

Email Integration with Jenkins and Plugin Setup

Step-1:

The screenshot shows the Jenkins Plugins page. In the top navigation bar, there are links for Dashboard, Manage Jenkins, and Plugins. The Plugins section is currently selected. A search bar at the top right contains the text 'email'. Below the search bar, there is a button labeled 'Install' with a dropdown arrow. The main area displays a list of plugins related to email:

Install	Name	Released
<input type="checkbox"/>	build user vars 195.v6c35f9d5c3dc	29 days ago
<input checked="" type="checkbox"/>	Email Extension Template 224.va_f6a_ecd5d46e	28 days ago
<input type="checkbox"/>	GitHub Integration 0.7.2	1 mo 11 days ago
<input type="checkbox"/>	View Job Filters 396.veea_3d19b_9551	5 days 15 hr ago

On the left sidebar, there are links for Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress.

Step-2:

- Go to your Gmail and click on your profile
- Then click on Manage Your Google Account → click on the security tab on the left side panel you will get this page (provide mail password).

The screenshot shows the Google Account Security page. The URL in the browser is myaccount.google.com/security?gar=WzEyMjM0&hl=en&utm_source=OGB&utm_medium=act. The left sidebar has a navigation menu with options: Home, Personal info, Data & privacy, Security (which is selected and highlighted in blue), People & sharing, Payments & subscriptions, and About. The main content area is titled 'Security' and includes the sub-section 'Your account is protected'. It states: 'The Security Checkup checked your account and found no recommended actions.' There is a 'See details' link. Below this is the 'Recent security activity' section, which says 'No security activity or alerts in the last 28 days'. At the bottom is the 'How you sign in to Google' section, which says 'Make sure you can always access your Google Account by keeping this information up to date.' It shows '2-Step Verification' is turned 'On since May 2, 2023'.

- On search bar type app you will see app password

Google Account

app

Google Account results

- Your connections to third-party apps & services
- App passwords**
- Web & App Activity
- Help Center articles
- Sign in with app passwords
- Manage app info from your devices
- Share some access to your Google Account with third-party apps

2-Step Verification phones	094928 30601
Recovery phone	094928 30601
Recovery email	kaveridevid@gmail.com

You can add more sign-in options

Authenticator **Backup 2-Step Verification phones** **Backup codes**

Privacy Terms Help About

Google Account

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.

[Learn more](#)

You don't have any app passwords.

To create a new app password, type a name for it below...

App name: Jenkins

Create

Google Account

← App passwords

Generated app password

Your app password for your device

gjln wzjl ooyj lkcs

How to use it

Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.

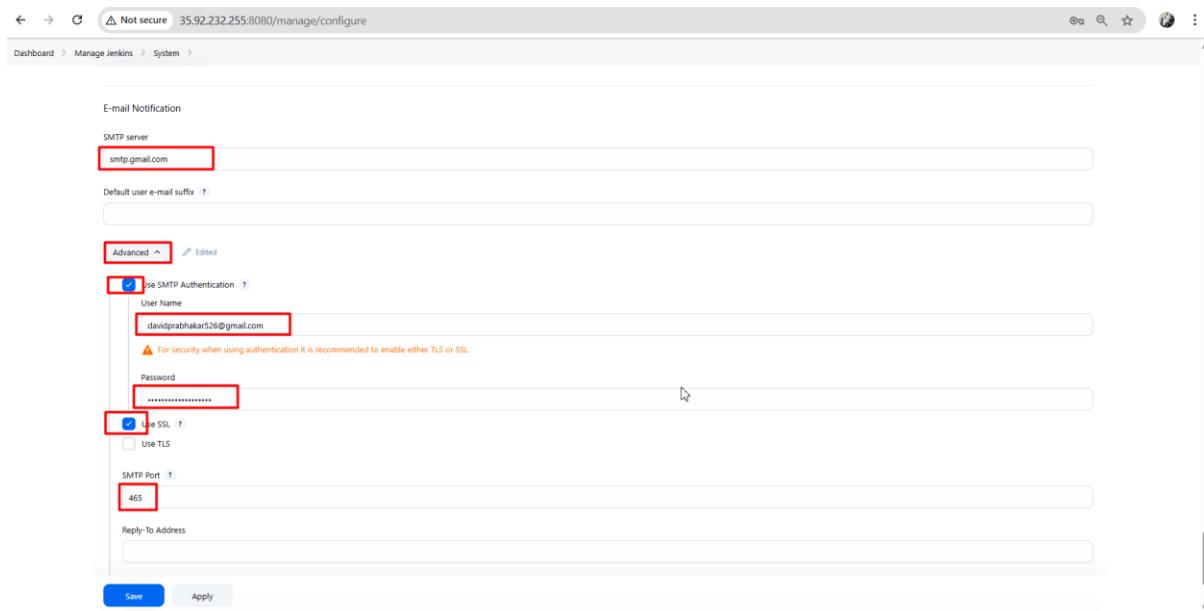
Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

Done

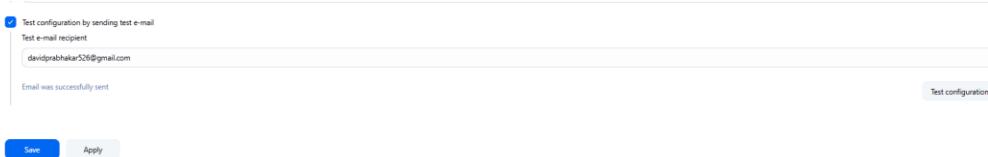
- Copy n save the passwd

Once the plugin is installed in Jenkins, click on manage Jenkins → configure system there under the E-mail Notification section configure the details as shown in the below image

- Configure as below highlighted



- Provide ur gmail for testing purpose



- After check ur mail it will send on test mail



- After click on apply and save

- Click on Manage Jenkins → credentials and add your mail username and generated password

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

davidprabhakar526@gmail.com

Treat username as secret ?

Password ?

ID ?

mail

Description ?

mail

Create

Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

+ Add Credentials

ID	Name	Kind	Description
mail	davidprabhakar526@gmail.com/***** (mail)	Username with password	mail

Icon: S M L

Now under the Extended E-mail Notification section configure the details as shown in the below images

Dashboard > Manage Jenkins > System >

Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

465

Advanced ^ (0 filled)

Credentials

davidprabhakar526@gmail.com/***** (mail)

+ Add

Use SSL

Use TLS

Use OAuth 2.0

Advanced Email Properties

Default user e-mail suffix ?

Press the 'Ctrl Shift' on your keyboard to take a screenshot

- Next follow steps ↗

Dashboard > Manage Jenkins > System >

Default user e-mail suffix:

Advanced ^ Edited

Charset: UTF-8

Additional accounts: Add

Default Content Type: HTML (text/html)

List ID:

- Next click on additional groovy classpath > default triggers ↗

Dashboard > Manage Jenkins > System >

Additional groovy classpath ?

Add

Enable Throttling ?

Enable Debug Mode ?

Require Administrator for Template Testing ?

Enable watching for jobs ?

Allow sending to unregistered users ?

Default Triggers ^

Default Triggers ?

- Aborted
- Always
- Before Build
- Failure - 1st
- Failure - 2nd
- Failure - Any
- Failure - Still
- Failure - X
- Failure -> Unstable (Test Failures)
- Fixed
- Not Built
- Script - After Build
- Script - Before Build
- Status Changed
- Success
- Test Improvement
- Test Regression
- Unstable (Test Failures)
- Unstable (Test Failures) - 1st
- Unstable (Test Failures) - Still
- Unstable (Test Failures)/Failure -> Success

Content Token Reference ?

- After click on apply n save ↗

SonarQube – Jenkins pipeline script

```
pipeline {

    agent any

    environment {
        SCANNER_HOME = tool 'sonar-scanner'
    }

    stages {
        stage('Checkout') {
            steps {
                git branch: 'main', credentialsId: 'githubbt-token', url: 'https://github.com/Davidgit526/Admin-Retail-Application.git'
            }
        }

        stage('SonarQube Analysis') {
            steps {
                withSonarQubeEnv("sonar") {
                    sh """
${SCANNER_HOME}/bin/sonar-scanner \\
-Dsonar.projectKey=Retail-app-code-quality \\
-Dsonar.projectName=Retail-app-code-quality
"""
                }
            }
        }
    }
}
```

```
}
```

```
}
```

```
}
```

```
post {
```

```
    success {
```

```
        emailext (
```

```
            subject: "SUCCESS: SonarQube Quality Gate Passed",
```

```
            body: """
```

```
Hi Team,
```

The SonarQube quality gate passed successfully.

You can view the report here:

<http://44.227.100.103:9000/dashboard?id=Clahan-app-code-quality>

username: admin, passwd: pass

If you want to download a PDF report, please use the SonarQube interface or a PDF plugin.

Regards,

Jenkins

"""",

```
        to: 'muraliclahan526@gmail.com'  
    )  
}  
  
failure {  
    emailext (  
        subject: "FAILED: SonarQube Quality Gate",  
        body: """  
        Hi Team,  
    )
```

The SonarQube quality gate has failed.

Please check the details:

<http://44.227.100.103:9000/dashboard?id=Clahan-app-code-quality>
username: admin, passwd: pass

Regards,

Jenkins

""",
 to: 'muraliclahan526@gmail.com'
)

```
}  
}
```

}

