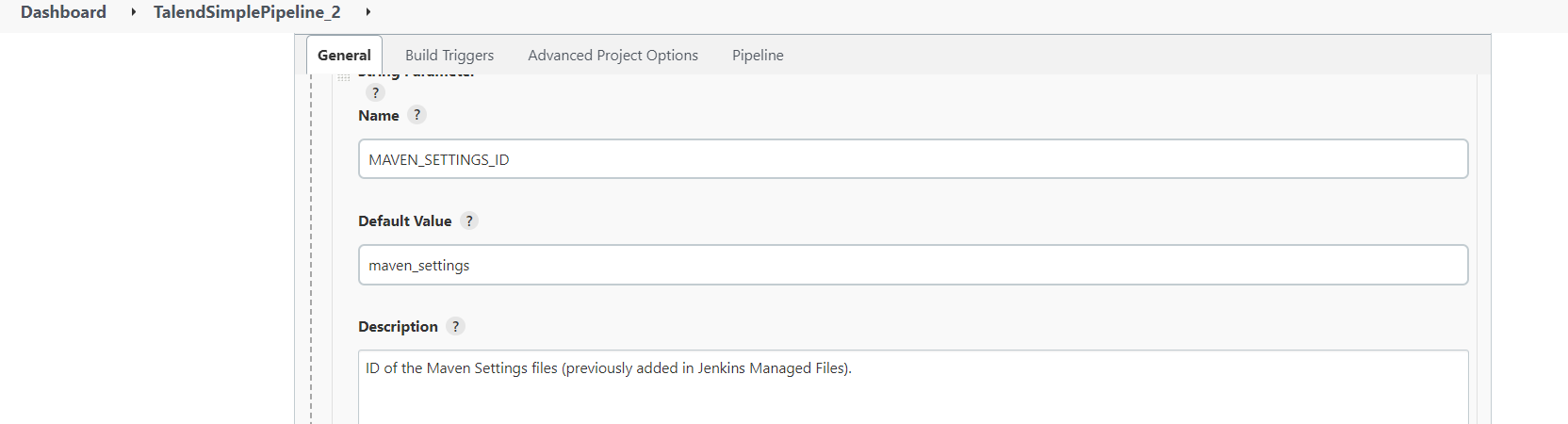


GIT\_CREDENTIONALS\_ID

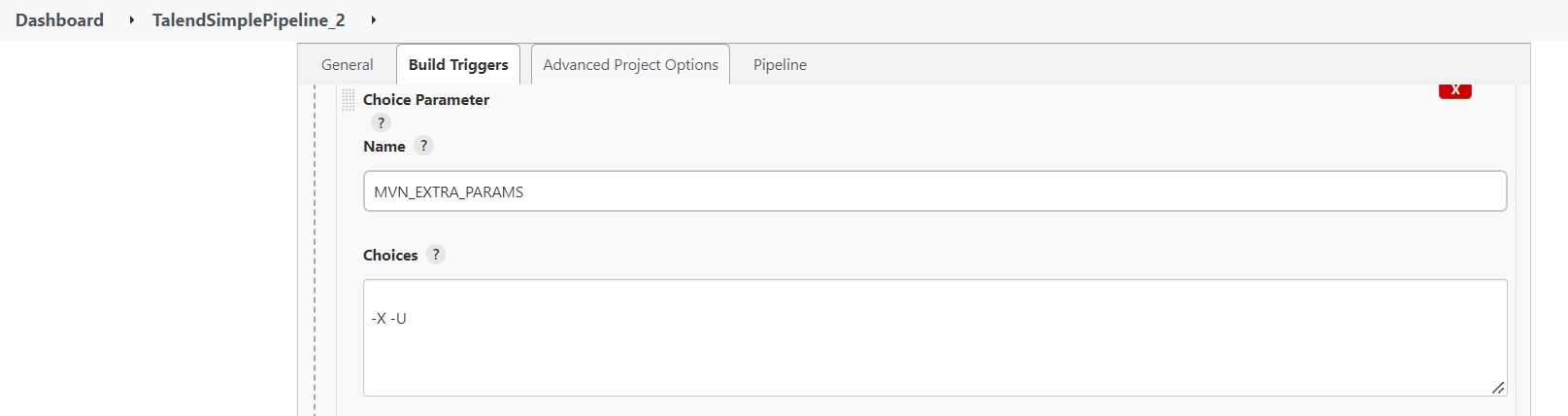


MAVEN\_SETTINGS\_ID

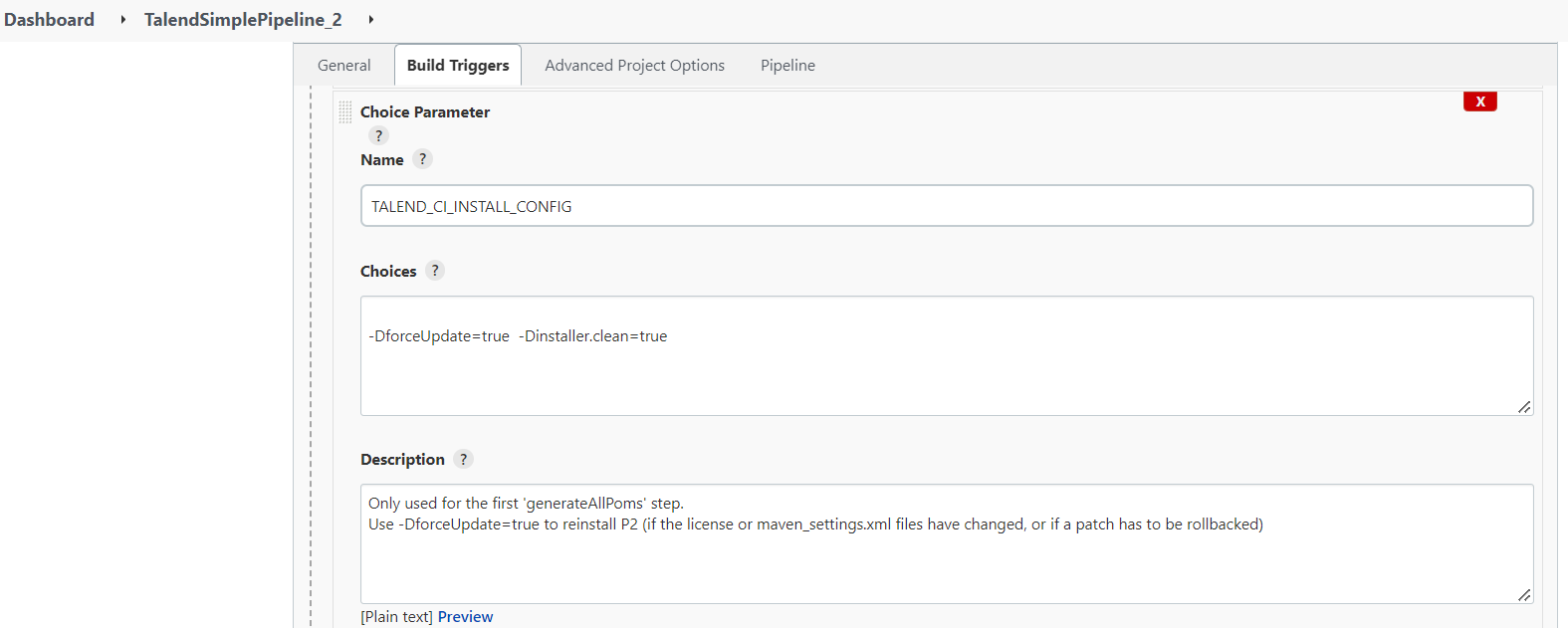
Make sure



MVN\_EXTRA\_PARMS



TALEND\_CI\_INSTALL\_CONFIG

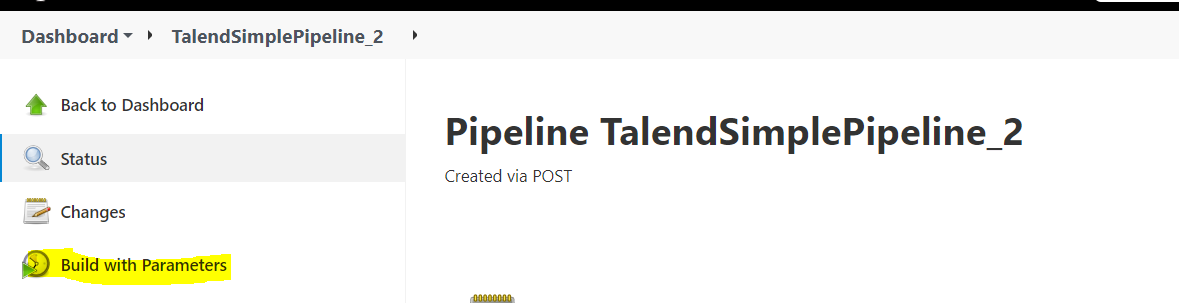


PIPELINE Script



Click on apply and save.

2.Click **Build With Parameters** on the left panel to open the pipeline configuration page.



3. Set your own values for the environment variables defined in the script (Git project name and branch, Job name/version/type, Maven goals, repository URL, etc.).