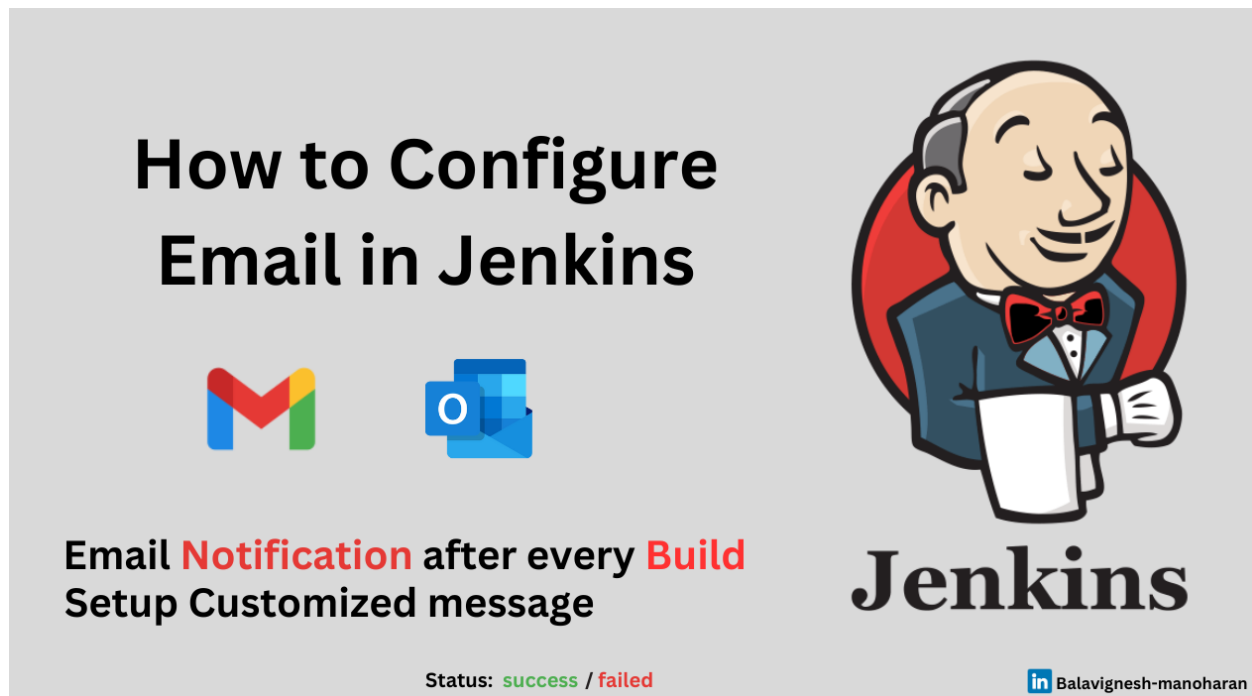


Jenkins Mail Configuration - Gmail & Outlook



Introduction:

Jenkins is an open source automation server that can be used for many different actions. It is mostly used for continuous integration and continuous delivery process in which we automate the build, test and deploy the code.

Email notifications are super important because they allow developers, testers and stakeholders to stay informed about the build and the deployment status. These email notifications can be sent automatically whenever the build is success, or failed or stopped due to any other reason,

In this article, we will see two different email services configured to receive the email from Jenkins.

Pre-requisites:

1. Gmail Account

2. Outlook Account
3. AWS Account

Basically there are two ways in which jenkins provides to send the email notifications.

- **Default Email Notifier:** This is a basic email notification plugin which is provided by default in jenkins.
- **Email Extension Plugin:** This is a more advanced feature of email notification that provides additional features such as customized email notifications, email templates and more.

In this article, we will configure the Email Extension Plugin so that we can set customize message and templates.

We will configure the email notification for both the email services.

GMAIL Configuration:

1. First step is to create the app password. This password will be used instead of the Gmail password for authentication.
2. Go to <https://myaccount.google.com/> and on the left hand side click on security.
3. Make sure your 2-step verification is turned on.
4. Once you're in the security console, search for app password in the search section.
5. Give a name for this app password, and click on create.
6. Copy your app password somewhere safe, we will need it later.

Jenkins Server Configuration for Gmail:

1. Go to AWS account, search for EC2 and launch an new instance with all default configuration.
2. Make sure to enable Port 8080 in the security group.
3. Once the server is launched, connect to it using your terminal.
4. Update your server using the below command.

```
sudo apt update
```

5. Install java using the below command.

```
sudo apt install openjdk-17-jre-headless -y
```

6. Install jenkins using the below command

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
```

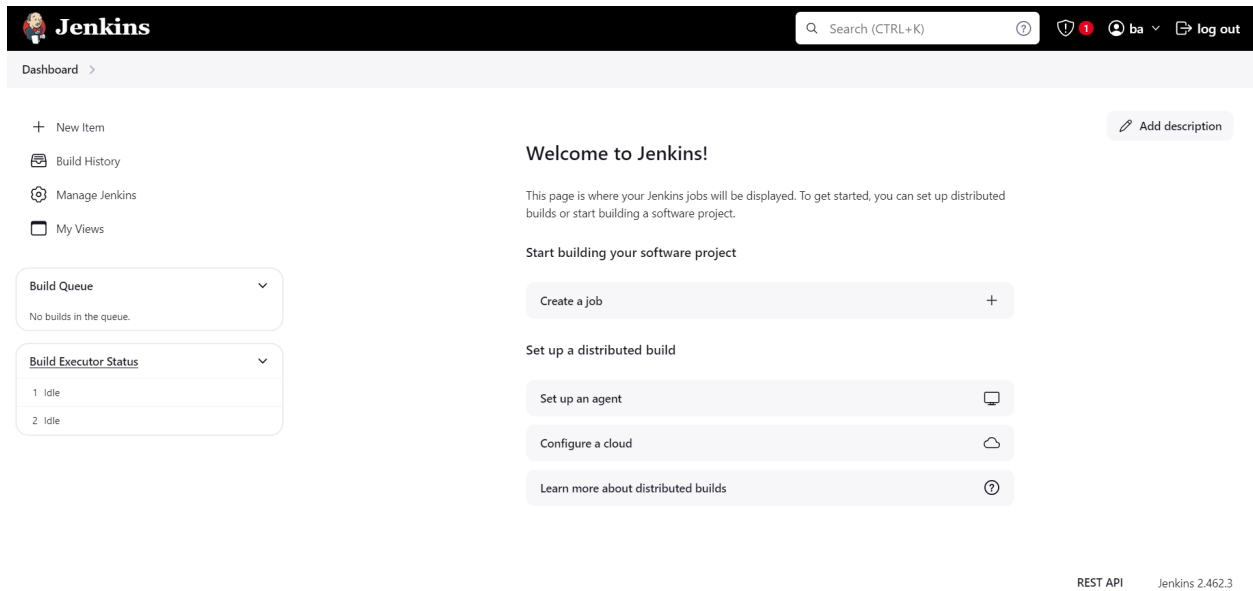
7. Access jenkins server using your public IP and with port 8080. (Open Port 8080 in your security group).
8. Get your initial admin password from the server. Use the below command and run it in your terminal

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

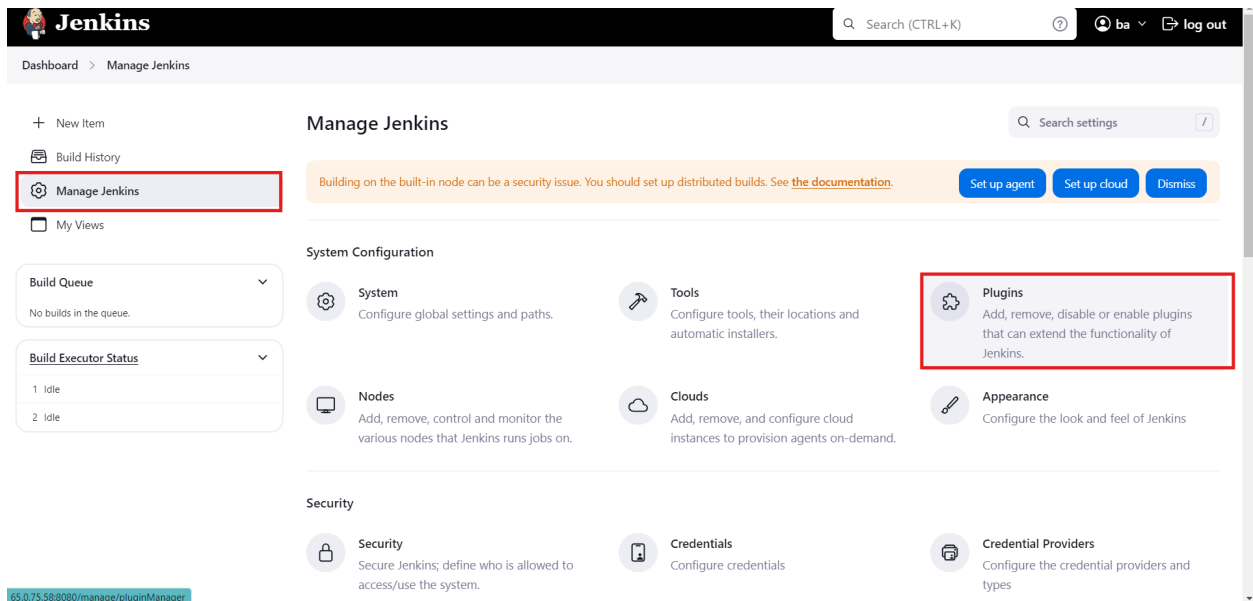
9. Copy the generated password and paste it in jenkins server.
10. Click on "Install suggested plugins".
11. Create first admin user by filling all the details.

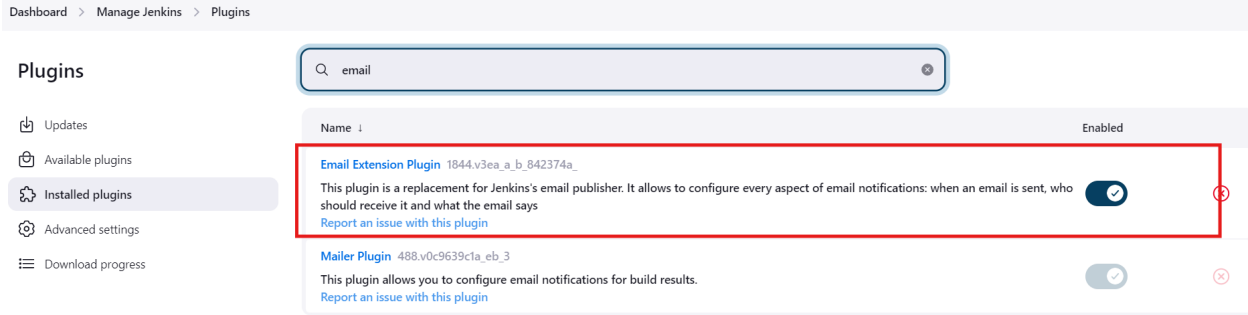
12. Click on save and continue, and click on start jenkins.

13. You will see welcome to jenkins home page.

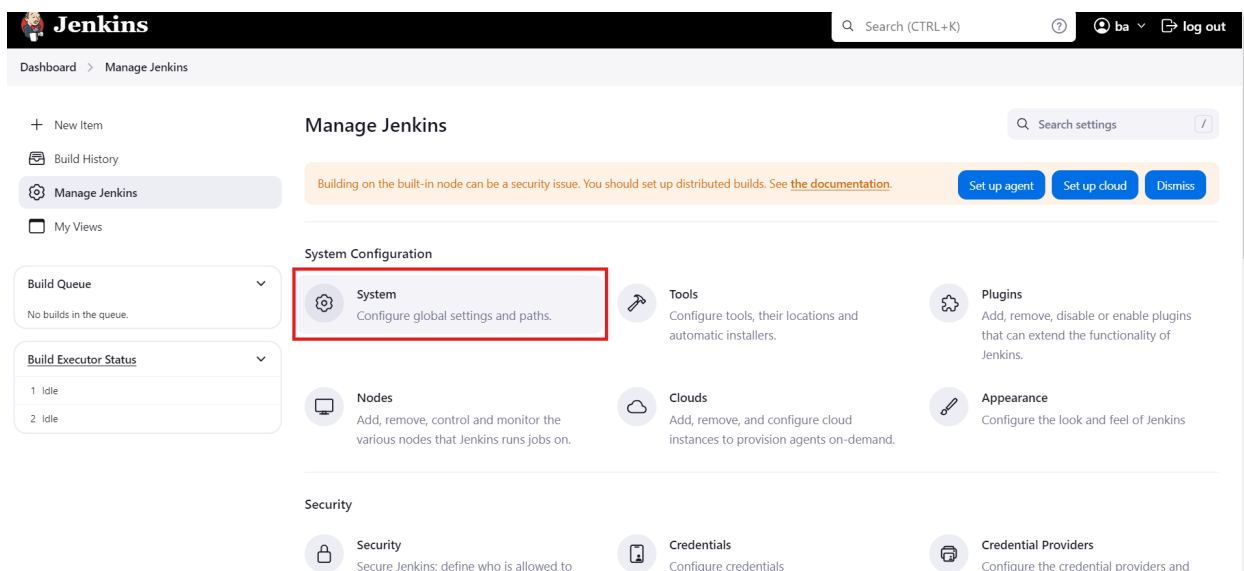


14. Go to manage jenkins > click on plugins and check in the installed plugin if you have email extension plugin installed. If not, click on available plugin and install the same.





15. Click on manage jenkins again, and click on system to configure the email.



16. Scroll down to the bottom, and check for **"Extended Email notification"** and put the below values.

- SMTP Server: smtp.gmail.com
- SMTP Port: 587
- Use TLS: Enable it
- Click on add credentials and Give your Username (Gmail username) and Password (App password) and save it.
- Make sure to select the credentials from the dropdown.
- In the default email suffix give the value as **"@gmail.com"**.
- In the reply to list give the value as your **"Email ID"**.

- Set the default trigger to “**Always**”.

17. In the E-mail notification section

- SMTP Server: smtp.gmail.com
- Use TLS: Enable it
- Use SMTP Authentication: Put your Username and App password
- Reply-to-Address: Add your Email ID here as well

18. Apply and Save

19. Create a simple pipeline job and paste the below pipeline syntax in the script section.

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
    }

    post {
        always {
            script {
                def jobName = env.JOB_NAME
                def buildNumber = env.BUILD_NUMBER
                def pipelineStatus = currentBuild.result ?: 'UNSTABLE'
                def bannerColor = pipelineStatus.toUpperCase()

                def body = """
                <html>
```

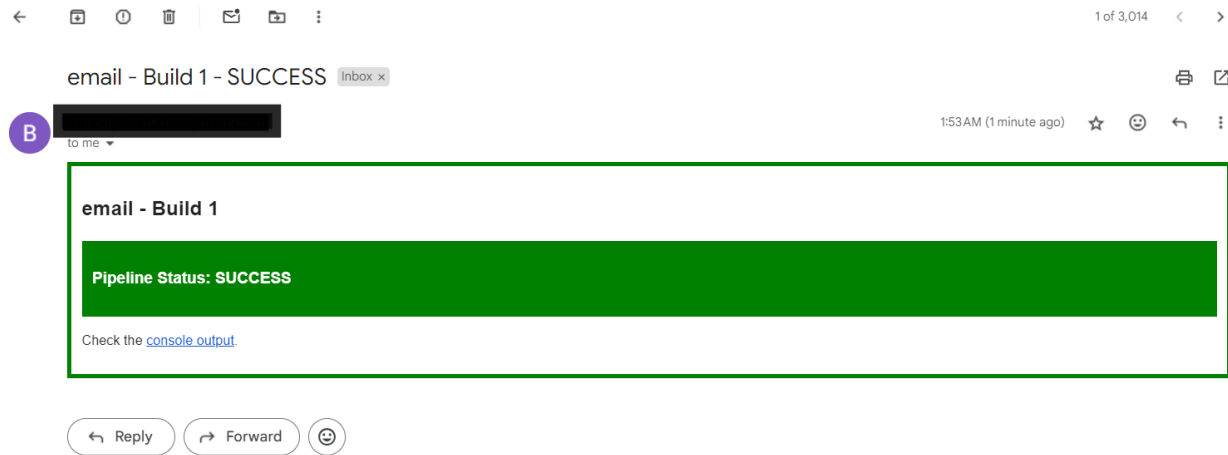
```
<body>
    <div style="border: 4px solid ${bannerColor}>
        <h2>${jobName} - Build ${buildNumber}</h2>
        <div style="background-color: ${bannerColor}>
            <h3 style="color: white;">Pipeline</h3>
        </div>
        <p>Check the <a href="${env.BUILD_URL}">build</a> for details</p>
    </div>
</body>
</html>
"""

emailtext(
    subject: "${jobName} - Build ${buildNumber}",
    body: body,
    to: 'YourEmailID',
    from: 'YourEmailID',
    replyTo: 'YourEmailID',
    mimeType: 'text/html'
)
```

Replace '**YourEmailID**' with your own Email ID. Apply and Save.

20. Click on build now.

21. If your build is successful, you will receive a mail in your gmail account.



22. We will also check for any error in pipeline, and check the notification.

23. Just edit the echo statement, keeping everything same.

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo abcd
            }
        }
    }

    post {
        always {
            script {
                def jobName = env.JOB_NAME
                def buildNumber = env.BUILD_NUMBER
                def pipelineStatus = currentBuild.result ?: 'UNKNOWN'
                def bannerColor = pipelineStatus.toUpperCase() :

                def body = ""
```



```

<html>
  <body>
    <div style="border: 4px solid ${bannerColor}>
      <h2>${jobName} - Build ${buildNumber}</h2>
      <div style="background-color: ${bannerColor}>
        <h3 style="color: white;">Pipeline</h3>
      </div>
      <p>Check the <a href="${env.BUILD_URL}">Build</a> for details</p>
    </div>
  </body>
</html>
"""

emailext(
    subject: "${jobName} - Build ${buildNumber}",
    body: body,
    to: 'YourEmailID',
    from: 'YourEmailID',
    replyTo: 'YourEmailID',
    mimeType: 'text/html'
)
}
}
}
}
}

```

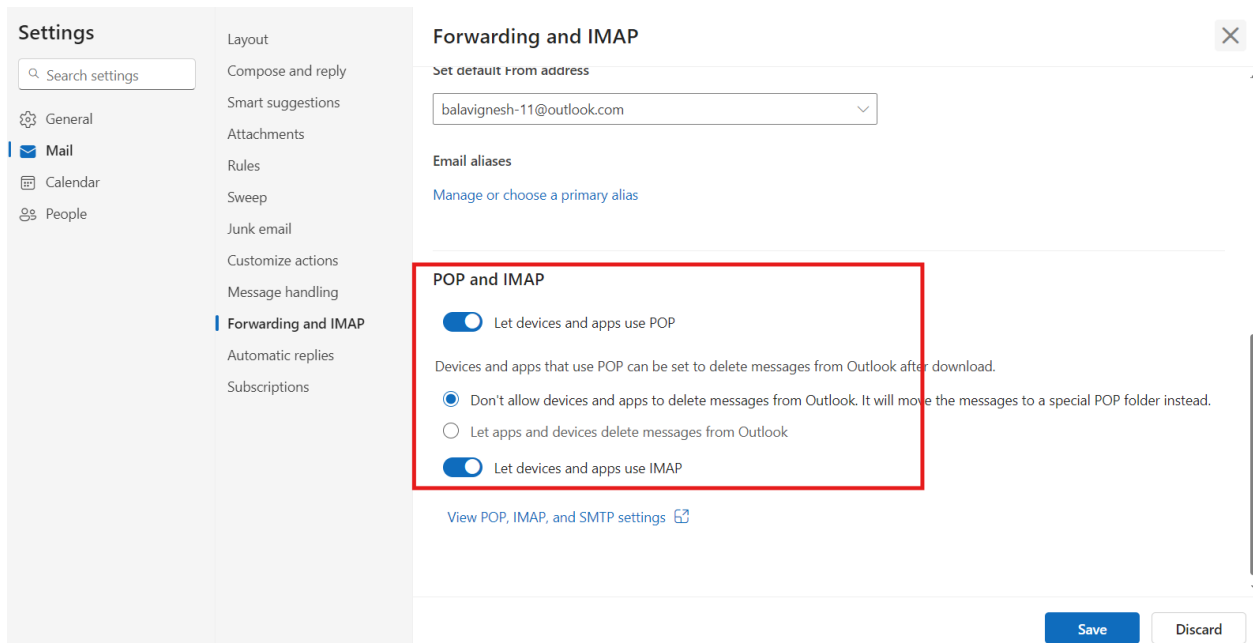
Save this code, and build your job again.



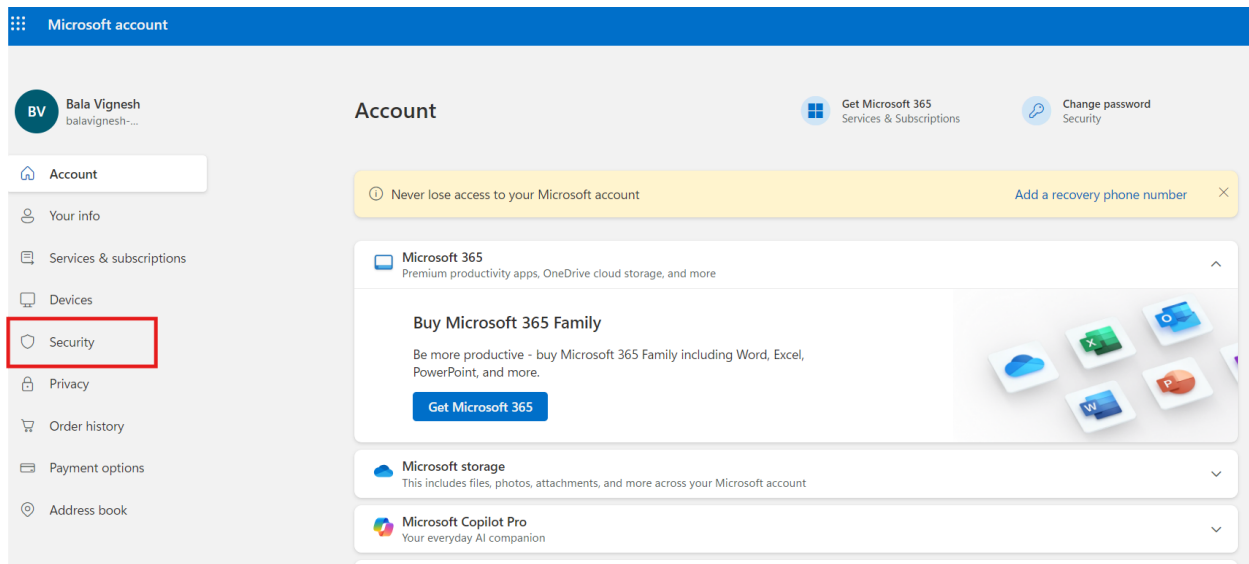
We have successfully configured Email notification for Gmail Service.

Outlook Configuration

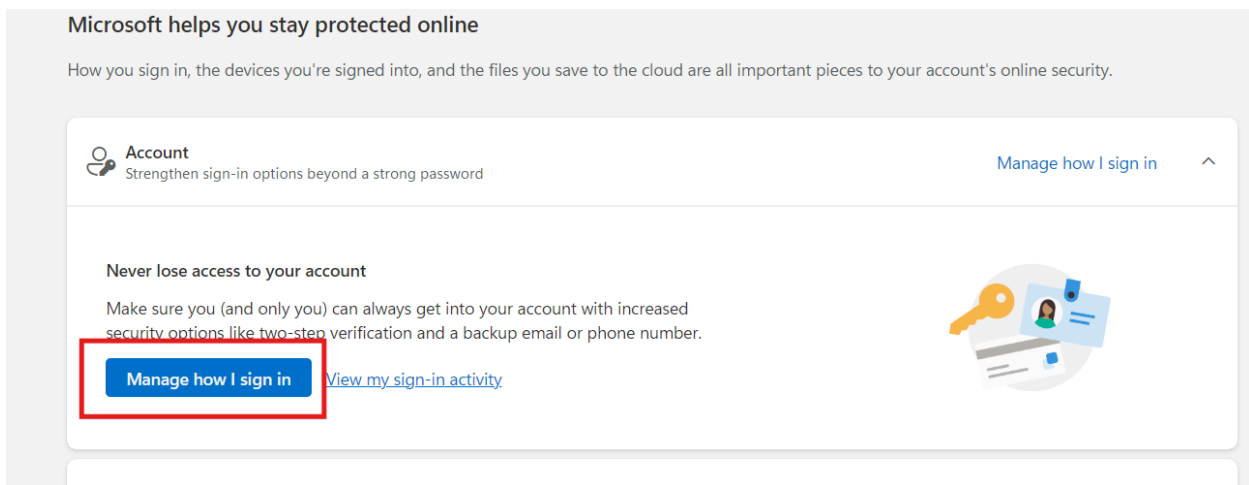
1. Login into your outlook account.
2. Click on the settings button on the top right of the page.
3. Click on the mail, and forwarding and IMAP and enable both of them. Save it.



- Go to <https://account.microsoft.com/> Click on security on the left side of the page.



- Click on 'Manage How I Sign in '



- Enable the **Two-step verification**.
- Once you've enabled the Two-step verification, you will be able to see the App password section below.
- Click on the Create a new app password.

App passwords

Some apps and devices (such as Xbox 360, Windows Phone, or mail apps on your other devices) don't support security codes for two-step verification. [Learn more about app passwords.](#)

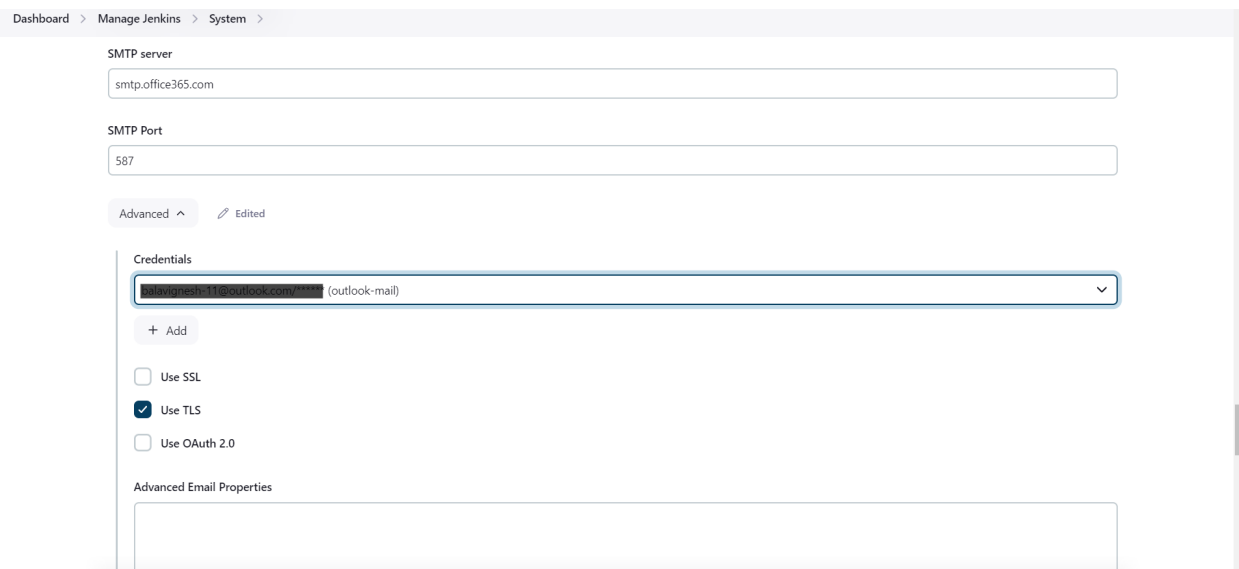
[Create a new app password](#)

9. Copy this password for further use.

Jenkins Server Configuration for Outlook:

Every settings remains the same as we did for Gmail. The only difference will be changing few parameters which I have mentioned below.

- SMTP Server: smtp.office365.com
- Username: Outlook Email ID
- Password: Outlook Generated App password
- Default E-mail suffix: @outlook.com

The screenshot shows the Jenkins 'System' configuration page for SMTP. The 'SMTP server' field is set to 'smtp.office365.com' and the 'SMTP Port' is '587'. Under the 'Advanced' section, the 'Credentials' dropdown is set to 'JenkinsMail1@outlook.com (outlook-mail)'. There are checkboxes for 'Use SSL' (unchecked), 'Use TLS' (checked), and 'Use OAuth 2.0' (unchecked). The 'Advanced Email Properties' section is currently empty.

Replace the **Gmail username** with the **Outlook username** when you're configuring for Outlook service. Rest all the settings remain same other than the one mentioned above.

Create a similar job for the Outlook mail demo.

1. Create a new pipeline job and use the same script as before, but replace the Email ID with your Outlook account address.
2. Apply and save, build the job.
3. Once the build is completed, check your outlook email inbox for the notification of the build job.

If you did not receive the email, check the jenkins console output for any error messages related to the email notification.

That's it! You have successfully configured email notification in jenkins.

Conclusion:

In conclusion, setting up the email notification in jenkins is much a simpler process and can be completed in few steps:

- Create the app password first, and save it.
- Go to Jenkins and check if the email extension plugin is installed, if not then install the same.
- Configure the SMTP server settings.
- Add the details in the email notification section.
- Create a job and add the pipeline script.
- Once the job is executed, check the email.

It's that simple!

By following these six steps, you can ensure that you and your team are always upto date on the status of your Jenkins builds and can quickly address any issues that arise anytime.

Stay tuned!

If you found this post useful, give it a like 👍

Repost 🔄

Follow @Bala Vignesh for more such posts 🎯 100 🚀

