Abstract

The following document demonstrates how to trigger build in Jenkins when anyone commits a change in the GitHub repository.

The document also demonstrates how to configure Jenkins to send an email to the Configured Email ID when any build fails along with the Build Log file.

Contents

Prerequisites

- **Jenkins Up** and **Running**.
- **GitHub** account
- **GitHub Repository** with a **Maven Project**
- **Git Plugin installed** on **Jenkins**

Configuring GitHub

Step 1:

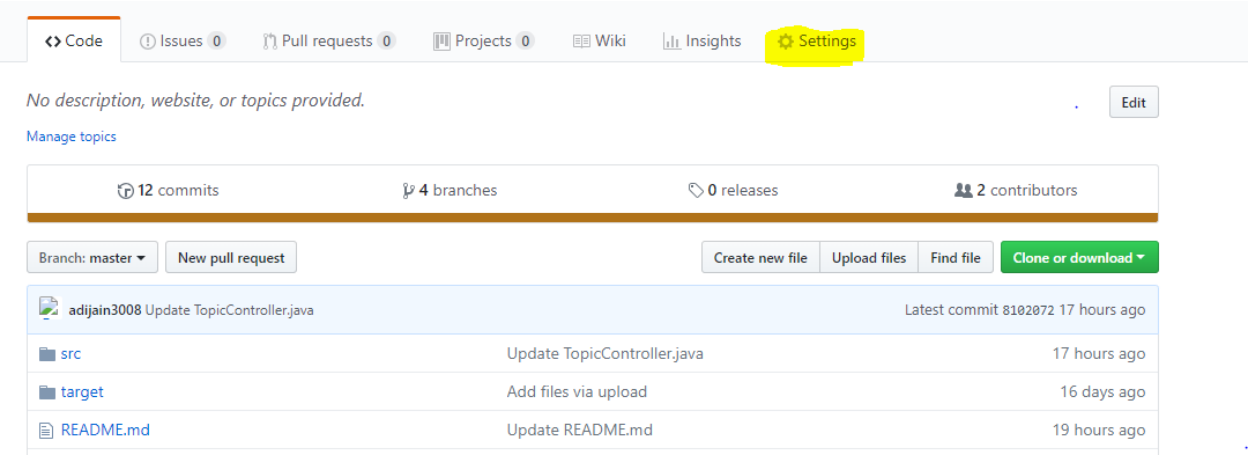
Open https://github.com/ and login with your ID and Password

Step 2:

Open the Repository which contains the Maven Project

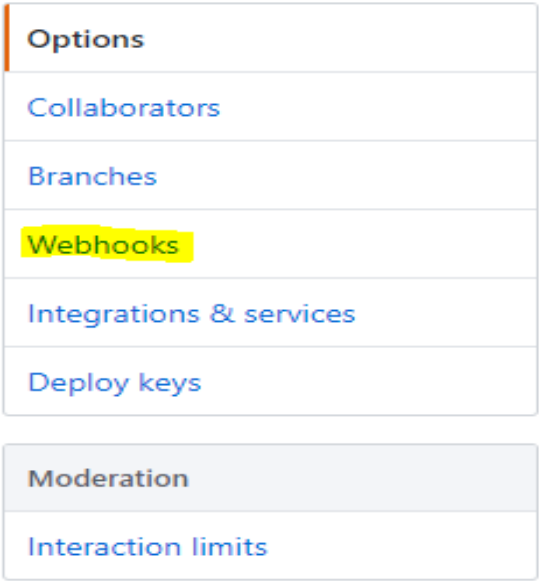
Step 3:

Click on the Settings Button



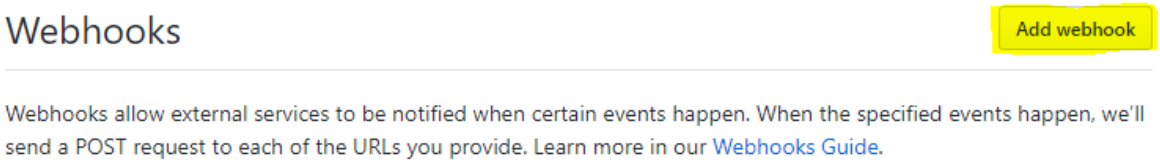
Step 4:

Click on the Webhooks Button



Step 5:

Click on Add webhook Button



Step 6:

Go to Jenkins → Manage Jenkins → Configure System and scroll down to GitHub section. Click on this button

Step 7:

Copy this URL

By default

This plugin doesn't do anything with the GitHub API unless you add a configuration with credentials. So if you don't want to add any configuration, you can set up hooks for this Jenkins instance manually.
In this mode, in addition to configuring projects with "*Build when a change is pushed to GitHub*", you need to ensure that Jenkins gets a POST to its `http://yuhur54667dns.westus.cloudapp.azure.com:8080/github-webhook/`.

Step 8:

Paste it in Payload URL section and Click on Add Webhook Button

Payload URL *

http://yuhur54667dns.westus.cloudapp.azure.com:8080/github-webh

Content type

application/x-www-form-urlencoded

Secret

Which events would you like to trigger this webhook?

- Just the `push` event.
- Send me **everything**.
- Let me select individual events.

☑ **Active**
We will deliver event details when this hook is triggered.

Add webhook

Step 9:

Your Webhook has been added

Webhooks | Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://yuhur54667dns.westus.cloudapp.azure.com:8080/github-webhook/ *(push)* | Edit | Delete

Configuring Jenkins

Step 1:

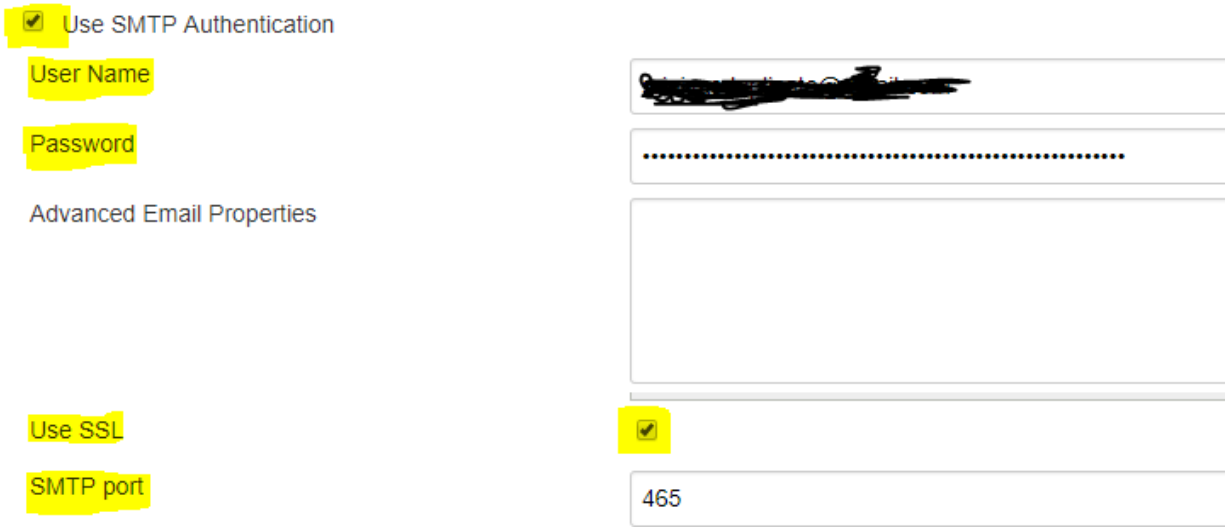
Go to Jenkins → Manage Jenkins → Configure System → Extended Email Notification Section

Step 2:

Write SMTP Server address of the sender's Email ID in the SMTP Server Textbox (In this case Gmail). Then click on Advanced Button

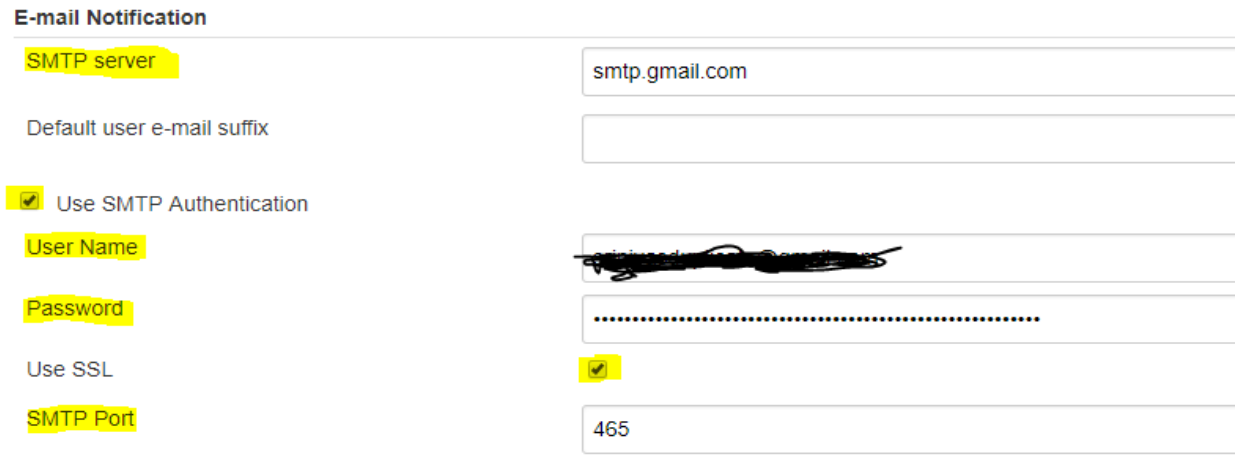
Step 3:

Select the Use SMTP Authentication and Use SSL CheckBox. Add Sender's Email ID and Password. Write SMTP Port as 465 (for Gmail)



Step 4:

Scroll down to E-Mail Notification Section and follow the above 2 Steps again



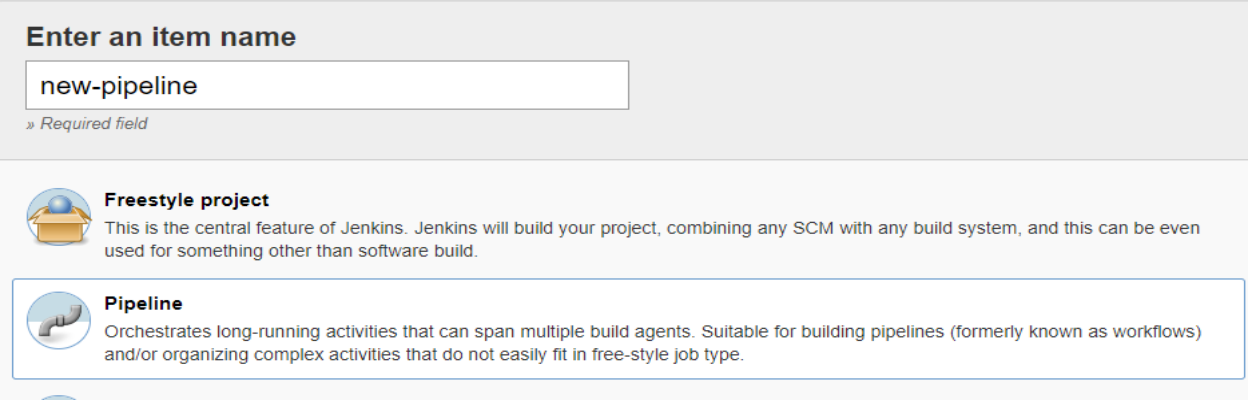
Step 5:

Click on Save Button

Creating a New Project

Step 1:

Click on New Item. Give a name for the project and Select Pipeline Project and Click OK

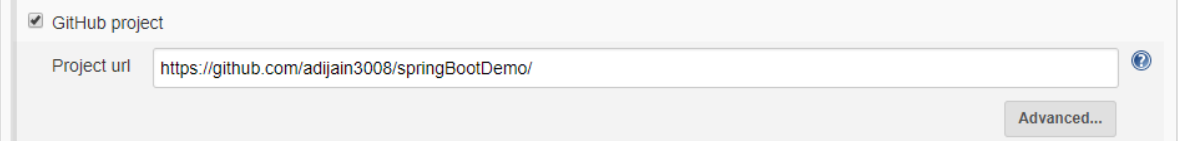


Step 2:

Click on your Project Name and then Click on Configure

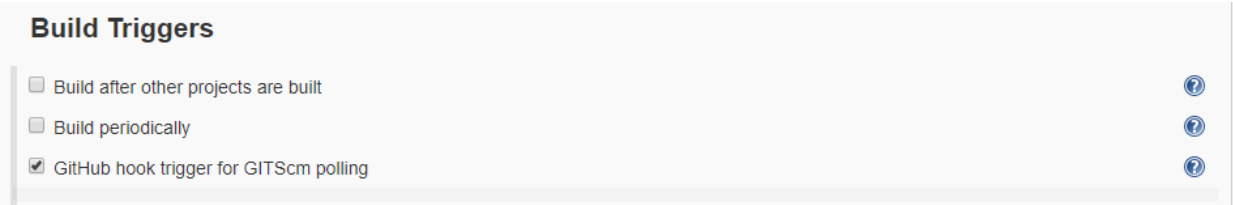
Step 3:

Provide the Git Repository Pages URL in GitHub Project TextBox



Step 4:

Select the GitHub hook trigger for GITScm polling CheckBox in Build Triggers Section



Step 5:

Enter the following Script in the Pipeline Section

```
node {
  def mvn_version = 'M3'

  try{
     stage('Preparation') {
        git 'https://github.com/adijain3008/springBootDemo.git'
        withEnv( ["PATH+MAVEN=${tool mvn_version}/bin"] ) {
           sh "mvn clean install"
        }
     }
  }catch(err){
     emailext attachLog: true, body: 'Build failed', subject: 'Build Failure', to: 'receiver's-email-address'
           currentBuild.result = 'FAILURE'
  }
}
```

Step 6:

Click on Save to Save the Changes

Step 7:

Create any error in your GitHub project and Commit the Changes

Step 8:

A build should be triggered in Jenkins as soon as the commit is saved

Step 9:

Check your supplied mail address for a mail with the error log and supplied message

Build failed