Contents

[Login 2](#_Toc9168208)

[Main Screen 2](#_Toc9168209)

[Plugin Installation 3](#_Toc9168210)

[Veracode-scanner plugin 4](#_Toc9168211)

[XL Deploy plugin 4](#_Toc9168212)

[Maven pipeline integration plugin 4](#_Toc9168213)

[Version number plugin 4](#_Toc9168214)

[Create a new pipeline 5](#_Toc9168215)

[Copy a pipeline from an existing one 7](#_Toc9168216)

[Create a webhook (github) 10](#_Toc9168217)

[Folders and files needed 12](#_Toc9168218)

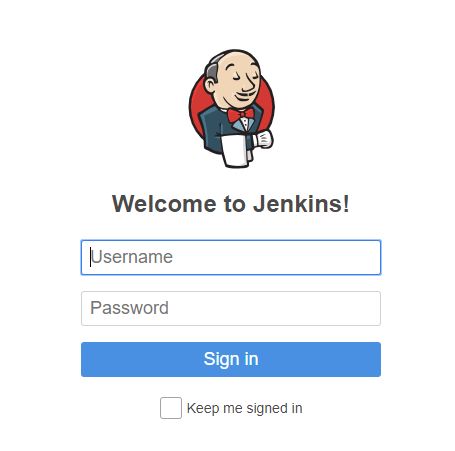
[Modifying template Values 13](#_Toc9168219)

[Update jenkins file 13](#_Toc9168220)

## Login

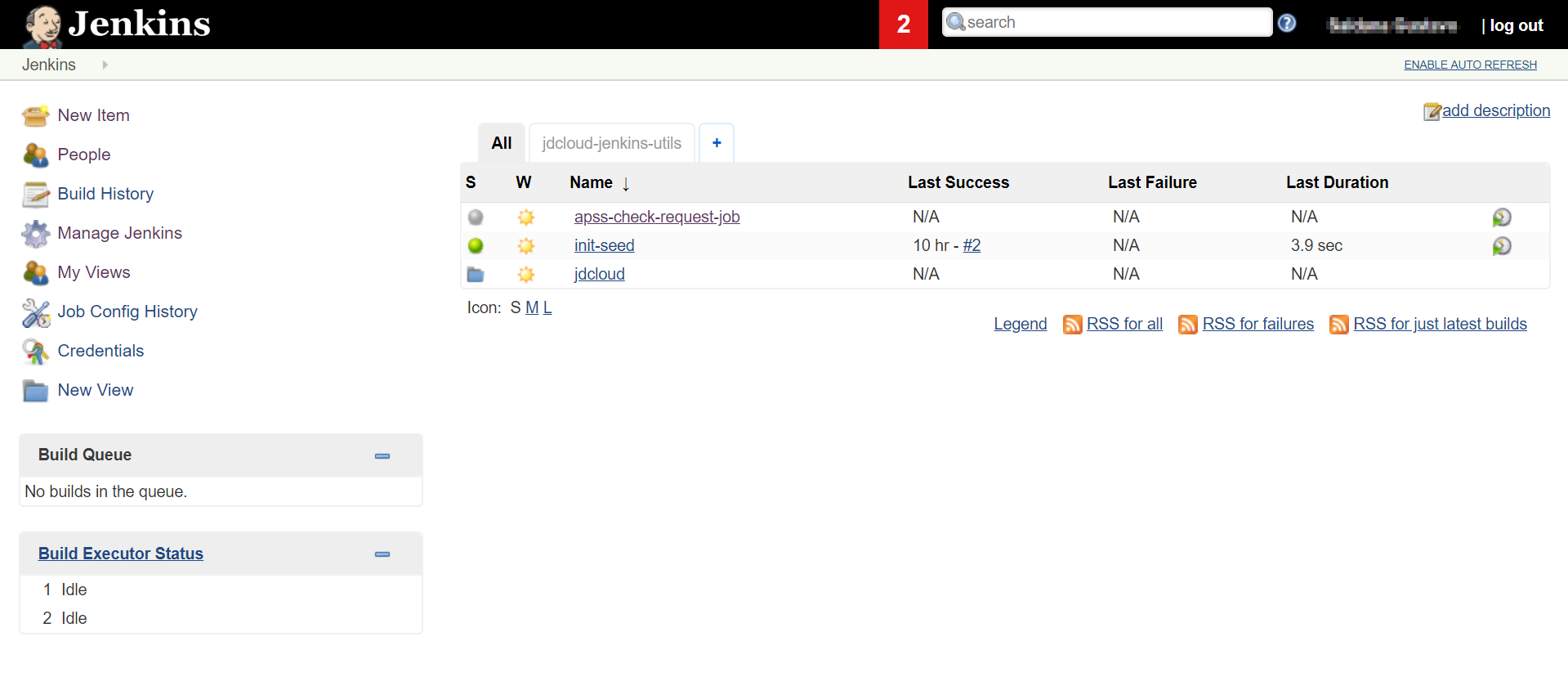
For login into Jenkins server go to the next URL and enter your RACFID and password:

<https://ci-scsdevelopment.svc.us.i00.c01.johndeerecloud.com/>



### Main Screen

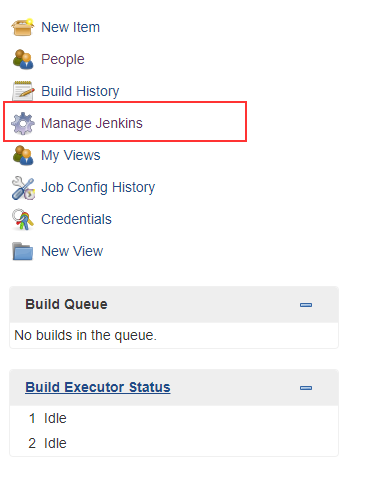
After login, you will be able to see the following screen

****

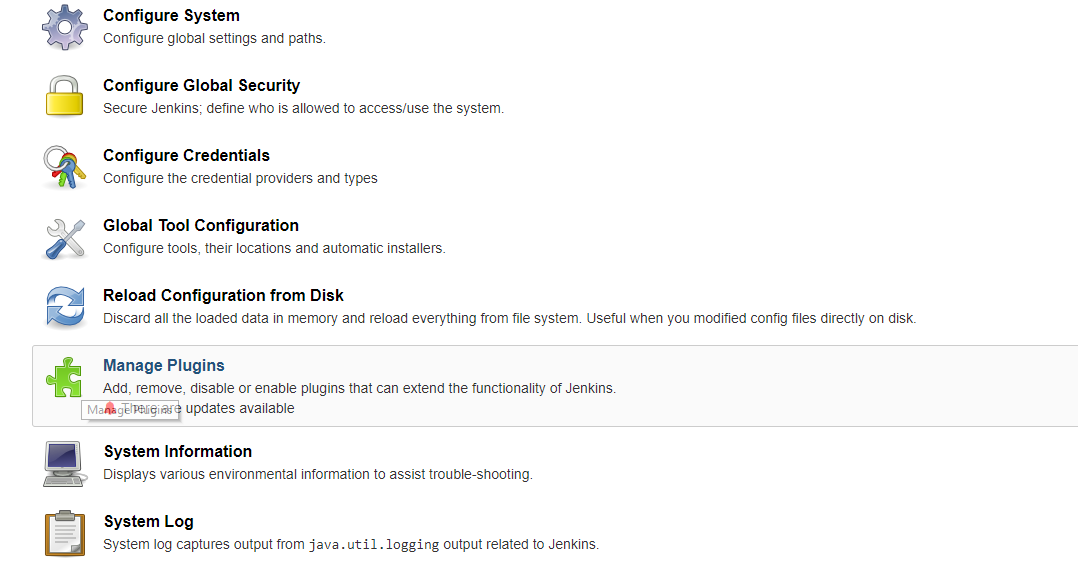
## Plugin Installation

For install a new plugin for Jenkins service you must follow the next steps:

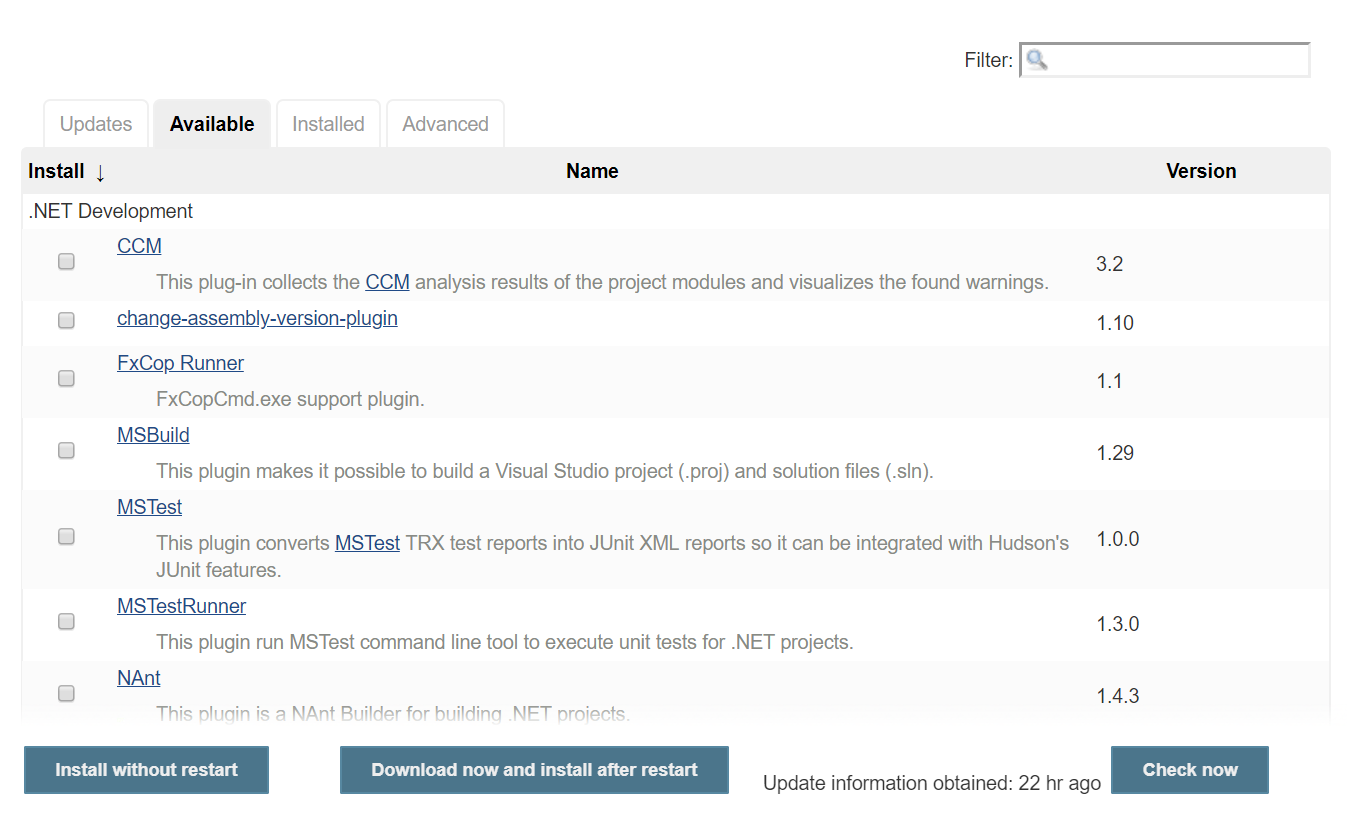
* In the main screen, go to **Manage Jenkins**



* Then click on the option **Manage Plugins**

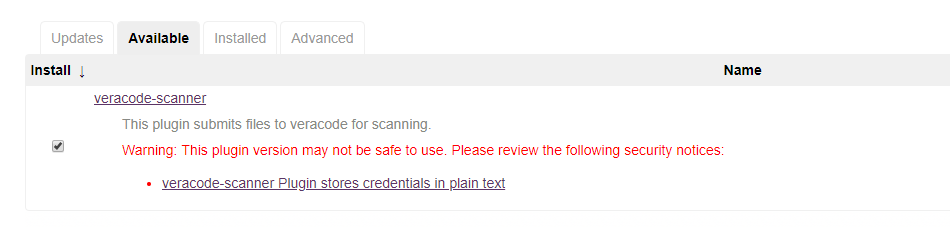


* You will see the following screen:



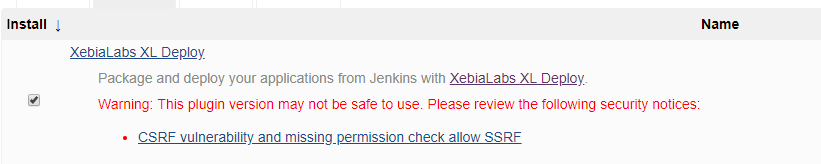
### Veracode-scanner plugin

In the plugin manager screen, click on the tab **Available** and in the filter textbox type in the word **veracode** and select the following plugin:



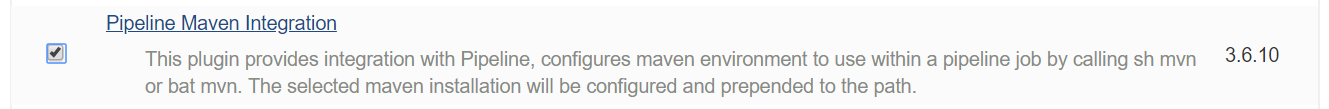
### XL Deploy plugin

In the plugin manager screen, click on the tab **Available** and in the filter textbox type in the word **xebialabs** and select the following plugin:



### Maven pipeline integration plugin

In the plugin manager screen, click on the tab **Available** and in the filter textbox type in the word **pipeline** **maven** and select the following plugin:

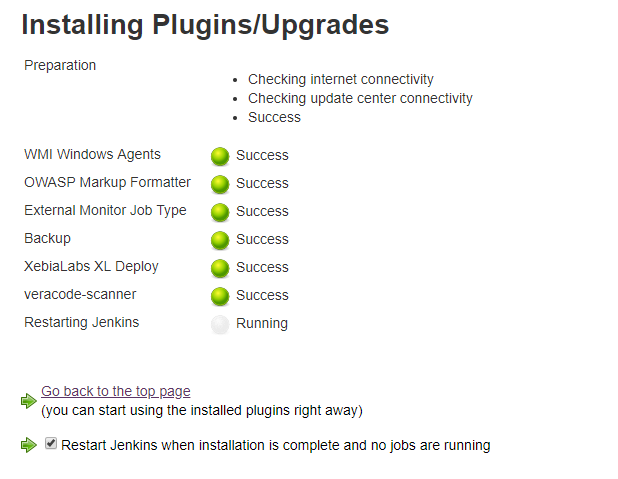


### Version number plugin

In the plugin manager screen, click on the tab **Available** and in the filter textbox type in the word **version number** and select the following plugin:

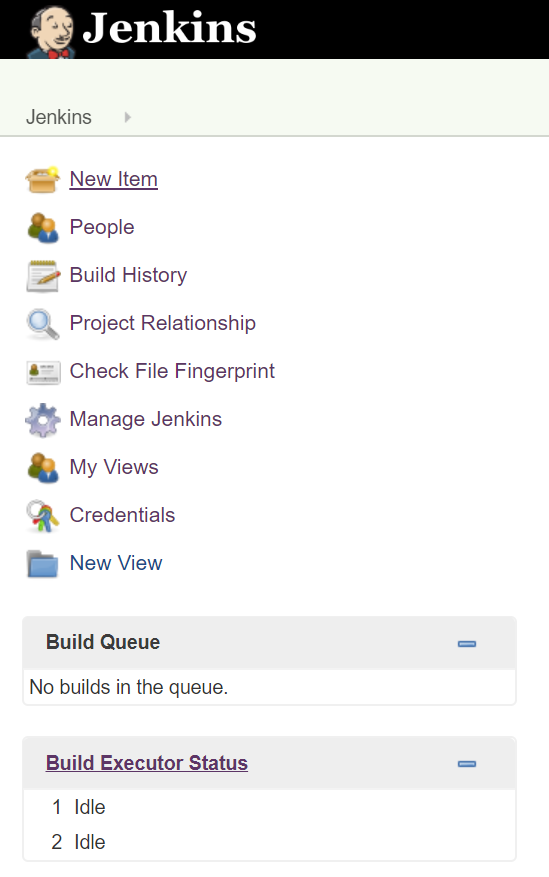


Once you have selected all the plugins and the installation has started, you should see the next screen:

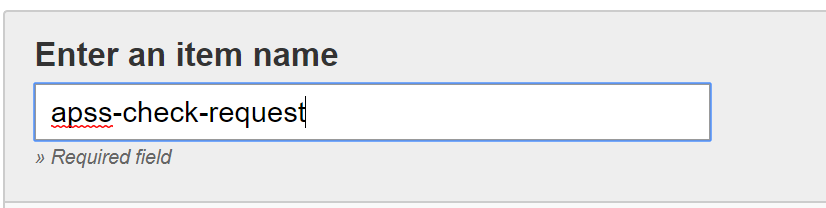


## Create a new pipeline

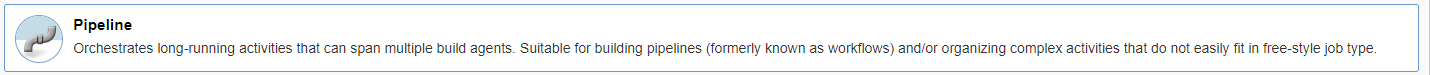
In the main menu, click on **“New Item”** option.



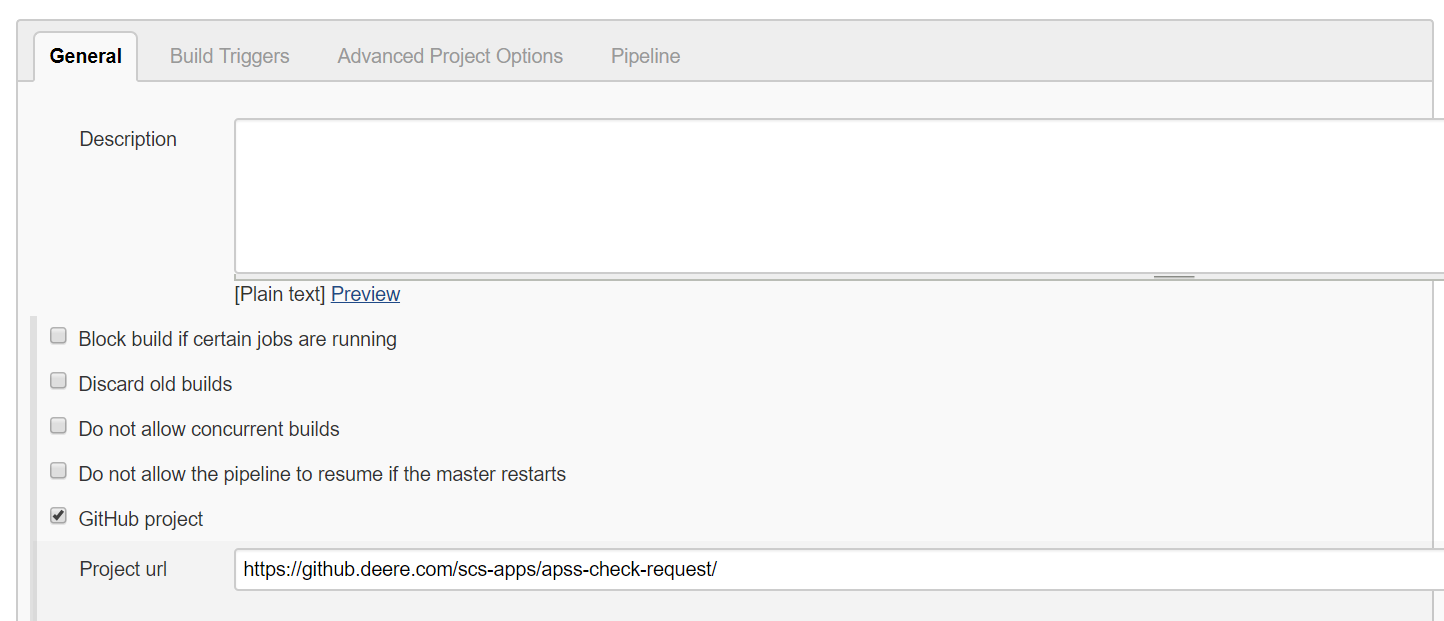
Enter the name for the pipeline



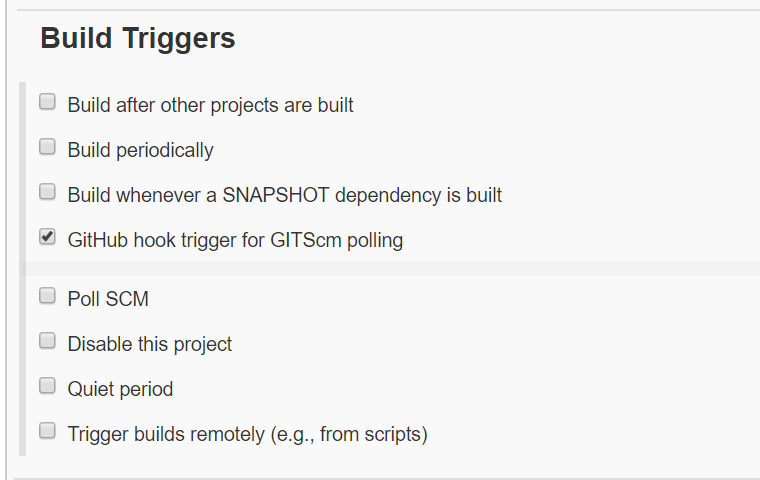
Select the **“Pipeline”** option and then **OK** button.



In the pipeline configuration select that is a Github project and type in the repository url.



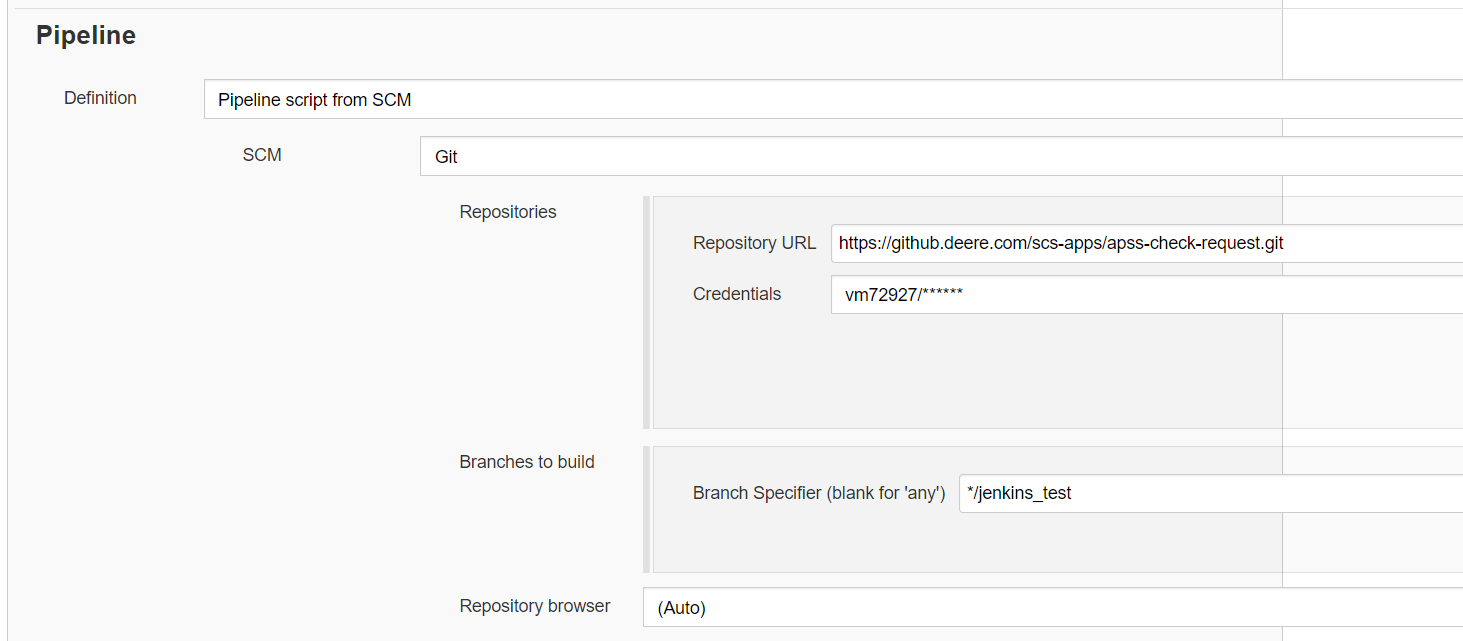
In the build trigger section select the “**Github hook trigger for GITScm polling**” option.



Note: See **How to** [Create a webhook](#_Create_a_webhook)

In the “**Pipeline**” section you just need to write the following information.

* Definition: Pipeline script from SCM
* SCM: Git
* Repositories: The repository URL
* Credentials: The configured credentials for Jenkins server
* Branches to build: All the branches that you want to build with this pipeline (in this case it’s **jenkins\_test**)
* Repository browser: Auto



And also, you have to write the path where (and name) of the file that you’re going to use to build the pipeline.

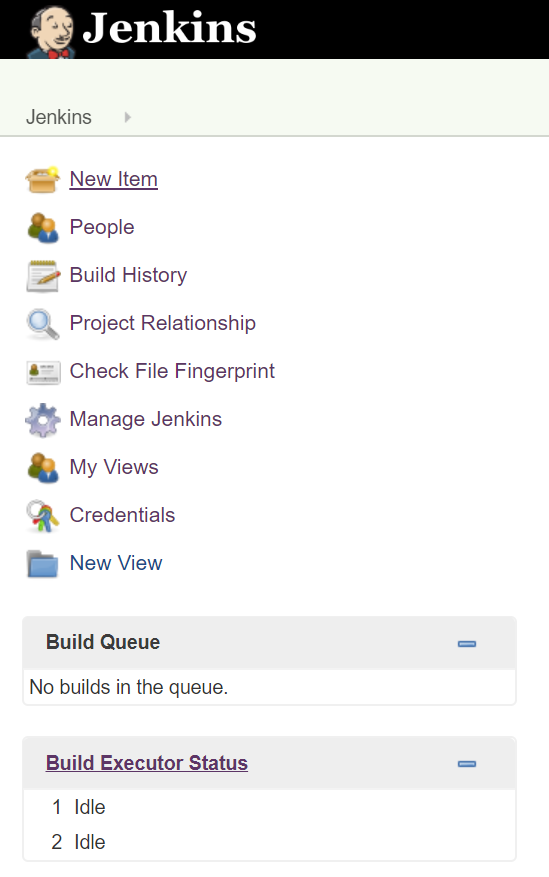
In this example the file is in the root of the repository and its name is “**jenkinsfile**”.



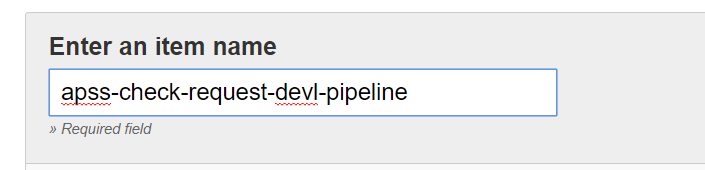
## Copy a pipeline from an existing one

#### Test environment (Devl)

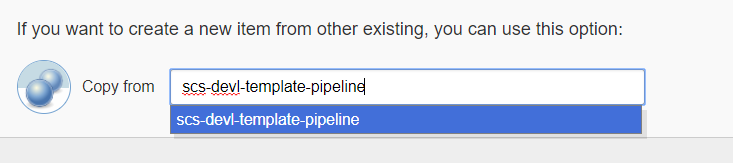
In the main menu, click on **“New Item”** option.



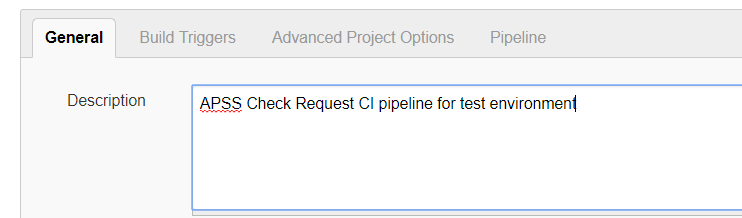
Enter the name for the pipeline



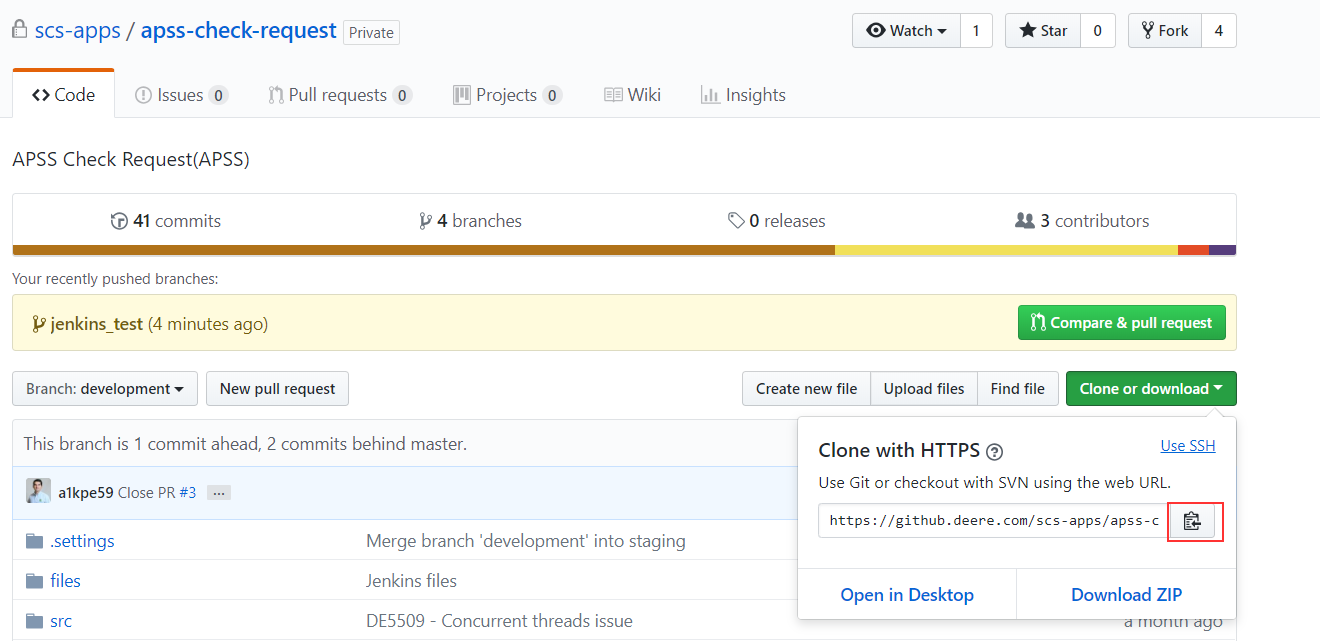
Then type in the name of the pipeline that you want to clone and press the “**OK**” button.



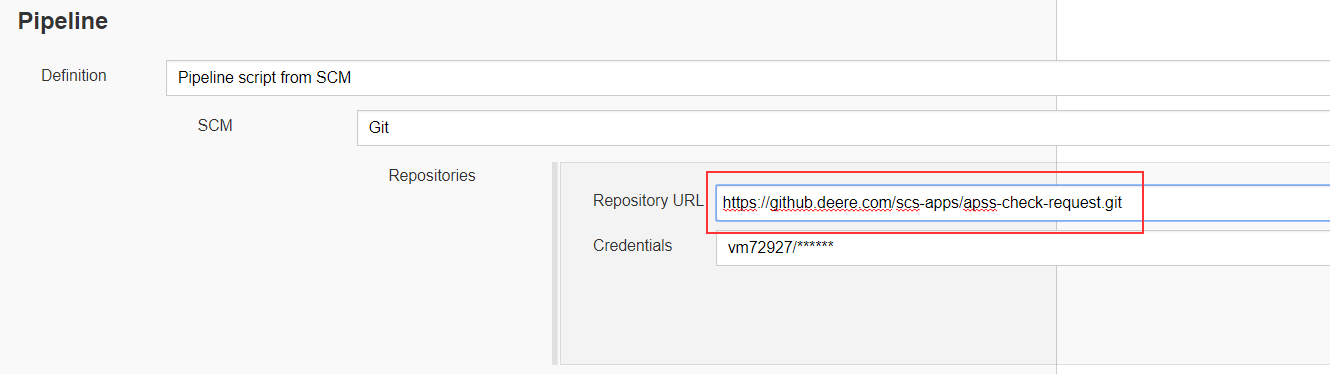
Change the “default” description for your CI pipeline.



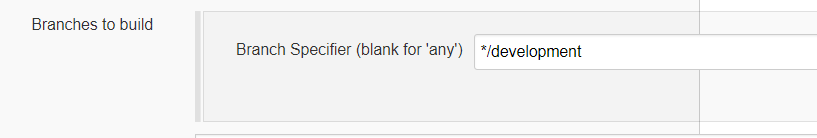
Now go to your repository and copy the url.



Now, in the **Pipeline** section. Paste this URL into the repository field.



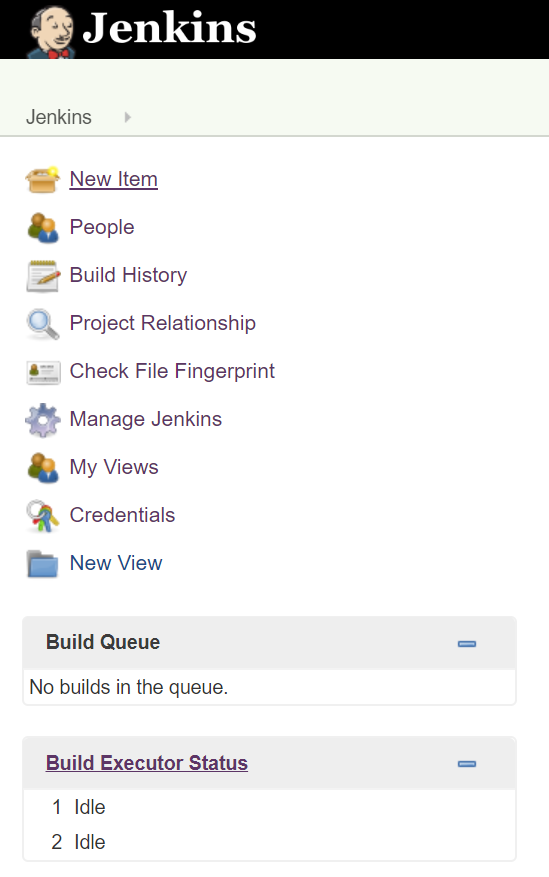
Specify the name of the branch which will be trigger the automatic compilation when a commit is pushed. In this case the name is “**development**”



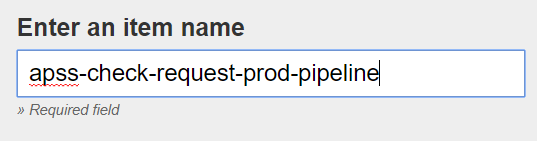
Finally press “**Save**” button.

#### Production environment

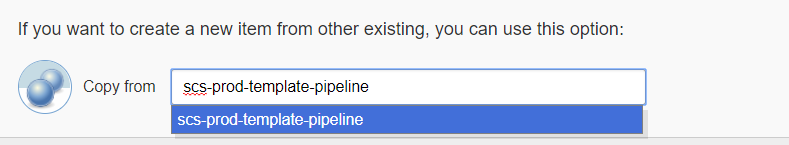
In the main menu, click on **“New Item”** option.



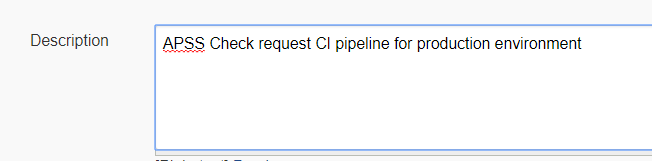
Enter the name for the pipeline



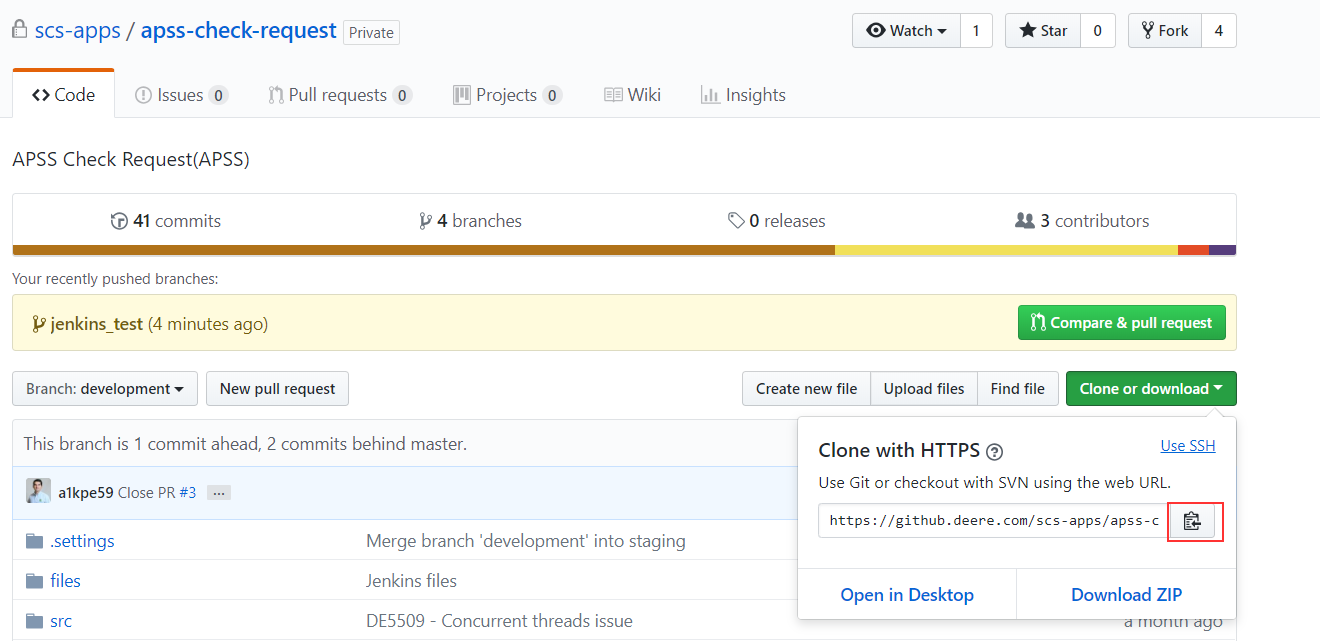
Then type in the name of the pipeline that you want to clone and press the “**OK**” button.



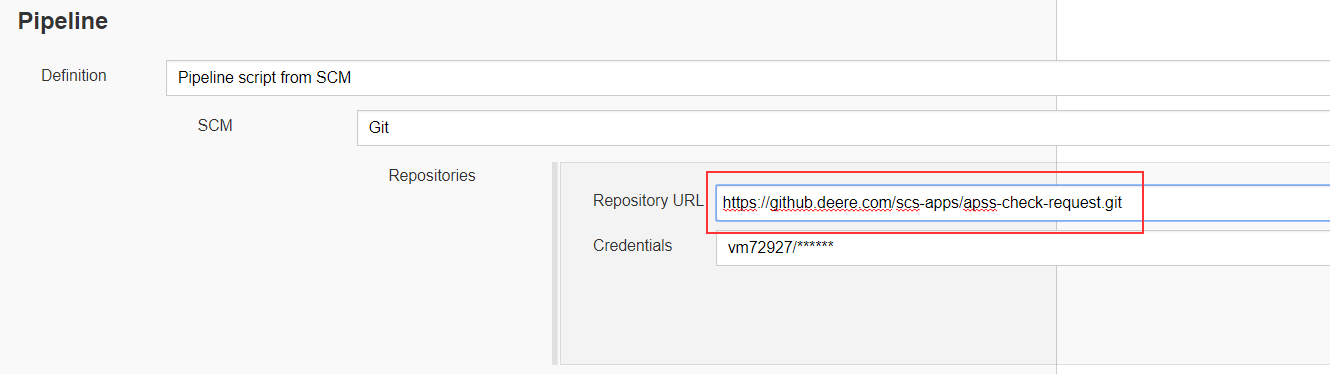
Change the “default” description for your CI pipeline.



Now go to your repository and copy the url.



Now, in the **Pipeline** section. Paste this URL into the repository field.



Specify the name of the branch which will be trigger the automatic compilation when a commit is pushed. In this case the name is “**master**”

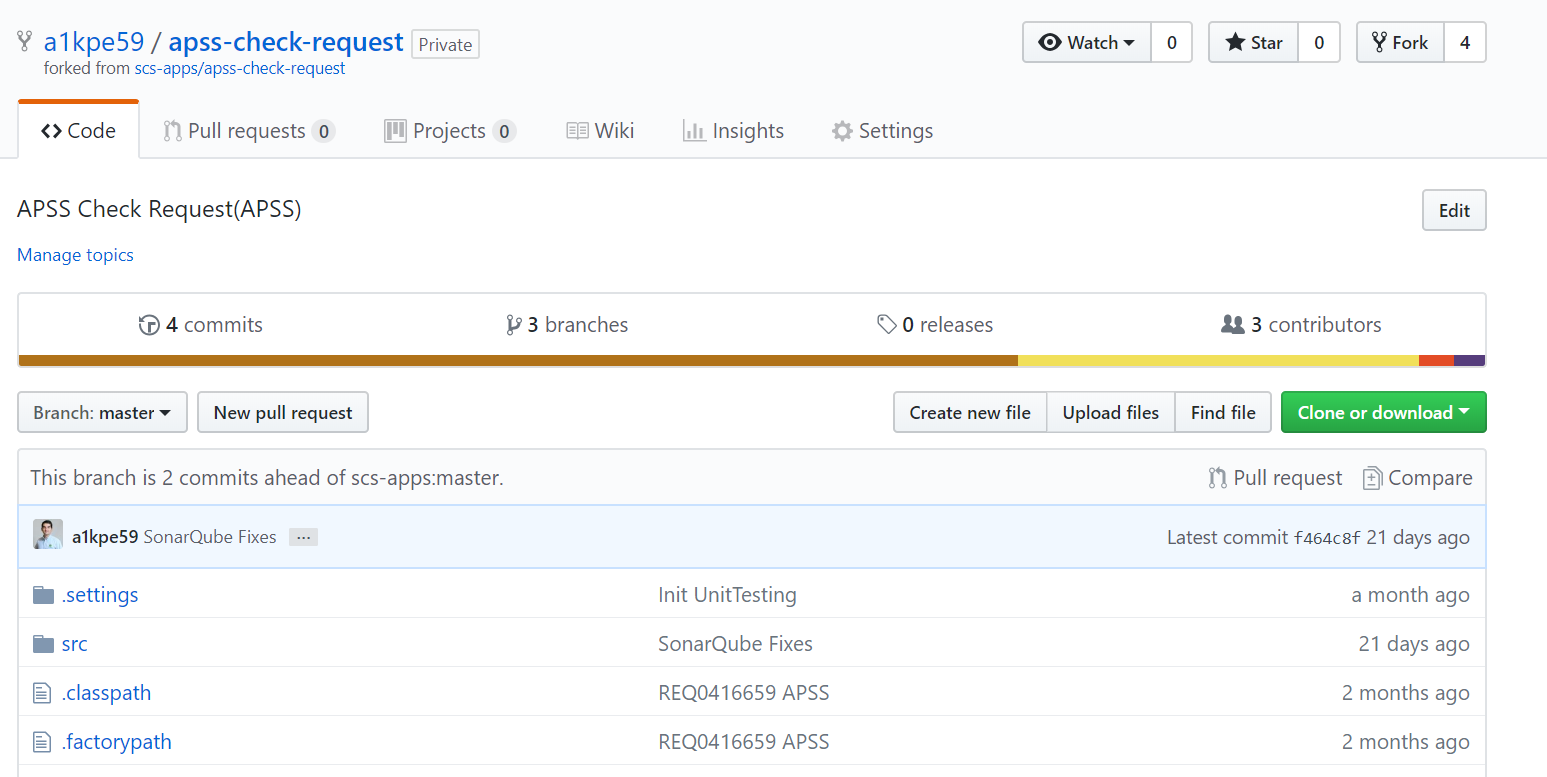


Finally press “**Save**” button.

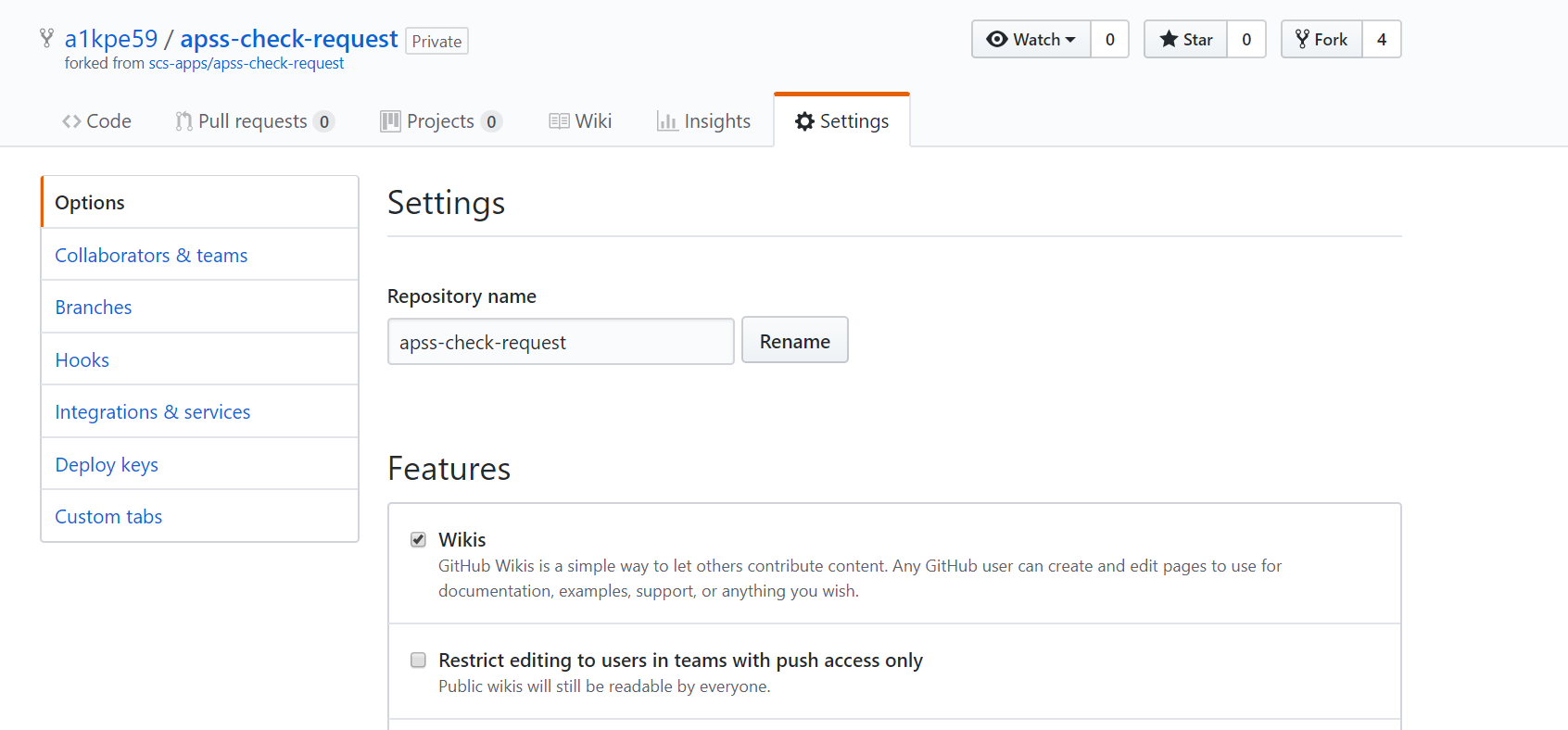
Note: Before you run the pipeline job, copy the following [folders and files](#_Structure_and_files) into your repository and [modify the following values](#_Modifying_template_Values).

## Create a webhook (github)

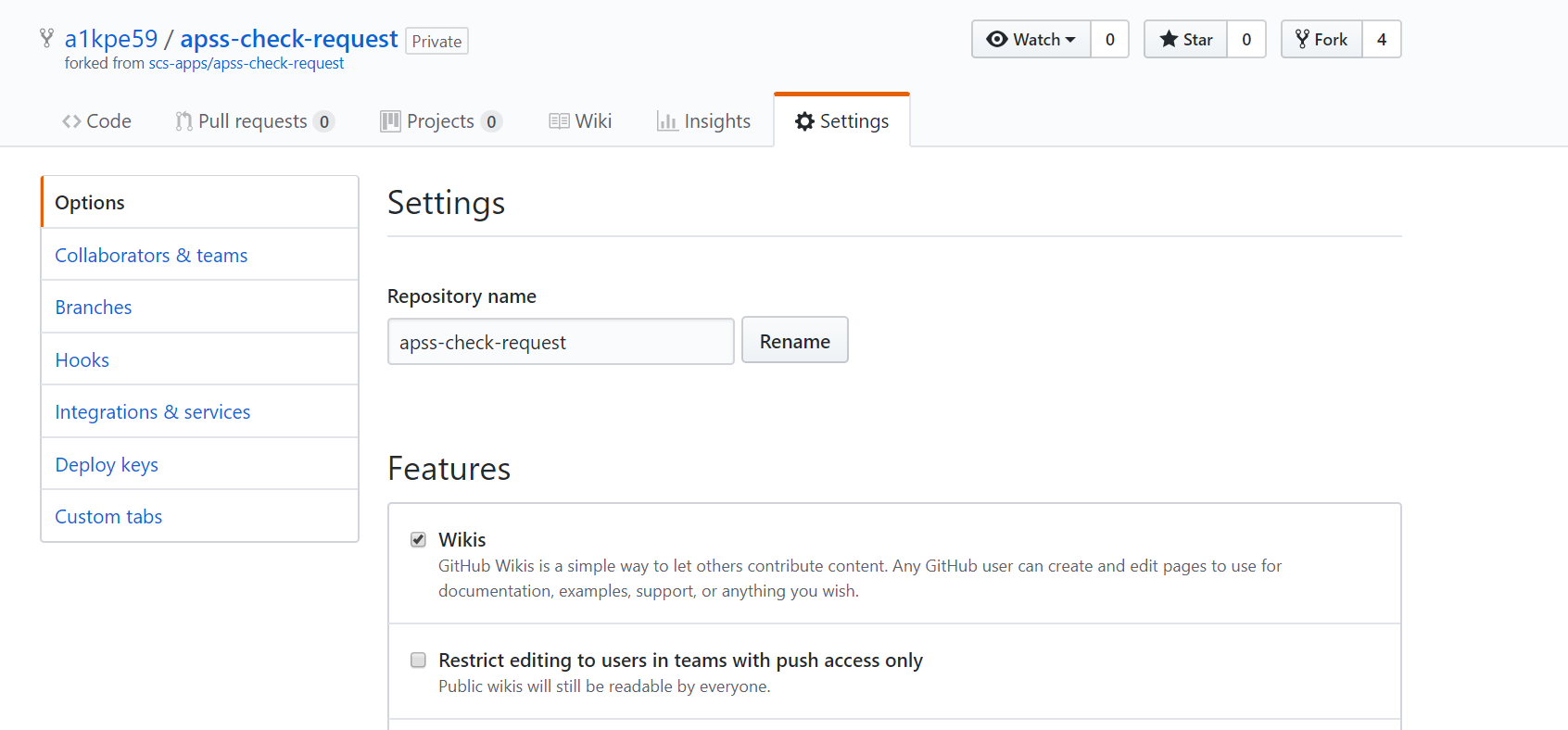
Go to your repository through <https://github.deere.com/>



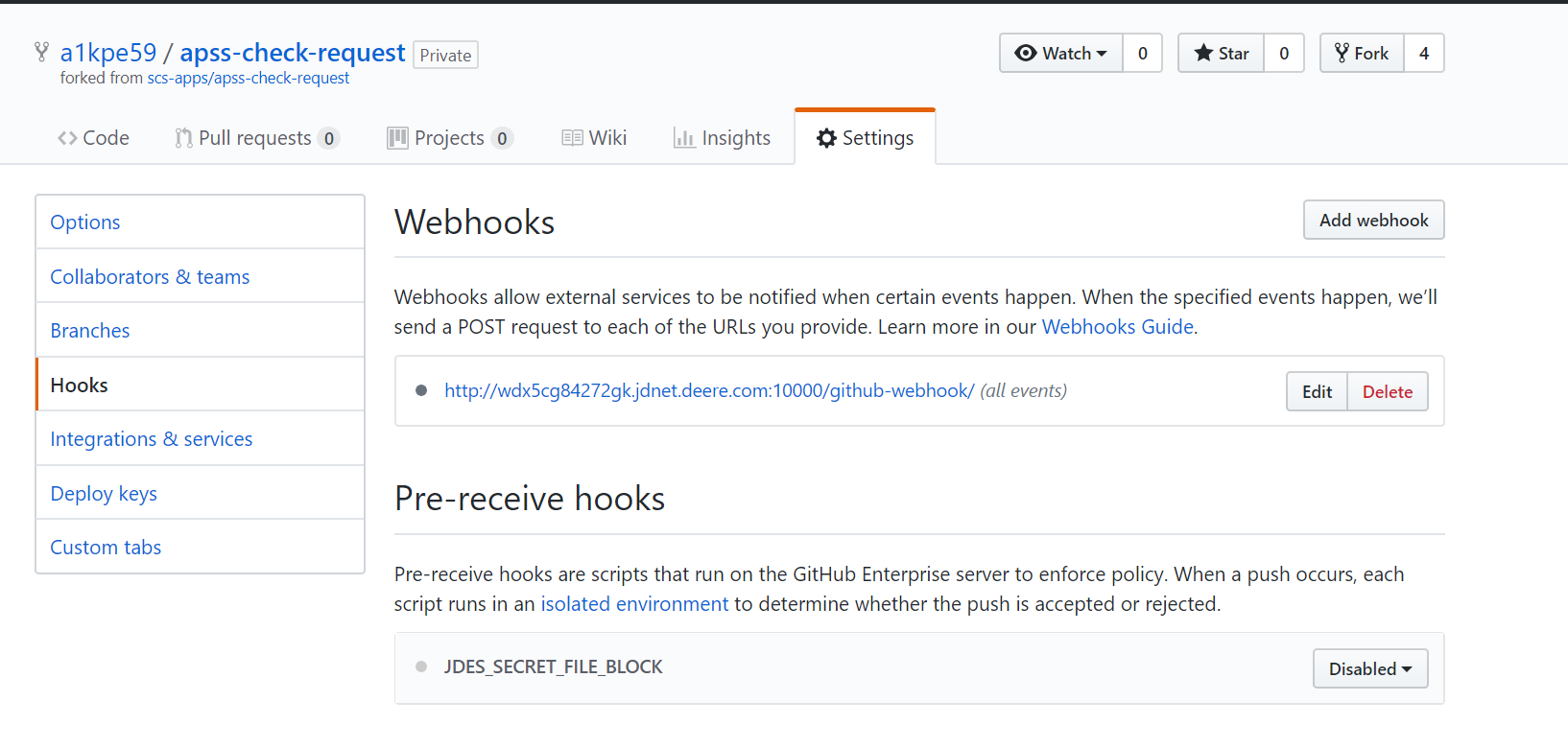
Then go to “**Settings**” tab



Now in the right side, select the “**Hooks**” option.

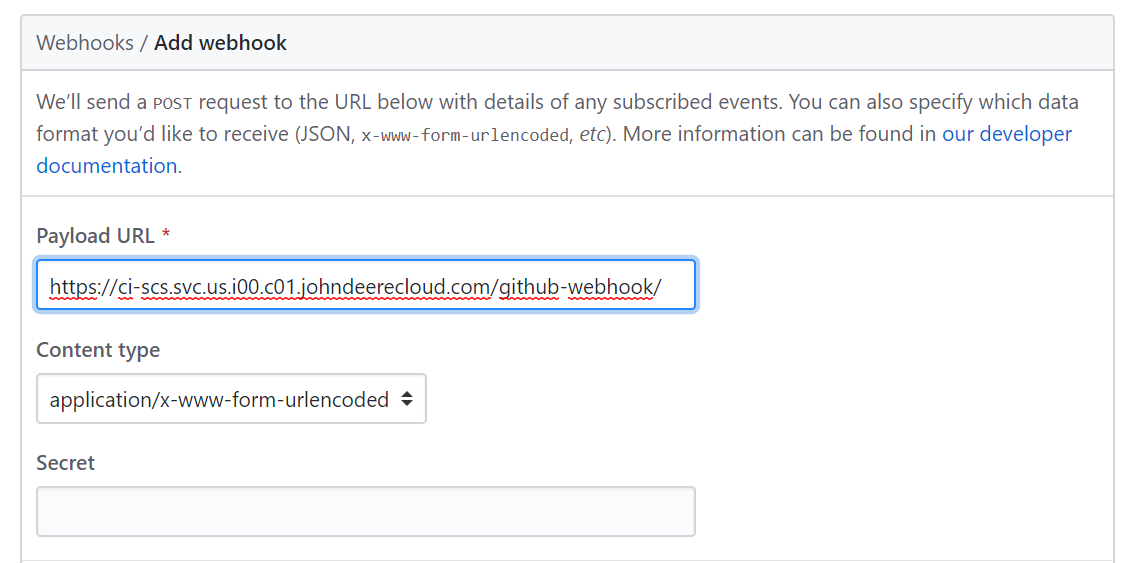


In this screen, press the “**Add webhook**” button.

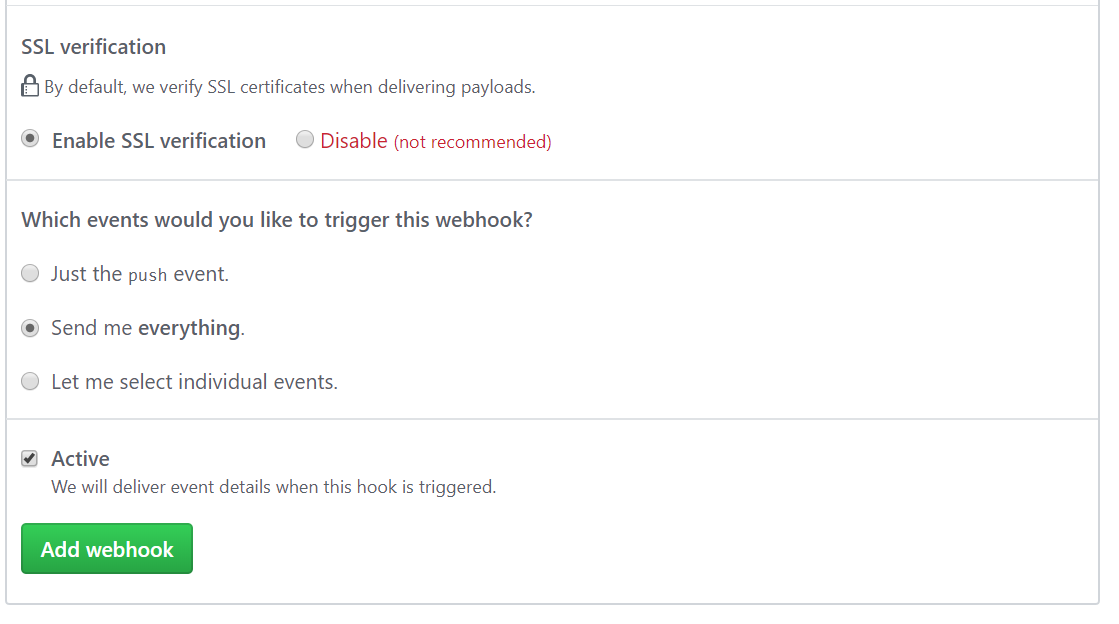


Now you have to specify the url for the webhook. It is composed with the following structure:

**[jenkinsURL]/github-webhook/**

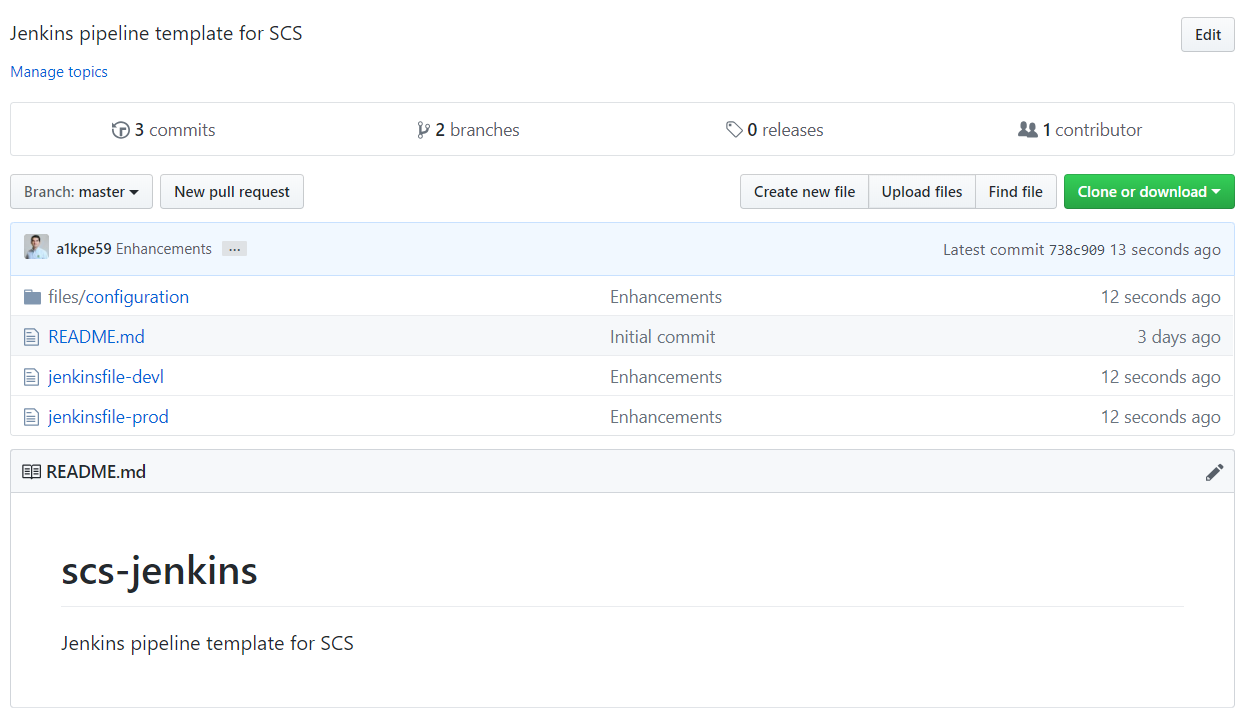


Complete the configuration of the webhook with the following values:



## Folders and files needed

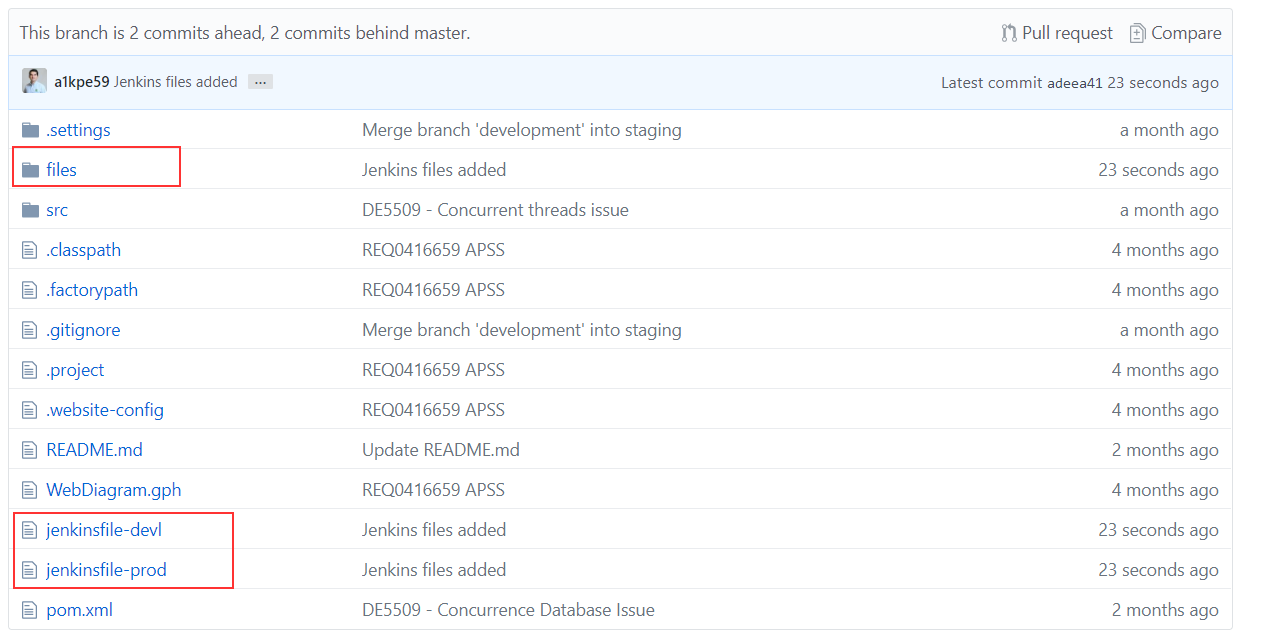
Go to the url: <https://github.deere.com/scs-apps/scs-jenkins>



Copy (or download) the following folder and files and paste them into your repository:

* files/configuration
* jenkinsfile-devl
* jenkinsfile-prod

After that, your repository may look like this:



Now let’s [modify the following values](#_Modifying_template_Values).

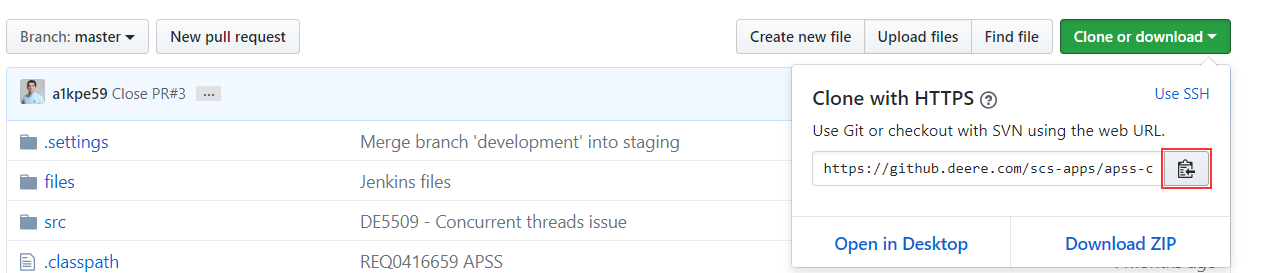
## Modifying template Values

### Update jenkins file

Open the file into a notepad. You will see a section with a comment: “**Values that must be modified**”. These values must be modified as follows.

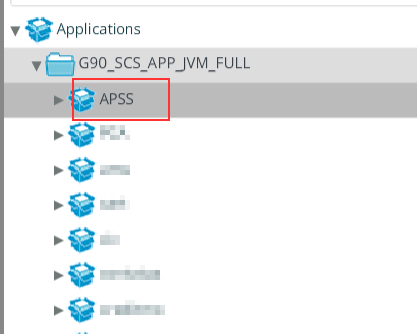
* GIT\_REPOSITORY

This value corresponds to the github repository. It can be taken from this button.



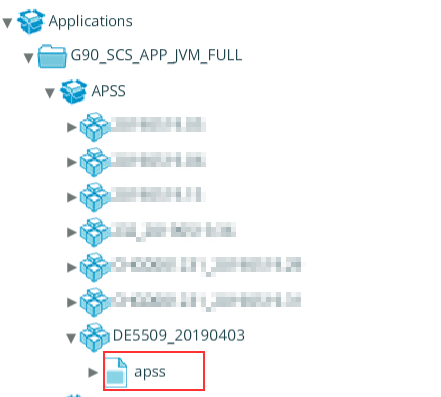
* DEPLOYIT\_APPLICATION\_NAME

This value is used in the [Deployit.deere.com](https://deployit.deere.com/) page to identify the application. The value corresponds to this.



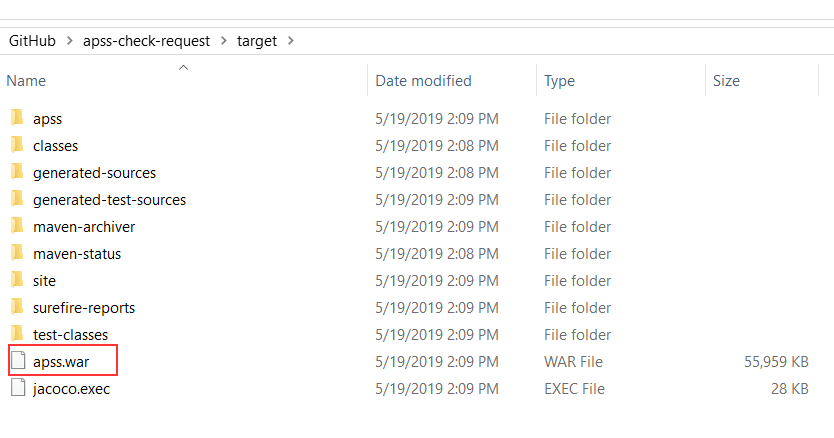
* DEPLOYIT\_ARTIFACT\_NAME

This value is used in the [Deployit.deere.com](https://deployit.deere.com/) page to identify the deployment package. The value corresponds to this.



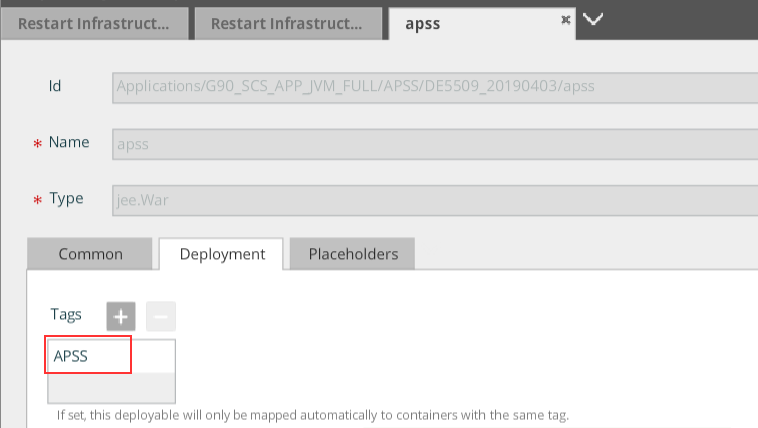
* DEPLOYIT\_ARTIFACT\_FILE

This value is the name of the war file that is created by the compilation process and is going to be uploaded to the [Deployit.deere.com](https://deployit.deere.com/) page. It could be seen in the /target folder of the project.



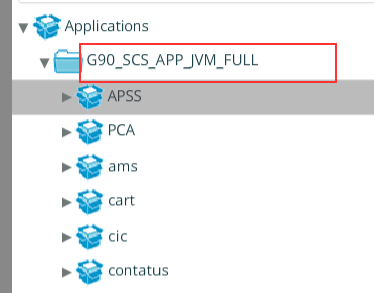
* DEPLOYIT\_ARTIFACT\_TAG

This value is used in the [Deployit.deere.com](https://deployit.deere.com/) page to deploy the application in the selected server. The value corresponds to this.



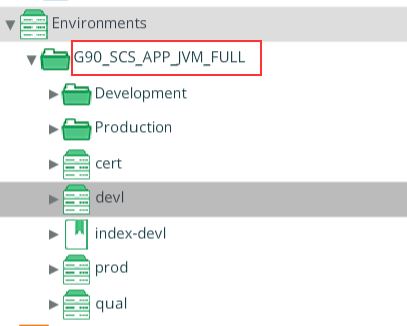
* APPLICATION\_GROUP

This is the name of the group where the application is published. The value corresponds to this.



* ENVIRONMENT\_GROUP

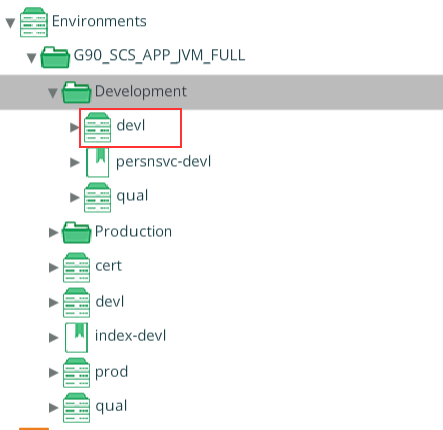
This is the name of the group where the deployment server is located. The value corresponds to this.



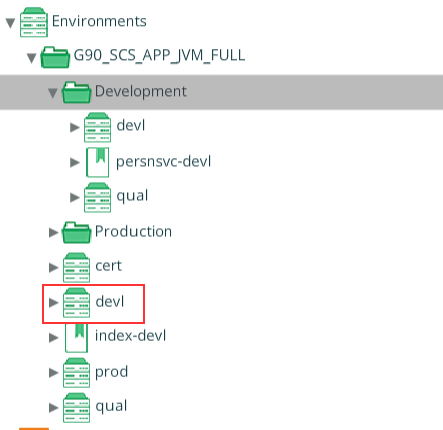
* ENVIRONMENT\_PATH

This value is the path where the deployment server is located. In this case, the parameter can take two different values.

* + In the first scenario, if the environment is the marked one. The path should be: “**Development/devl**”



* + In the second scenario, if the environment is the marked one. The path should be just: “**devl**”



* EMAIL\_TO

This value should be any valid email address. The value is used to notify about the status of each job compilation.

**Note:** All these values are case-sensitive, so take care of what you’re writing in each parameter.

#### Example of values:

After doing the updates to the Jenkins file, you should have something like this.

GIT\_REPOSITORY = 'https://github.deere.com/scs-apps/apss-check-request.git'

DEPLOYIT\_APPLICATION\_NAME="APSS"

DEPLOYIT\_ARTIFACT\_NAME="apss"

DEPLOYIT\_ARTIFACT\_FILE="apss.war"

DEPLOYIT\_ARTIFACT\_TAG="APSS"

APPLICATION\_GROUP="G90\_SCS\_APP\_JVM\_FULL"

ENVIRONMENT\_GROUP="G90\_SCS\_APP\_JVM\_FULL"

ENVIRONMENT\_PATH="devl"

EMAIL\_TO="SaldanaGustavo@johndeere.com"