# Slack jenkins alert

Now we will create a notification to slack after building the project.

In slack workspace, go to Settings & administration - manage apps

devsecops v 
# devsecops v 
devsecops-rcm2595.slack.com 
90 days left to view full history 
On the free version of Slack, messages and 
files sent more than 90 days aw will be 
hidden. See upgrade options 
Invite people to devsecops 
Create a channel 
Preferences 
Settings & administration 
Tools 
Add workspaces 
Open the desktop app 
Sign in to devsecops on mobile 
Sign out of devsecops 
a bookmark 
This is the very beginning of the devsecops 
This channel is for everything #devsecops. Hold 
2+ Add people 
Workspace settings 
Customize devsecops 
Edit workspace details 
o 
Manage members 
kil 

Search Jenkins CI and add to slack.

< Browse Apps 
Add to Slack 
Jenkins CI 
Description 
Configuration Security & Compliance 
Jenkins CI is a customizable continuous integration serverwith over 600 
to meet your needs. 
This integration will post build notifications to a channel in Slack. 

**Setup Instructions**

**Step 1**

In your Jenkins dashboard, click on **Manage Jenkins** from the left navigation.

e Jenkin 
Jenkins 
New Item 
People 
Build History 
Manage Jenkins 

**Step 2**

Click on **Manage Plugins** and search for **Slack Notification** in the **Available** tab. Click the checkbox and install the plugin.

Filter: 
Updates 
Install 
Available 
Installed 
Advanced 
Name 
Slack Notification 
This plugin is a Slack notifier that can publish build status to Slack channels. 
C). slack notificationl 
Version 
2.8 

**Step 3**

After it's installed, click on **Manage Jenkins** again in the left navigation, and then go to **Configure System**. Find the **Global Slack Notifier Settings** section and add the following values:

* **Team Subdomain:** devsecops-rcm2595
* **Integration Token Credential ID:** Create a secret text credential using BX2nTsWQSqRz4cu37UiVxzLj as the value

The other fields are optional. You can click on the question mark icons next to them for more information. Press **Save** when you're done.

**Note:** Please remember to replace the Integration Token in the screenshot below with your own.

We will create an entry in the workspace field as devsecops-rcm2595 this is the Team Subdomain provided after installing the jenkins app

Next we create a credential named slack-token with the secret text BX2nTsWQSqRz4cu37UiVxzLj which has been provided.

default channel is the channel to receive notifications on slack

Slack 
Workspace 
devsecops-rcm2595 
Credential ? 
slack-token 
+ Add 
Default channel / member id 
#devsecops 
o 
Custom slack app bot user 
Advanced v 
Apply 

Run test and we can have this message in slack



**Step 4**

For each Project that you would like receive notifications for, choose **Configure** from the project's menu.

aun6uuOO 
Ρθ!0Ι8 
se6ue•l.10 

**Step 5**

Then you'll need to add **Slack Notifications** to the **Post-build Actions** for this project.

Build 
Add build Step 
Post-build Actions 

**Step 6**

In the **Slack Notifications** section, choose the events you'd like to be notified about.

Post-build Actions 
Slack Notifications 
Notify Build Start 
Notify Aborted 
Notify Failure 
Notify Not Built 
Notify Success 
Notify Unstable 
Notify Regression 
Notify Back To Normal 
Add post-build action 
Advanced... 

Now, we will create Shared Libraries to customize the message to send to slack.

1. Go to your Jenkins dashboard and click on "Manage Jenkins".

2. Click on "Configure System".

3. Scroll down to the "Global Pipeline Libraries" section and click on the "Add" button.

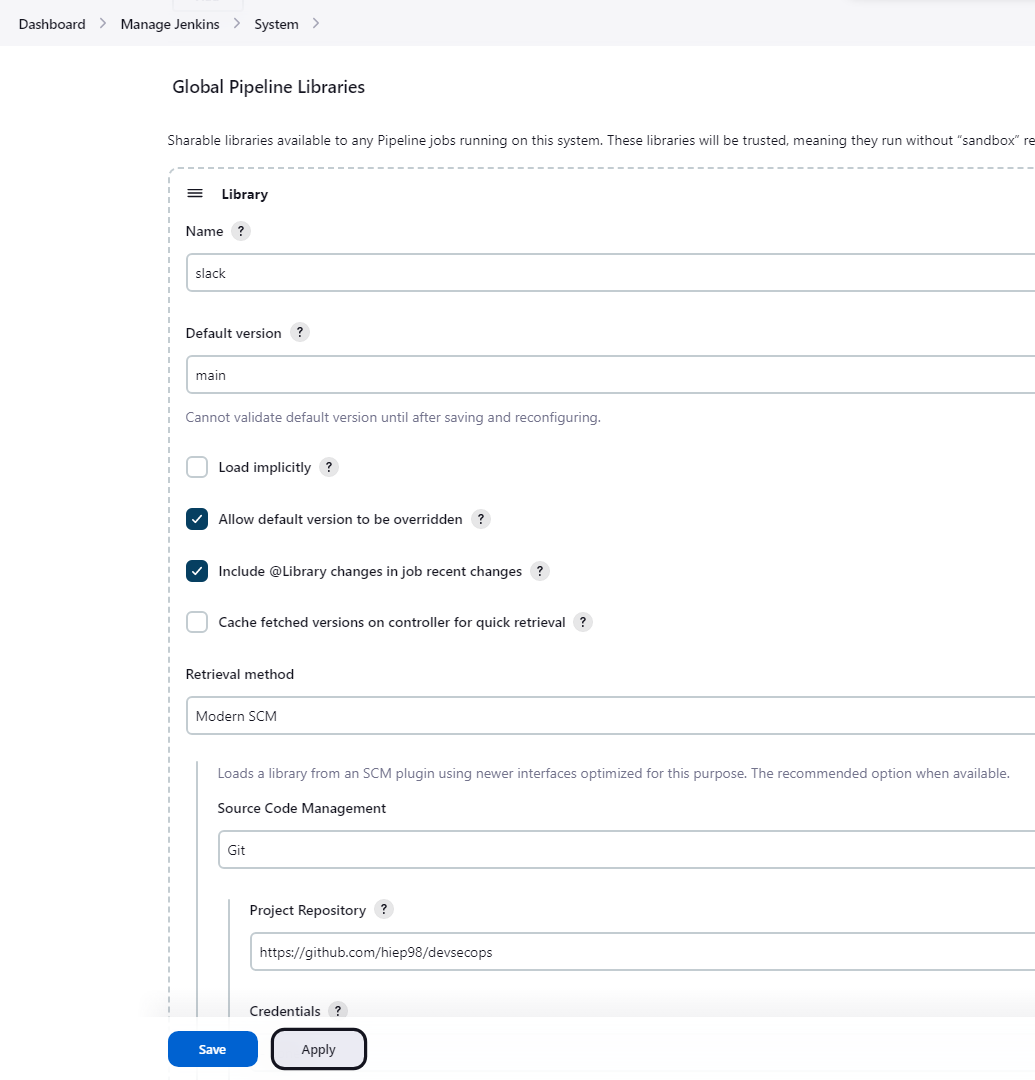
4. In the "Name" field, enter a name for your library (for example, "slack-library").

5. In the "Default Version" field, enter the default version of your library (for example, "main").

6. In the "Modern SCM" section, select the source code management system you are using for your library (for example, "Git").

7. Enter the repository URL for your library in the "Project Repository" field (for example, " <https://github.com/myusername/my-slack-library.git>").

8. Click on "Save" to save your changes.



Now, you can use your new global pipeline library in your Jenkins pipelines to send custom messages to Slack. You need add @Library('slack') \_ on the top of jenkins pipeline and here is script.

|  |
| --- |
| @Library('slack') \_  always {  sendNotification currentBuild.result  }  } |

This is a Jenkinsfile that uses a shared library called ‘slack’ and sends a notification with the current build result.

The ‘@Library’ annotation is used to specify the name of the shared library to use. The underscore character (‘\_’) specifies the default version of the library to use.

The ‘always’ block is a pipeline step that will always be executed regardless of whether the previous steps have failed or succeeded. It sends a notification with the current build result using the ‘sendNotification’ function.

The 'currentBuild.result' variable is used to access the result of the current build in Jenkins. It can be used to determine if there were any test failures or if the build was successful.

For example, in your Jenkinsfile, you are using the ‘sendNotification’ function to send a notification with the current build result. The ‘currentBuild.result’ variable is used to get the result of the current build and pass it as an argument to the ‘sendNotification’ function.

Now we create file sendNotification.groovy in vars folder to custom slack notification.

|  |
| --- |
| def call(String buildStatus = 'STARTED') {  buildStatus = buildStatus ?: 'SUCCESS'  def color  if (buildStatus == 'SUCCESS') {  color = '#47ec05'  } else if (buildStatus == 'UNSTABLE') {  color = '#d5ee0d'  } else {  color = '#ec2805'  }  def msg = "${buildStatus}: `${env.JOB\_NAME}` #${env.BUILD\_NUMBER}:\n${env.BUILD\_URL}"  slackSend(color: color, message: msg)  } |

This is a Groovy script file that includes a function called "call". The function takes a string argument called "buildstatus" and if no value is specified, it defaults to "started".

The function then checks the value of "buildstatus" and assigns a color based on its value. If "buildstatus" is equal to "success", then the color "#47ec05" is assigned. If "buildstatus" is equal to "unstable", then the color "#d5ee0d" is assigned. Otherwise, the color "#ec2805" is assigned.

The function then creates a message string that contains the value of `buildStatus`, the name of the job (`env.JOB\_NAME`), the build number (`env.BUILD\_NUMBER`) and the build URL (`env.BUILD\_URL`) using Groovy's string interpolation feature¹. Finally, it calls another function named `slackSend` with two parameters: `color` and `message`. The function sends a notification message to Slack with the specified color and message.

You can see result below



Reference: [Extending with Shared Libraries (jenkins.io)](https://www.jenkins.io/doc/book/pipeline/shared-libraries/)

<https://e.printstacktrace.blog/jenkins-pipeline-environment-variables-the-definitive-guide/>

<https://stackoverflow.com/questions/39140191/how-to-send-slack-notification-after-a-jenkins-pipeline-build-failed>

Next, we go to with manual approve the Deployment to Production

|  |
| --- |
| stage('Prompte to PROD?') {  steps {  timeout(time: 2, unit: 'DAYS') {  input 'Do you want to Approve the Deployment to Production Environment/Namespace?'  }  }  } |

This stage is a timeout step that waits for 2 days for a user input. The user input is asking for approval to deploy the code to the production environment or namespace. If the user approves the deployment within the timeout period, the pipeline continues to the next stage, which is likely a stage that triggers the actual deployment to production. If the user does not approve the deployment within the timeout period, the pipeline will fail because the deployment was not approved. This stage adds an important layer of safety to the pipeline to ensure that code changes are not accidentally or haphazardly deployed to production without proper review and approval.

Now run build and you can you notification like below

I I < 48 
Branch: — 
Commit: 
Start Build Artifact 
Prompte to PROD? - 4m 35s 
@ 12m 44s 
Unit Tests 
Mutation Tests - 
Changes by nthiep1998 
Started by GitHub push by hiep98 
Docker Build and 
SonarQube - 
SAST 
Pipeline 
Integration Tests - 
DEV 
Changes 
Tests 
OWASP ZAP - 
DAST 
Logout 
Vulnerability Scan 
Dependency Scan 
OPA Conftest 
Tri" scan 
Push 
Vulnerability Scan 
- Kubernetes 
Kubesec Scan 
OPA scan 
Tri" scan 
K8S Deployment 
- DEV 
Deployment 
Rollout Status 
Prompte to 
PROD? 
End 
x 
4m 35s 
v Do you want to Approve the Deployment to Production Environment/Namespace? — Wait for interactive input 
Do you want to Approve the 
Deployment to Production 
Environment/Namespace? 
Proceed 
Abort 

You can select proceed and have slack notification like below

