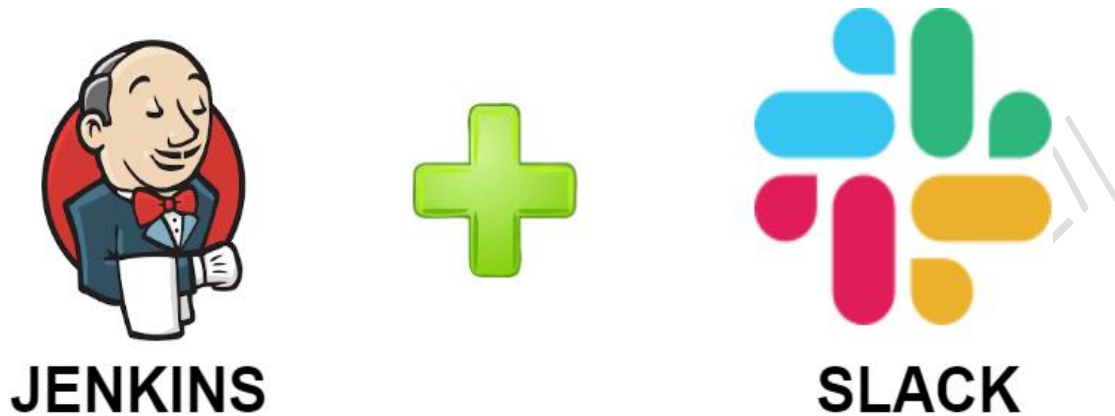# JENKINS PIPELINE TO NOTIFY SLACK USING JENKINSFILE


Prerequisites

1. AWS account
2. Jenkins CI server running on EC2
3. Slack channel

Today we will learn how to send success, error notification to slack through the Jenkins job and from the Jenkinsfile.

When developers commit changes to GitHub, according to devops best practices they should be able to receive feedback about the build process automatically so they can work on the errors and then recommit the new changes. This helps to increase development speed and agility.

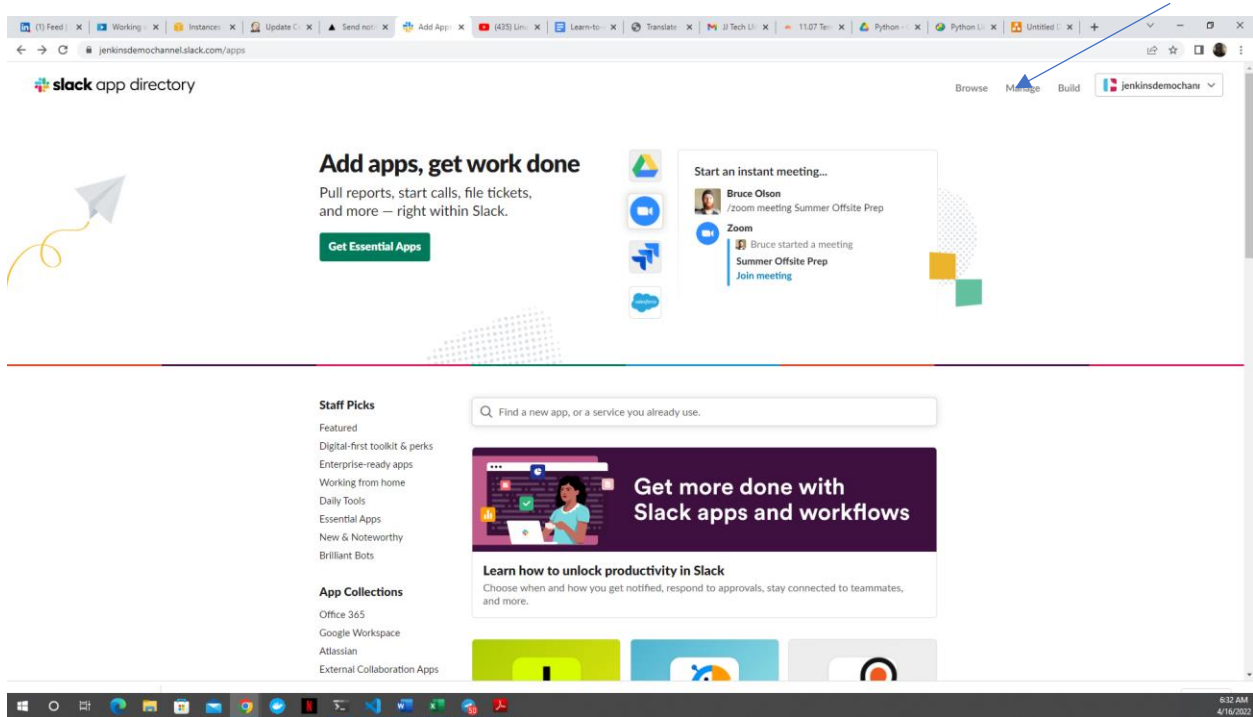# 1. Installing the slack notification plugin
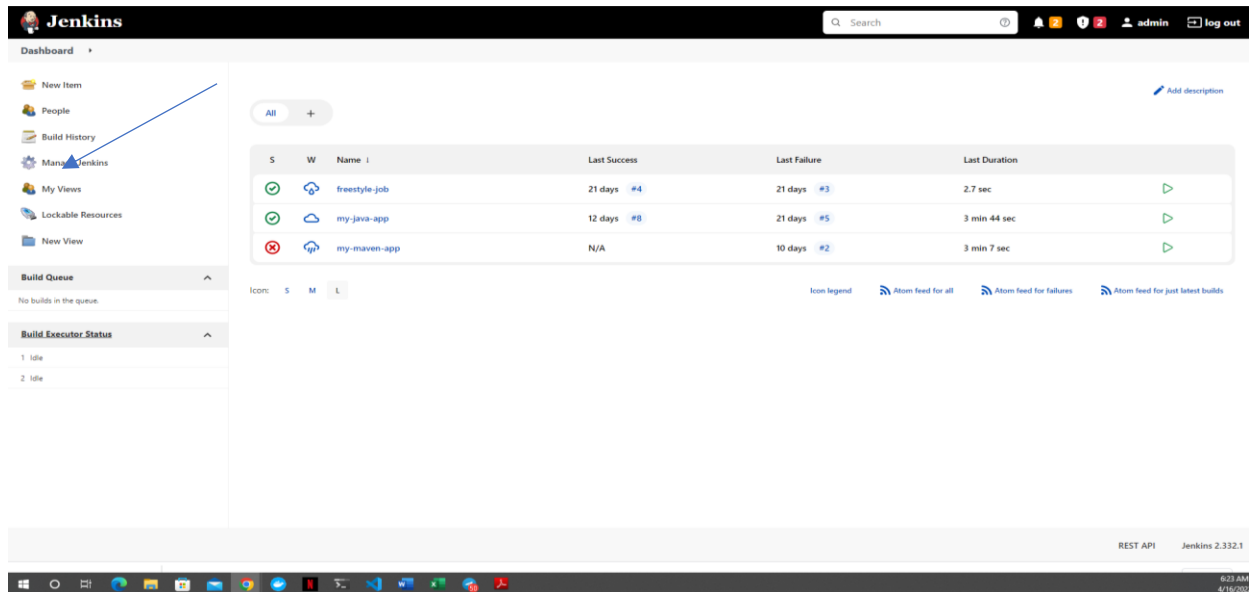
- Go to manage jenkins
- Navigate to manage plugins
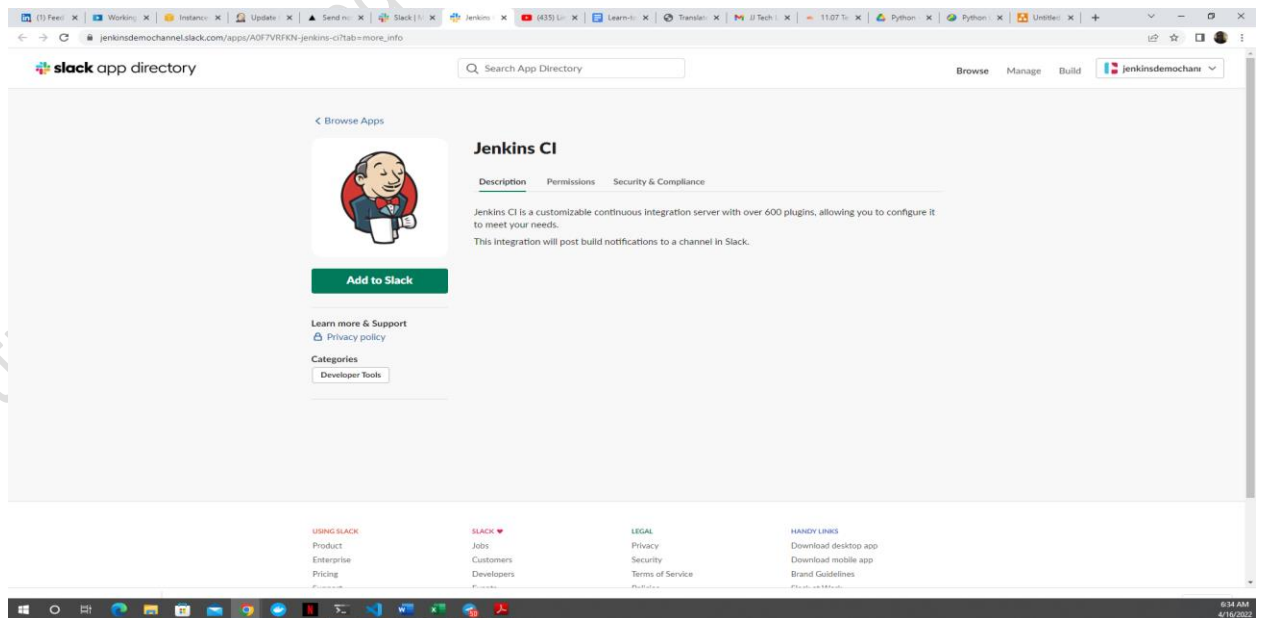- Click on available
- Search for slack notification
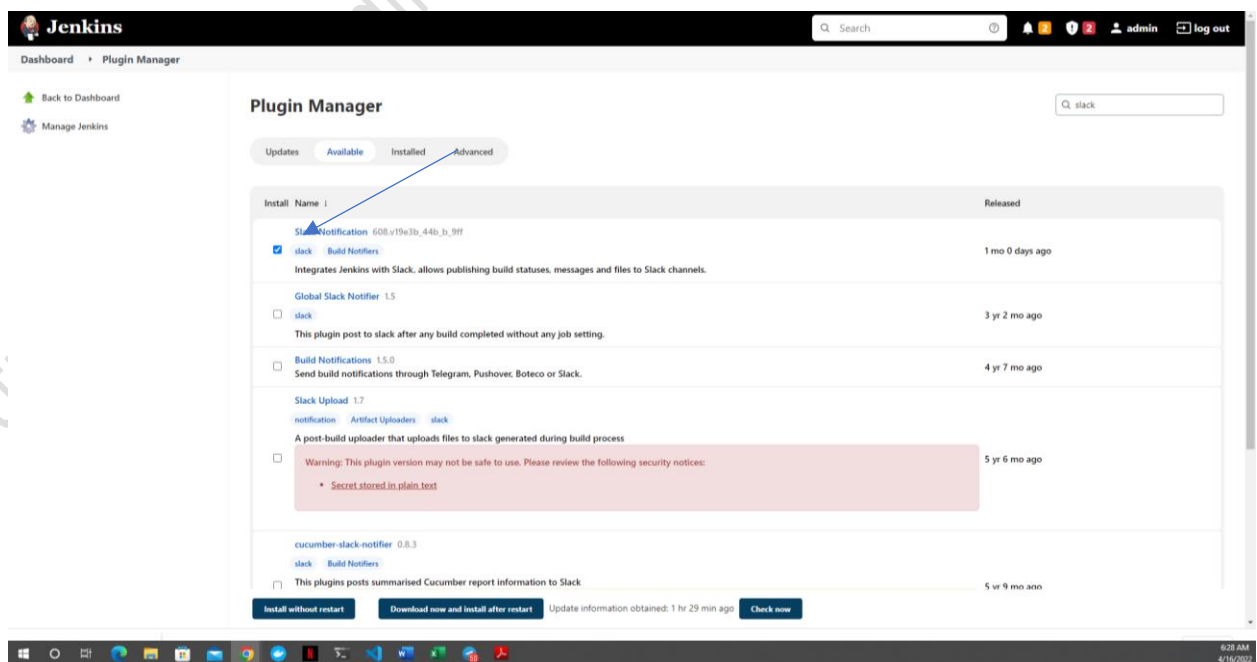
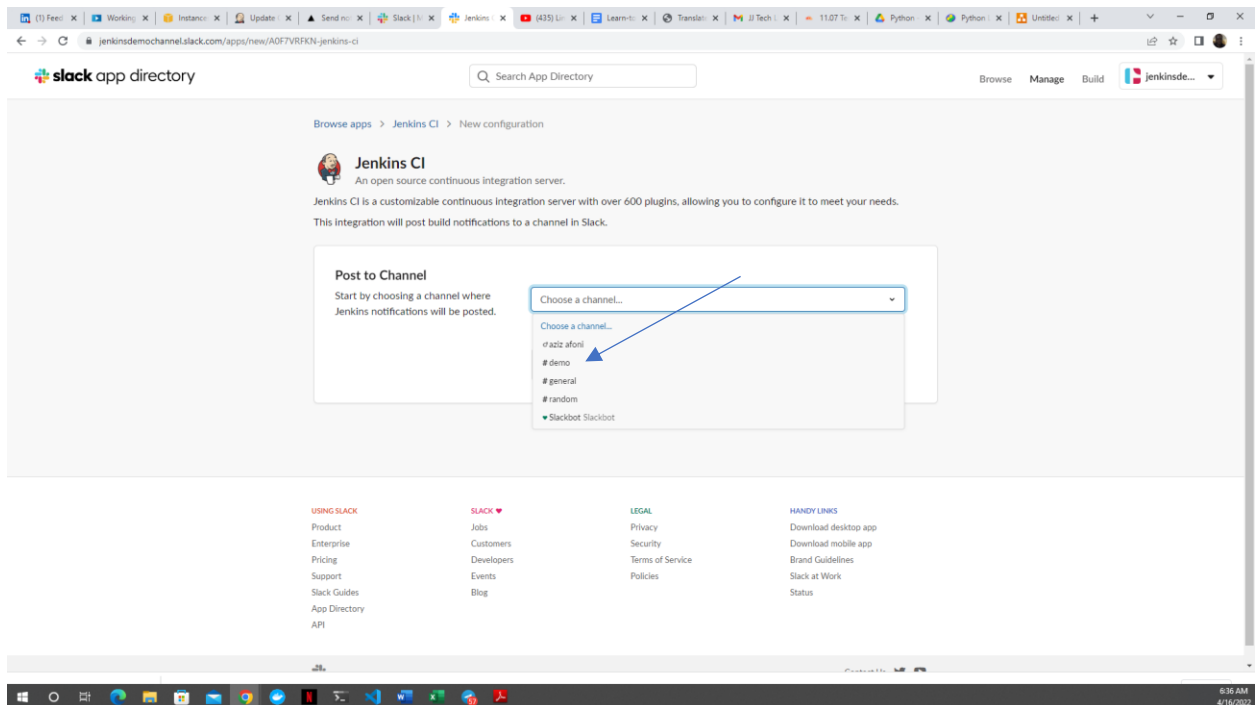- Select the plugin and scroll down click install without restart

2. Now we need to install jenkins in slack -> navigate to your slack workspace in the browser–> click on manage
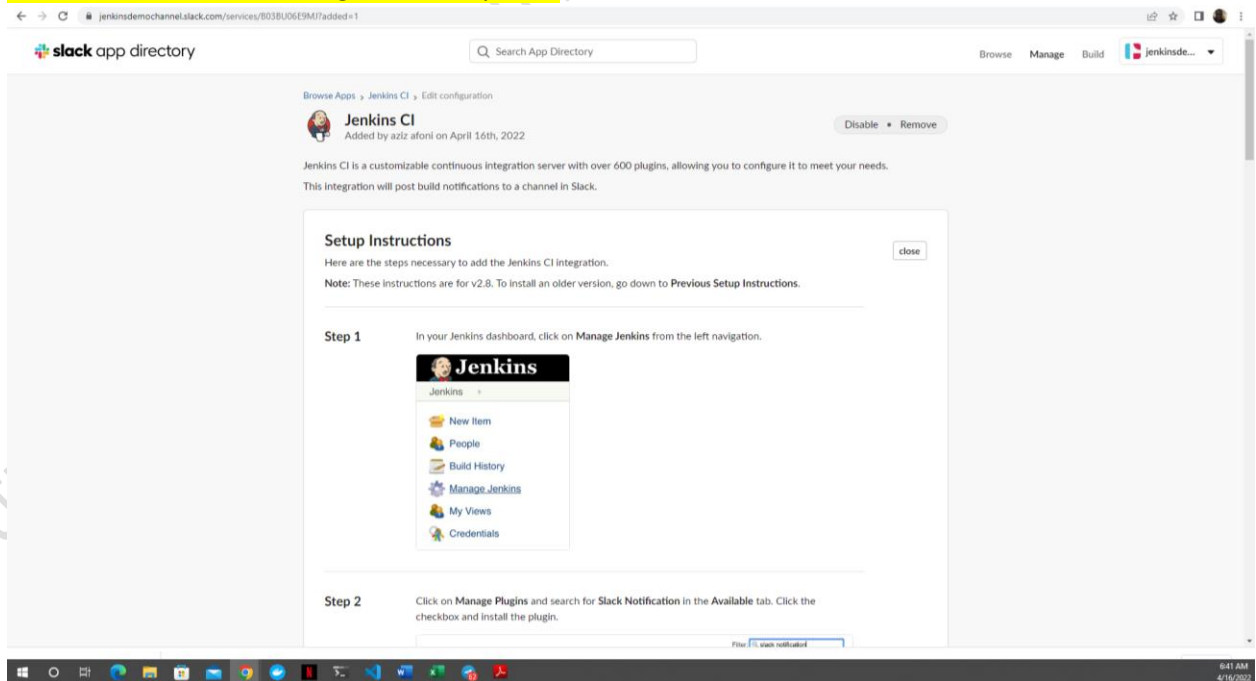

- On the search App directory search for jenkins CI
- Click ad to slack

- Choose the channel

- After selecting channel click add jenkins CI integration
- Slack will redirect you to a page with screenshots on how to configure slack on jenkins. since we installed the plugin already you can skip the first stage
  <mark>N: B take note of the token generated by slack</mark>

- Navigate back to jenkins and go to manage jenkins -> configure system-> find the global slack notifier setting and insert the credentials from the slack page
- <mark>i.e., the token and team subdomain</mark>

- depending on your jenkins version the interface might be different but the configuration remains the same

- input your workspace name
- click Add and select jenkins from dropdown-> select kind as secret text the paste the token from slack
- use description as slack
- after pasting token click save the select none beside Add -> from the drop down select the credentials description name you just added
- in default channel paste the channel from slack page
- click test connect to see if everything is working you should receive success as bellow and also in your slack channel

3. Configuration of project
- Navigate to manage jenkins->manage plugin ->available and search for pipeline plugin -> install the plugin
- Go back to jenkins dashboard and select -> New Item -> put job name ->select pipeline -> click ok

- Lets understand the stages first

```
pipeline
{
    agent any
    stages
    {
        stage('clone')
        {
            steps
            {
                echo 'stage 1'
            }
        }
        stage('Build')
        {
            steps
            {
```

```
                echo 'stage 2'
            }
        }


    }
    post
    {
        success
        {
            slackSend(message: "Build Successful")
        }

        failure
        {
            slackSend( failOnError: true, message: "Build Failed: ${env.JOB_NAME}
${env.BUILD_NUMBER}")


        }


    }


}
```

- In the first stage you pass the GitHub repo then in second stage you can pass the build command
- In the post build stage, the first blog is success. if the build is successful this blog is executed
- And if the build fails the failure blog is executed

- Open your pipeline job and paste the Jenkinsfile
- Paste the above code in pipeline and click apple and save -> click bild now on the left panel. Once the buils is complete you should receive a success message in your slack channel as bellow

- Now lets make the build fail I will do that by removing the 'o' from echo

```
ech 'stage 1'
```

- so, the command becomes invalid and the build will fail
- edit the pipeline then apply and save
- you should receive build failed in your slack channel
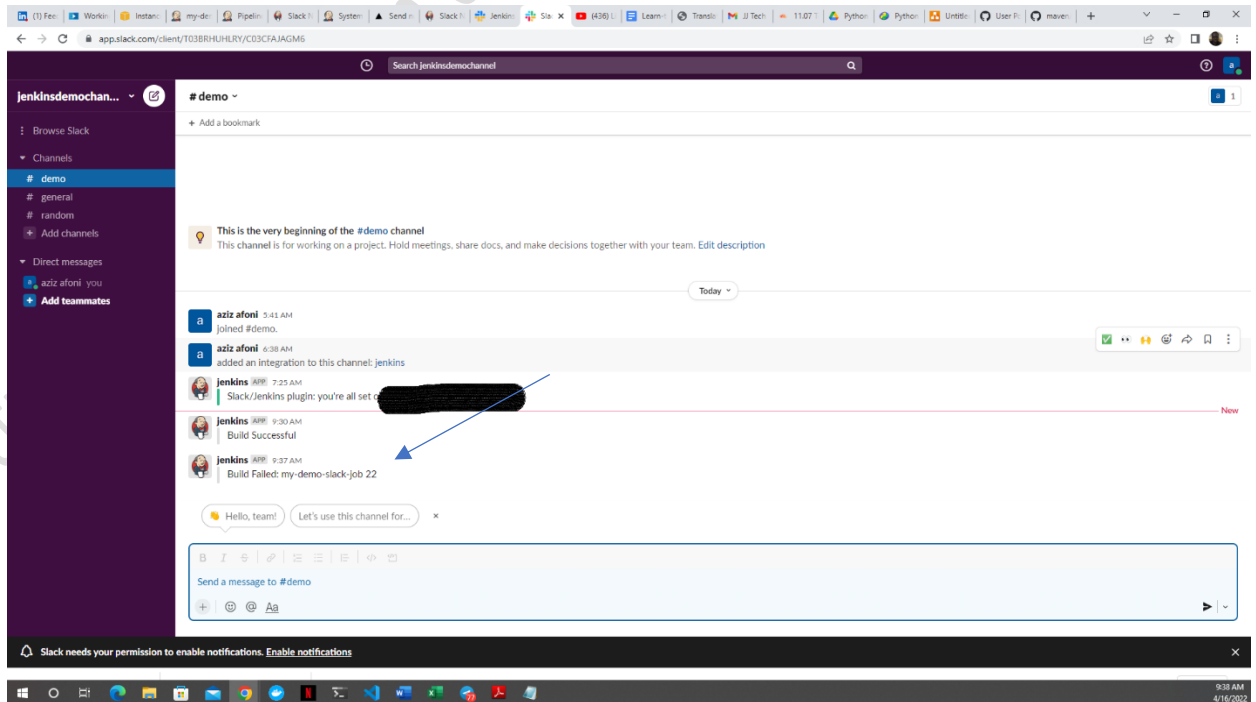
Please Follow my LinkedIn page for more

LinkedIn
https://www.linkedin.com/in/aziz-afoni-811503227/