

# ASSIGNMENT-3

---

**Course:** DevOps

**Name:** Vanguri Raja Vamsy

**Batch No:** 115 (9am-10am)

**Mail id:** [rajavamsyvanguri@gmail.com](mailto:rajavamsyvanguri@gmail.com)

**Trainer Name:** Mr. Madhukar Reddy

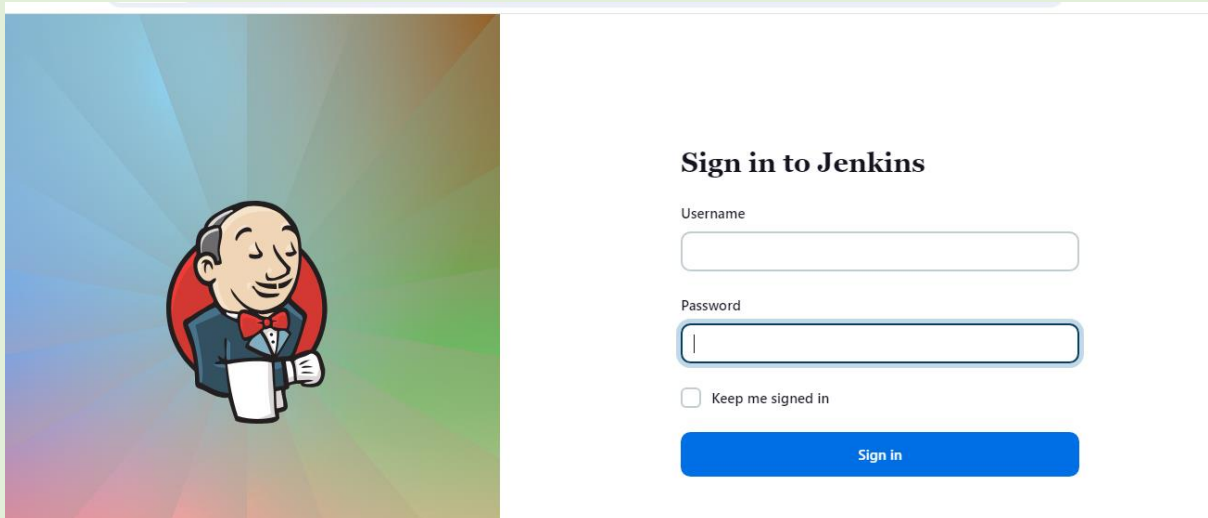
**Date:** 08-12-23

**Topic:** 1) Use Parameterize build in Jenkins pipeline

2) Try to implement email trigger from Jenkins pipeline

**1) Parameterize build:**

**JENKINS:**



This is the Sign page for Jenkins. Sign in to Jenkins add new item, create a job.

Dashboard > parameters > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

### General

Enabled ☒

Description

Plain text [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☒ This project is parameterized ?

**String Parameter** ?

Name ?

string

Default Value ?

vamsy

Description ?

Above shows the configure job creation in the Jenkins. By selecting “This project is parameterised”. Add parameter and select the string parameter. Give name string and Default value is vamsy.

Dashboard > parameters > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

**Boolean Parameter** ?

Name ?

boolean

☐ Set by Default ?

Description ?

Plain text [Preview](#)

**Choice Parameter** ?

Name ?

choice

Choices ?

DEV  
TEST  
QA  
UAT  
RE\_PROD

Description ?

Save Apply

Above Shows Adding the Boolean and choice parameters.

The screenshot shows the 'Configure' page in a web application. The left sidebar contains a 'Configure' section with a 'General' tab selected. The main content area displays the 'Credentials Parameter' configuration form. The form includes a 'Name' field with the value 'cred', a 'Credential type' dropdown menu set to 'Username with password', a 'Required' checkbox, a 'Default Value' dropdown menu set to 'aaaa', and a 'Description' field. At the bottom of the form are 'Save' and 'Apply' buttons.

Above figure shows the adding of credential parameter.

The screenshot shows the 'Configure' page in a web application. The left sidebar contains a 'Configure' section with a 'General' tab selected. The main content area displays two configuration forms. The first form is the 'Password Parameter' configuration, which includes a 'Name' field with the value 'passw', a 'Default Value' dropdown menu set to 'Concealed', and a 'Description' field. A 'Change Password' button is visible next to the 'Default Value' dropdown. The second form is the 'Multi-line String Parameter' configuration, which includes a 'Name' field with the value 'raja' and a 'Default Value' dropdown menu set to 'below all this is raja good morning'. At the bottom of the forms are 'Save' and 'Apply' buttons.

Above figure shows the adding of password parameter.

The screenshot shows the 'Configure' page with a sidebar on the left containing links to General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main content area is titled 'File Parameter' and includes a 'File location' field with the value 'file', a 'Description' text area, and a 'Plain text' preview link. Below these fields is an 'Add Parameter' button. Further down, there are checkboxes for 'Throttle builds' and 'Execute concurrent builds if necessary', both of which are unchecked. An 'Advanced' dropdown menu is also visible. At the bottom, there is a 'Source Code Management' section with a 'None' radio button selected. The page concludes with 'Save' and 'Apply' buttons.

Above figure shows the adding of file parameter.

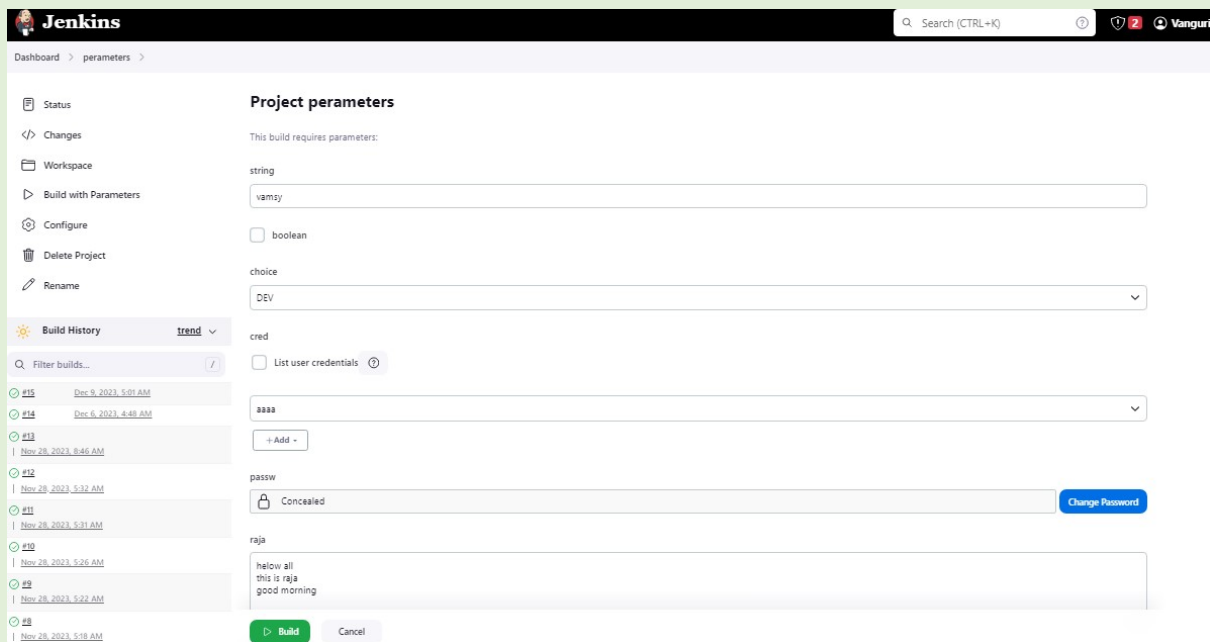
The screenshot shows the 'Configure' page with the 'Build Steps' section selected in the sidebar. The main content area is titled 'Execute shell' and includes a 'Command' field with a link to 'See the list of available environment variables'. The command field contains a shell script: 

```
#!/bin/bash
echo " this is my $string"
echo " print what you want to say $boolean"
echo " print what you want to $choice"
echo " I want to use my $cred"
echo " I want to know passwd $passwd"
echo " I want to $raja"
cat file
```

 Below the command field is an 'Advanced' dropdown menu. Further down, there is an 'Add build step' button. At the bottom, there is a 'Post-build Actions' section with an 'Add post-build action' button. The page concludes with 'Save' and 'Apply' buttons.

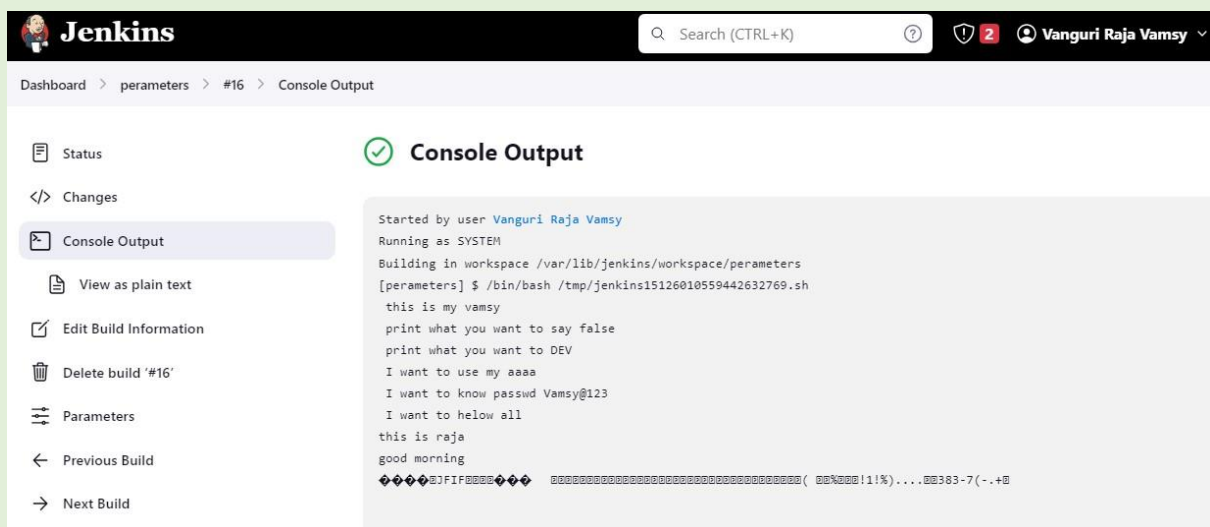
Above figure shows the Execute shell script, which is used to execute the parameters.

APPLY and SAVE.



The image shows the Jenkins 'Project parameters' configuration page. On the left is a sidebar with navigation links: Status, Changes, Workspace, Build with Parameters (selected), Configure, Delete Project, and Rename. Below these is the 'Build History' section showing a list of recent builds with their IDs and timestamps. The main area is titled 'Project parameters' and contains a list of parameters for the build. The parameters are: 'string' (a text input field containing 'vamsy'), 'boolean' (a checkbox), 'choice' (a dropdown menu showing 'DEV'), 'cred' (a checkbox for 'List user credentials'), 'aaaa' (a dropdown menu), 'passw' (a password field with a 'Change Password' button), and 'raja' (a text area containing 'hellow all', 'this is raja', and 'good morning'). At the bottom of the parameter list are 'Build' and 'Cancel' buttons.

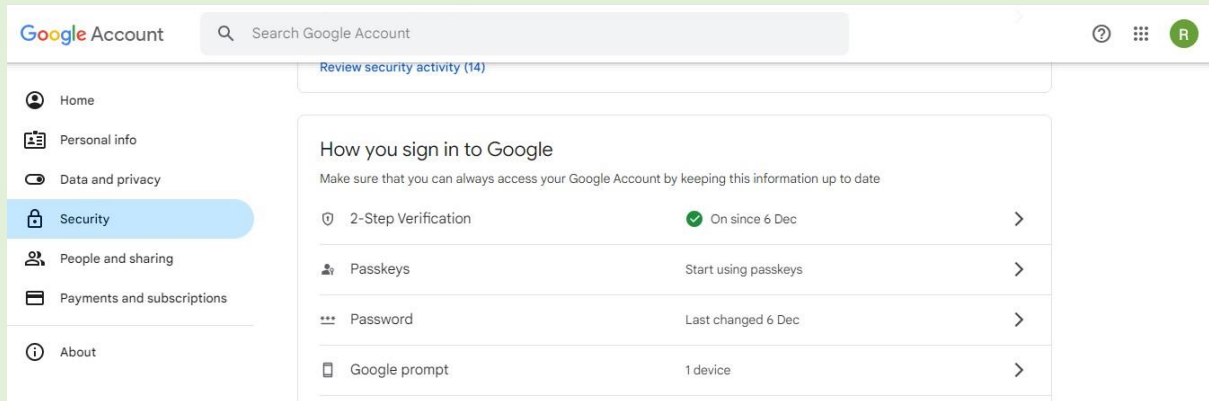
Above figure shows the Building process. Click on Build with parameters. select options and build.



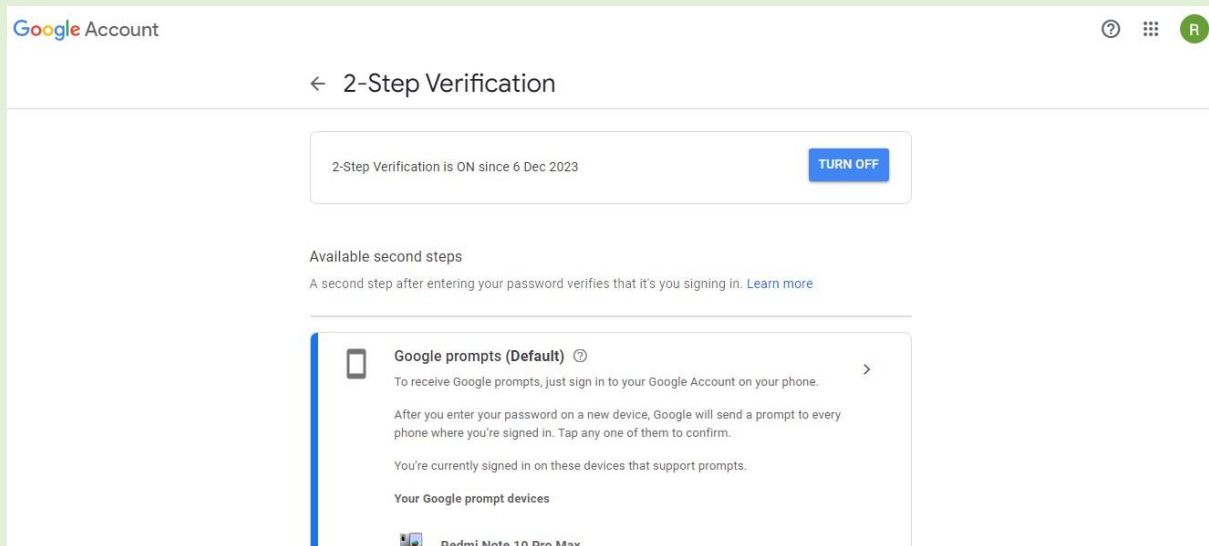
The image shows the Jenkins 'Console Output' page for build #16. The left sidebar has links for Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete build '#16', Parameters, Previous Build, and Next Build. The main area is titled 'Console Output' and displays the build's log. The log starts with 'Started by user Vanguri Raja Vamsy' and 'Running as SYSTEM'. It then shows the build directory and the execution of a shell script. The script's output is: 'this is my vamsy', 'print what you want to say false', 'print what you want to DEV', 'I want to use my aaaa', 'I want to know passwd Vamsy@123', 'I want to hellow all', 'this is raja', and 'good morning'. The log ends with a series of symbols and a timestamp.

Above figures shoes the **console output** of the parameter job.

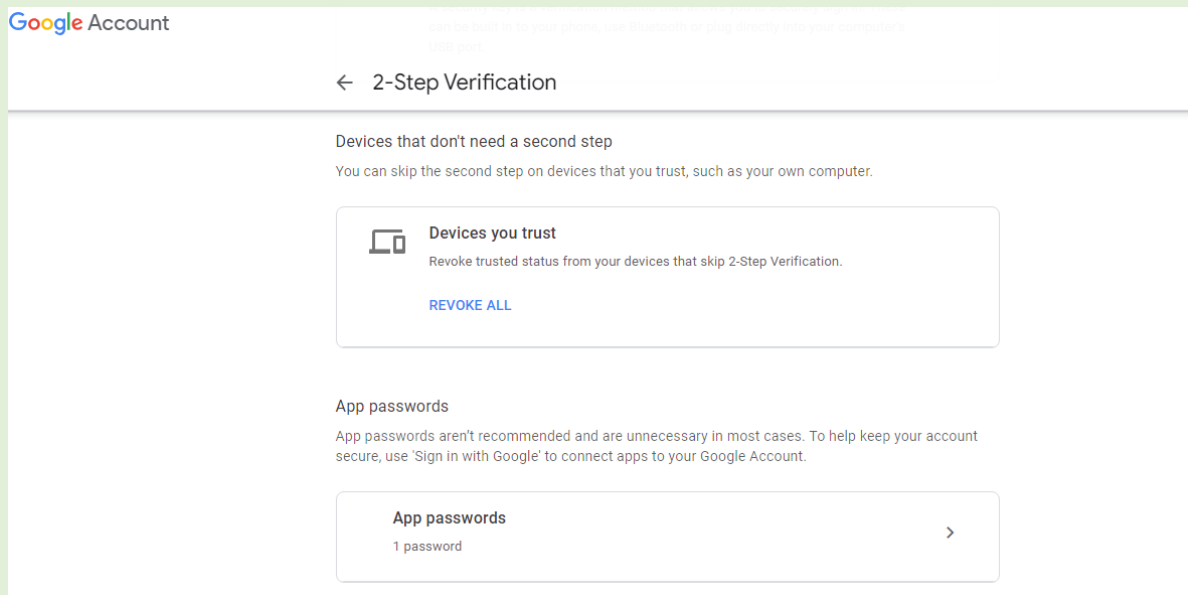
## 2)Email Trigger:



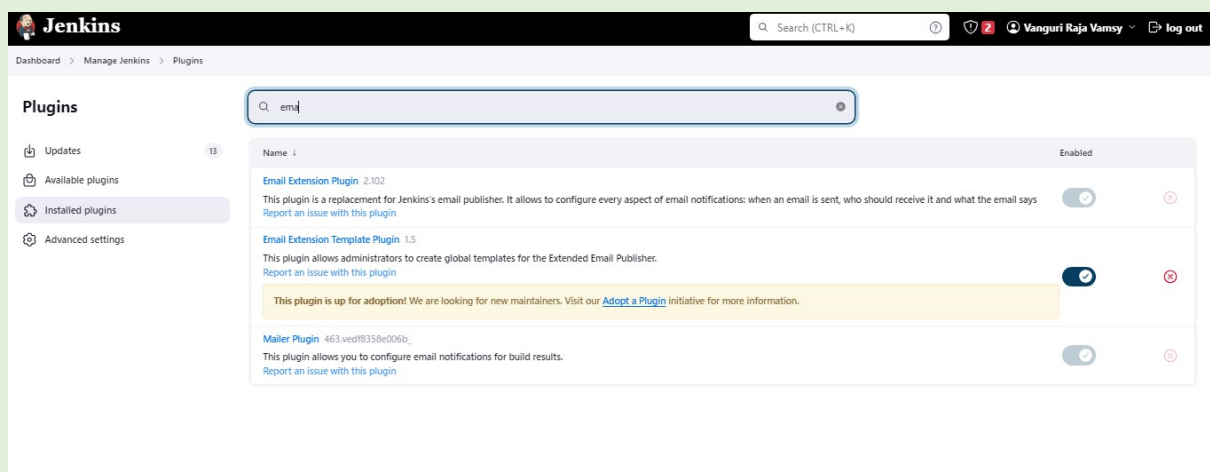
Open email, manage >>manage google account>> security>>2-Step Verification.



Turn on 2-step verification.



Go to APP passwords>> add jenkins create password>> copy the password and paste it in the jenkins email notification config.



Go to Manage Jenkins > plugins and install these plugins.

Dashboard > Manage Jenkins > System >

### E-mail Notification

SMTP server  
smtp.gmail.com

Default user e-mail suffix ?

Advanced ^ Edited

☒ Use SMTP Authentication ?

User Name  
rajavamsy745@gmail.com

Password  
Concealed [Change Password](#)

☒ Use SSL ?  
☐ Use TLS

SMTP Port ?  
465

Reply-To Address  
rajavamsy745@gmail.com

[Save](#) [Apply](#)

Go to manage jenkins>> system>> scroll down configure and give app password, apply and save.

Dashboard > email > Configuration

### Configure

[General](#)  
[Advanced Project Options](#)  
[Pipeline](#)

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Checkout') {
6       steps {
7         checkout([$class: 'GitSCM', branches: [[name: '*/dev']], userRemoteConfigs: [[url: 'https://github.com/
8       }
9     }
10
11    stage('Build') {
12      steps {
13        sh 'mvn package'
14      }
15    }
16    stage('email notification'){
17      email {
18        to 'rajavamsy745@gmail.com'
19      }
20    }
21  }
22 }
```

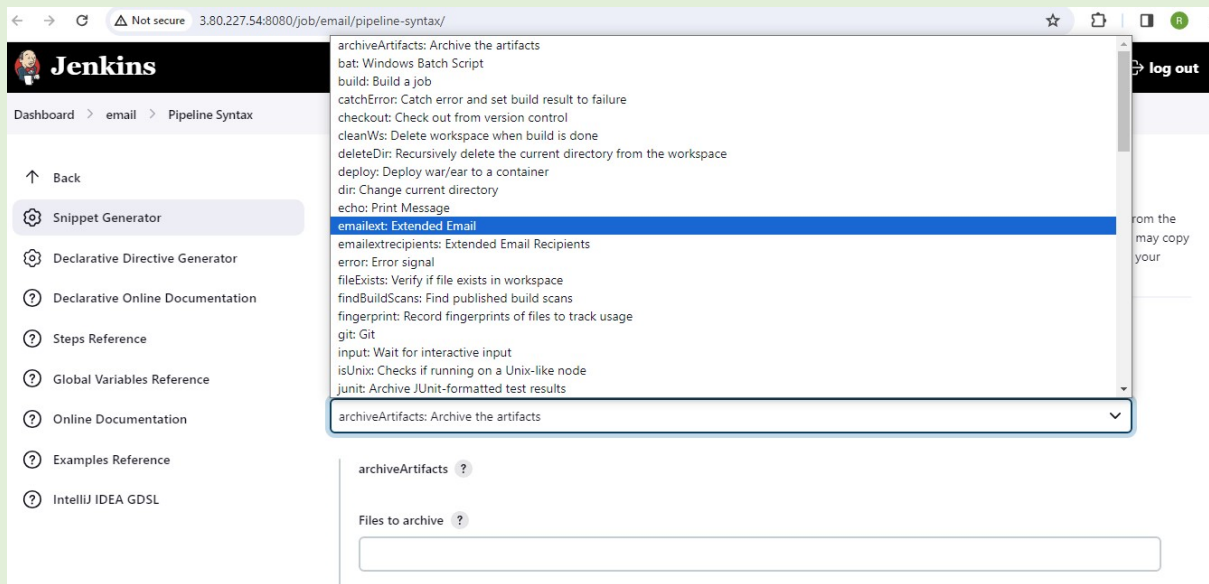
☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

[Save](#) [Apply](#)

NOW create the pipeline job.





Go to the pipeline syntax>> select the emailext Extended Email Recipients. And configure that by adding the git log in attachments.

## Pipeline Script:

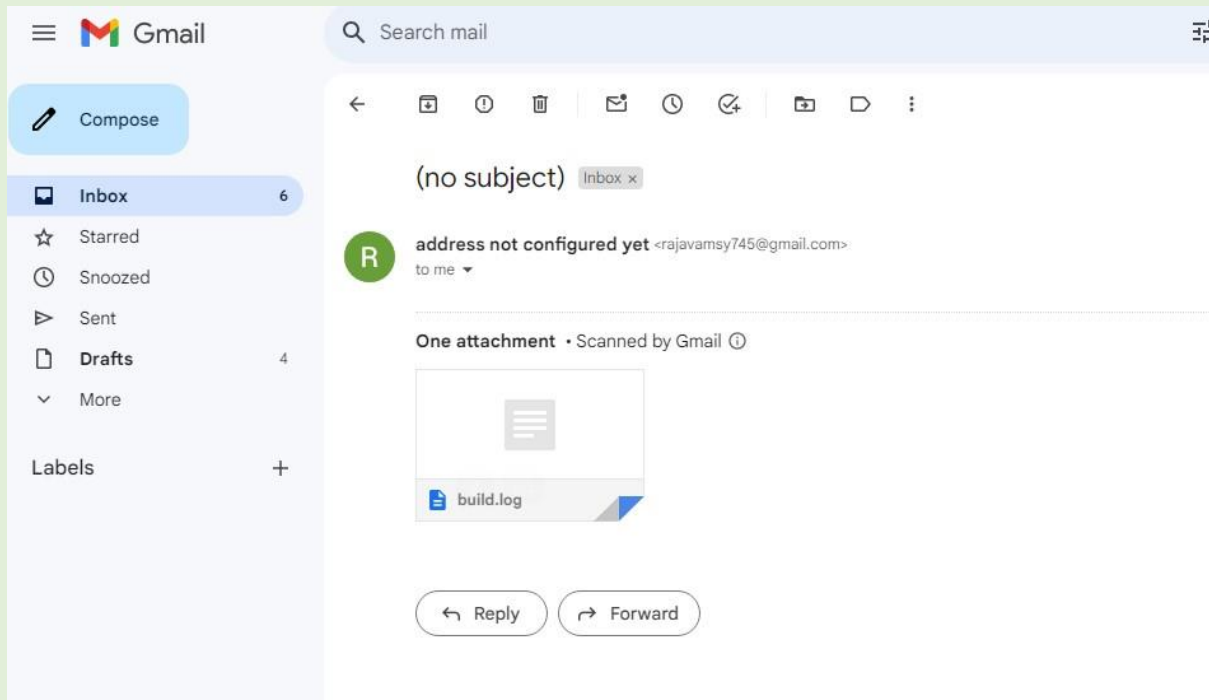
```
pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                checkout([$class: 'GitSCM', branches: [[name: '*/dev']],
userRemoteConfigs: [[url: 'https://github.com/VRaj745/taxi-booking.git']]])
            }
        }

        stage('Build') {
            steps {
                sh 'mvn package'
            }
        }

        stage('email notification'){
            steps{
```

Above shows the Console Output of the pipeline job.



This the RESULTANT OUTPUT email.



This the FINAL output for this Email Trigger.