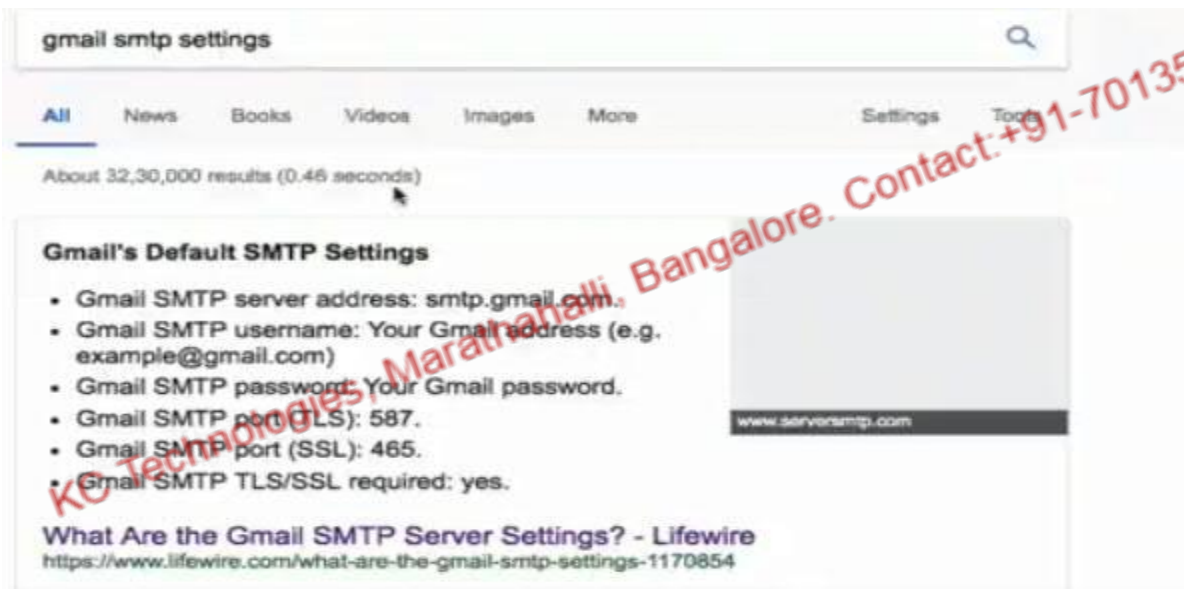


Sending Email Notification from Jenkins using Pipeline script

To do this first we need SMTP server details. For our practice we are going to use gmail server SMTP details. In real-time, our company will provide these details.

To get gmail SMTP details, search in google for "gmail smtp settings"



We need to configure these details in Jenkins. For this, login to **Jenkins**, then go to **Manage Jenkins**, then select **Configure System**. Then find the **E-mail notification** section and enter SMTP configurations like below.

Click on Advanced to get full details.

SMTP server: smtp.gmail.com

Select **Use SMTP Authentication**

User name: give your email id from which you want to send emails.

Password: give the password of the email id.

Use SSL: enable it

SMTP Port: 465 if SSL is enabled. If it is TLS, then it is 587. We use 465

Reply-To Address: If you want reply email, then enter the email to which you want reply. For now, ignore it.

Charset: UTF-8

To test our configurations, enable the "Test configuration by sending test email", then enter the email id to which you want to get test email and click on "Test Configuration". Then open your email inbox and verify whether you have received the email or not.

We are done with the SMTP configuration, so save the configurations.

Then go to Jenkins home and create a Job with below details.

```
node{
    stage('SCM Checkout'){
        git 'https://github.com/KCTechnologiesDevOps/DockerHubJenkinsIntegration'
    }
    stage('Maven Packaging'){
        def mavenHome = tool name: 'maven-3', type: 'maven'
        sh "${ mavenHome }/bin/mvn package"
    }
}
```

We created script to checkout the code from GIT and generated artifact using maven. But we need to add the script for sending Email notification as well.

For this we need to create one more stage.

```
stage('Sending Email Notification'){
}
```

We can take help of Jenkins Pipeline script generator help to get the email notification script.

Go to Pipeline script generator, select the Step as **mail: Mail** and give the details.

To: kctechnologiesdevops@gmail.com

CC: Leave it as empty

BCC: Leave it as empty

Subject: Jenkins Email notification using pipeline script.

Body: Put some static body here. but we can use dynamic content here by reading Environmental variables.

Then click on **Generate Pipeline Script** option. Copy the script and paste it inside our stage. So our script looks like below.

```
node{
    stage('SCM Checkout'){
```

```

git 'https://github.com/KCTechnologiesDevOps/DockerHubJenkinsIntegration'
}

stage('Maven Packaging'){
    def mavenHome = tool name: 'maven-3', type: 'maven'
    sh "${ mavenHome }/bin/mvn package"
}

stage('Sending Email Notification'){
    mail bcc: "", body: "'Hi Welcome to KC Technologies to learn Pipeline script to send Email Notifications
    Thanks

    KC Technologies'", cc: "", from: 'kctechnologiesdevops@gmail.com', replyTo: "", subject: 'Jenkins
    Email notification using pipeline script', to: 'kctechnologiesdevops@gmail.com'
}
}

```

We can write this file directly in Jenkins Job or else we can place in our project. Reading the script file from Jenkins and writing inside Jenkins are different options.

In the Jenkins Job under Pipeline section, if we select Pipeline script from SCM, then we can read the Jenkinsfile from Git repo.

Integrating Jenkins with Slack using Pipeline script:

To achieve this, we need to install Slack Notification plugin in Jenkins and we need to use this as part of Jenkins pipeline Job to send Notification.

First let's install Slack Notification Plugin

Login to Jenkins and go to **Manage Jenkins**, then go to **Manage Plugins**, select **Available** and search for **Slack Notification**.

Select the **Slack Notification** plugin and click on install without restart.

Go to Jenkins Pipeline Script generator option and there you can see a Step called: **slackSend: Send Slack Message**. Select that option and we need to enter few details like below:

Message: We can put some static message here for now. If required, we can create dynamic message using Environmental variables.

Color: Take help by clicking on I symbol and specify some color. For now give it as **good**. Then click on **Advanced**.

Channel: Here we need to give the slack channel name, to which we have to receive the notifications.

Ex: **Channel: #demo**

To get the channel details, go to your slack account, create a channel, click on **Add an app** option.

demo

You created this channel today. This is the very beginning of the # demo channel.

[Set a purpose](#) [+ Add an app](#) [Invite others to this channel](#)

Today

 **Krishna Chaitanya KC** 10:29 PM
joined #demo.

Then search for webhooks and select **Incoming WebHooks** option.

Staff Picks

Featured
Essential Apps
New & Noteworthy
Brilliant Bots

App Collections

webhooks



Incoming WebHooks
Send data into Slack in real-time.



Outgoing WebHooks
Get data out of Slack in real-time.



Cosmic JS
Connects your Cosmic JS Webhooks to the Slack channel of your choice.

Then click on **Add Configuration** if you are doing this for the first time and then scroll down and copy the webhooks url.

Incoming WebHooks

App Info **Settings**

Add Configuration

App Homepage

App help

This app was made by Slack.
This integration was made by a member of the Slack team to help connect Slack with a third party service; these Slack integrations may not be tested, documented, or supported by Slack in the way we support our core offerings, like Slack Enterprise Grid and Slack for Teams. You may provide feedback about these apps at feedback@slack.com.
It only uses data Slack already has access to (view our [Privacy Policy](#) to learn more). By enabling and/or using this integration, you may be connecting with a service that is not part of Slack.

If you are using it for more than once, then click on the edit option

Incoming WebHooks

App Info **Settings**

Add Configuration

App Homepage


App help

Terms

Categories:

[Report this app](#) to Slack for

Configurations

 Posts to #demo as incoming-webhook
Krishna Chaitanya KC on Jan 1, 2019

After clicking the Edit option, scroll down and select the Webhooks url.

Webhook URL

Send your JSON payloads to this URL.

[Show setup instructions](#)

<https://hooks.slack.com/services/TBVTGCGJE/BF3SG56BT/6Bkc5zGoWt9edNp!>

[Copy URL](#) • [Regenerate](#)

The webhooks has an authentication token. Using this token anyone can send notifications to anybody. So, keep it secret.

Copy the webhook url and go back to Jenkins.

Integration Token Credentials: Here we need to add the credentials by clicking on **Add** option. Then select the **Kind** as **Secret text**.

Here enter, the integration token from the copied webhooks, the integration token starts after **/services/**

Then give ID and description and click on Add.

Then select the added credentials.

Integration Token: If we don't want to use the integration as secret, then we can pass the token here directly. But is not a recommended way.

Team Domain: Here enter the team domain name: Ex: **devops-realtime**

Base URL: Here give the base url from the copied webhooks url, it starts from **https** and ends with **services/**

We have done with entering the details. Then click on **Generate Pipeline Script.**

Copy the generated script and update our previous example Jenkinsfile with this script by creating new Stage at the last.

Then commit these changes to GIT and trigger the build.

In the console, you can observe Jenkins sending Email notification along with Slack Notification.

You can verify this by checking you email inbox and Slack channel inbox.

Finally the generated script looks like below:

```
node{
  stage('SCM Checkout'){
    git 'https://github.com/KCTechnologiesDevOps/DockerHubJenkinsIntegration'
  }
  stage('Maven Packaging'){
    def mavenHome = tool name: 'maven-3', type: 'maven'
    sh "${ mavenHome }/bin/mvn package"
  }
  stage('Sending Email Notification'){
    mail bcc: "", body: "'Hi Welcome to KC Technologies to learn Pipeline script to send Email Notifications
    Thanks
    KC Technologies'", cc: "", from: 'kctechologiesdevops@gmail.com', replyTo: "", subject: 'Jenkins
    Email notification using pipeline script', to: 'kctechologiesdevops@gmail.com'
  }
  stage('Sending Slack Notification'){
    slackSend baseUrl: 'https://hooks.slack.com/services/',
    channel: '#demo',
    color: 'good',
```

```
    message: 'Sending slack Notification',  
    teamDomain: 'devops-realtime',  
    tokenCredentialId: 'integration-token'  
  }  
}
```

KC Technologies, Marathahalli, Bangalore. Contact: +91-7013558586