**Lesson-End Project**

**Sending Test Reports Using Email Notification**

**Project agenda:** To set up a Jenkins pipeline to compile code, run JUnit tests, and distribute test reports via email notifications

**Description**: Imagine a development team using Jenkins for automated email notifications of test reports, improving code quality and speed. Upon code commits, Jenkins triggers a pipeline that compiles, tests, and emails test results via an SMTP setup. This quick feedback loop is vital for agile practices, ensuring that developers immediately identify and address issues.

**Tools required:** Jenkins

**Prerequisites:** You must have Jenkins and GitHub access in the lab to proceed.

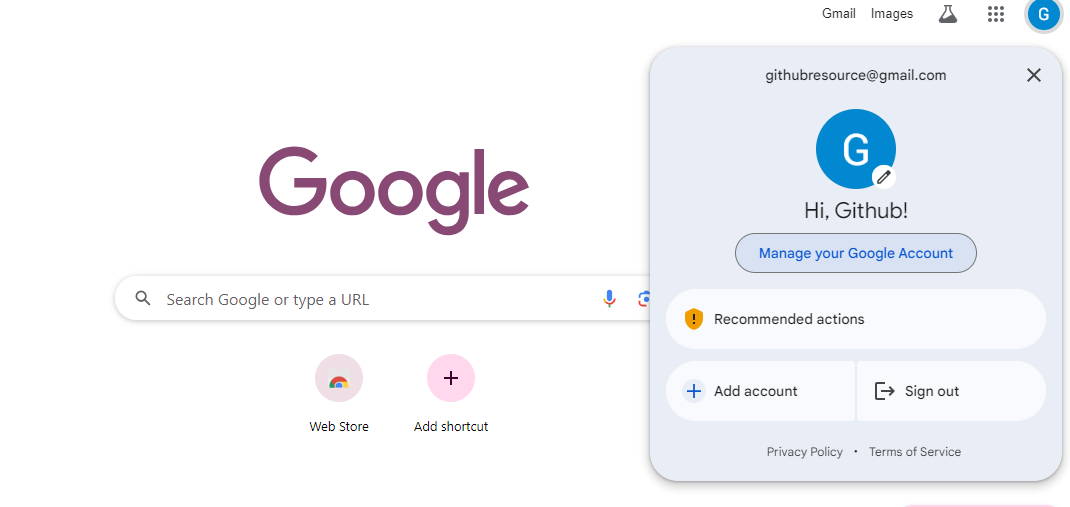
**Expected deliverables:** A fully configured Jenkins pipeline that compiles code, executes JUnit tests, and sends detailed test reports via email upon each build's completion

Steps to be followed:

1. Create an app password for SMTP configuration
2. Configure SMTP configurations in Jenkins
3. Configure Maven in Jenkins

**Step 1:** **Create an app password for SMTP configuration**

* 1. Navigate to your Google account and click on **Manage your Google Account**

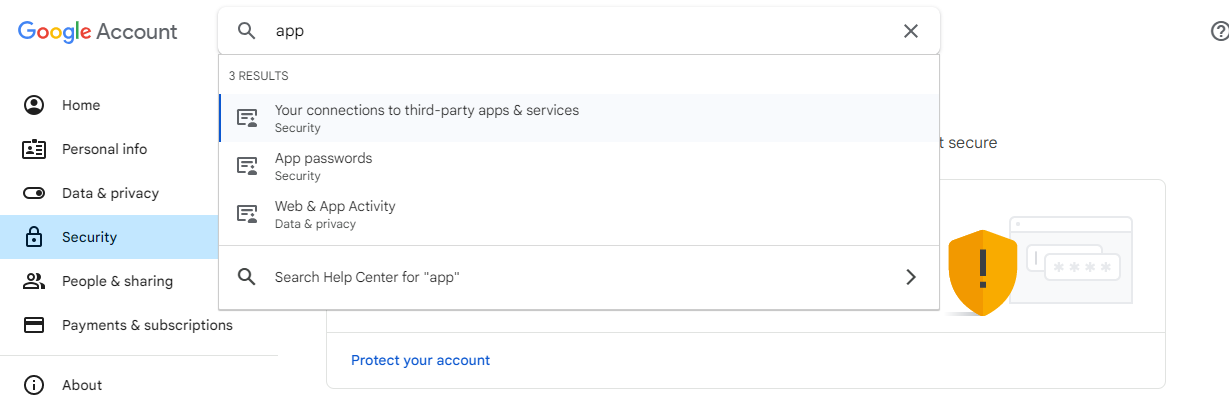


* 1. Now, navigate to **Security** and make sure you have **2-Step Verification** enabled

**A screenshot of a computer

Description automatically generated**

* 1. In **Security**, search for **App passwords** and click on it



* 1. Provide the app name as **Jenkins** and click on **Create**

A screenshot of a computer

Description automatically generated

* 1. It will generate the app password. Copy the password and save it for future reference, then click on **Done**.

A screenshot of a computer

Description automatically generated

**Step 2:** **Configure SMTP configurations in Jenkins**

* 1. Sign in to **Jenkins** using your credentials

A screenshot of a computer

Description automatically generated

**Note:** The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

* 1. Navigate to **Manage Jenkins** in the Jenkins dashboard

A screenshot of a computer

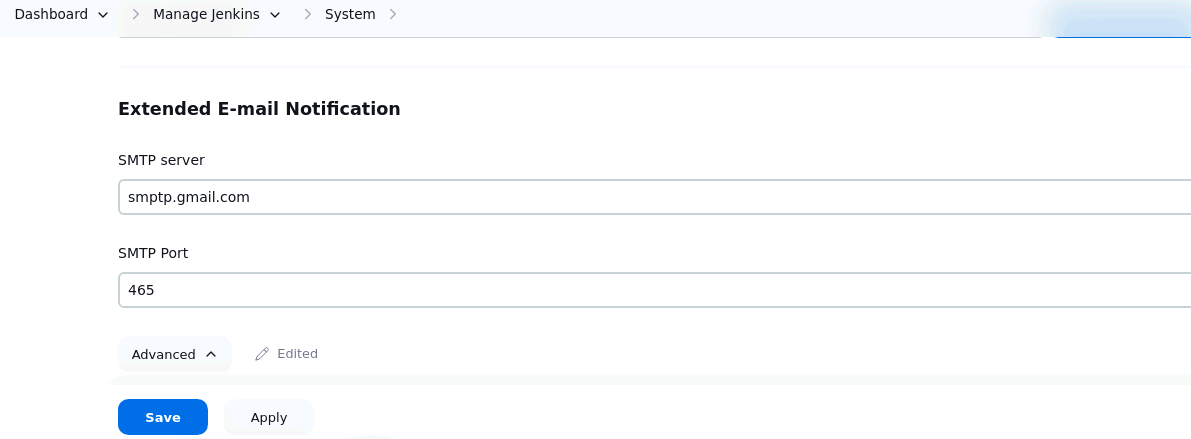
Description automatically generated

* 1. Now, click on **System** under **System Configuration**

A screenshot of a computer

Description automatically generated

* 1. In the **System** section,scroll down to **Extend** **E-mail Notification** section and provide **smtp.gmail.com** as **the SMTP server** and **465** as **SMTP Port**



* 1. Click on **Advanced**, and in the **Credentials,** click on **Add** to add your credentials

**A screenshot of a computer

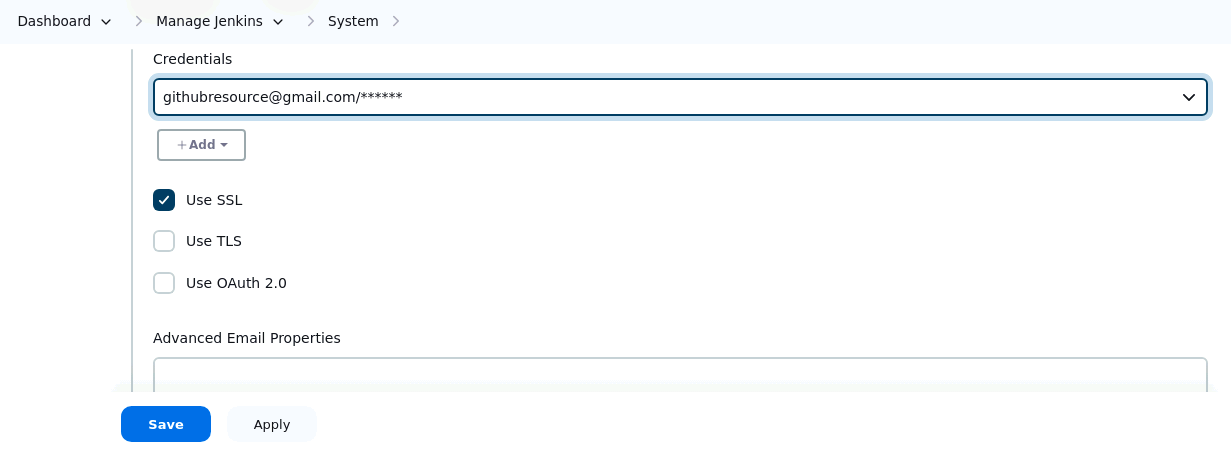
Description automatically generated**

* 1. In the **Username,** provide your email ID; for **Password,** add the previously created app password, and then for **ID**, write **gmail** and click on **Add**

**A screenshot of a computer

Description automatically generated**

* 1. Select **Use SSL** and click on **Save** to save the configurations

****

**Step 3:** **Configure Maven in Jenkins**

* 1. Navigate back to Jenkins **Dashboard** and click on **New Item**



* 1. Enter the name of your project as **Test-lep**, select **Freestyle project**, then click on **OK**

A screenshot of a computer

Description automatically generated

* 1. Navigate to **Source Code Management** and provide the below Git repository under **Repository URL**:

**https://github.com/github-simplilearn-net/MavenBuild.git**

A screenshot of a computer

Description automatically generated

* 1. Navigate to **Build Steps**,click on **Add build step,** and select **Invoke top-level Maven targets**

**A screenshot of a computer

Description automatically generated**

* 1. Write **test** in the **Goals**

**A screenshot of a chat

Description automatically generated**

* 1. Navigate to **Post-build Actions**, click on **Add post-build action**, and select **Editable Email Notification**

**A screenshot of a chat

Description automatically generated**

* 1. Under **Project Recipient List,** write your email address

A screenshot of a chat

Description automatically generated

* 1. Scroll down to **Attachments** and provide the below details:

**target/surefire-reports/\*.xml**

**A screenshot of a computer

Description automatically generated**

* 1. Click on **Advanced Settings**, scroll down to **Triggers,** and remove the existing trigger

**A screenshot of a chat

Description automatically generated**

**A screenshot of a chat

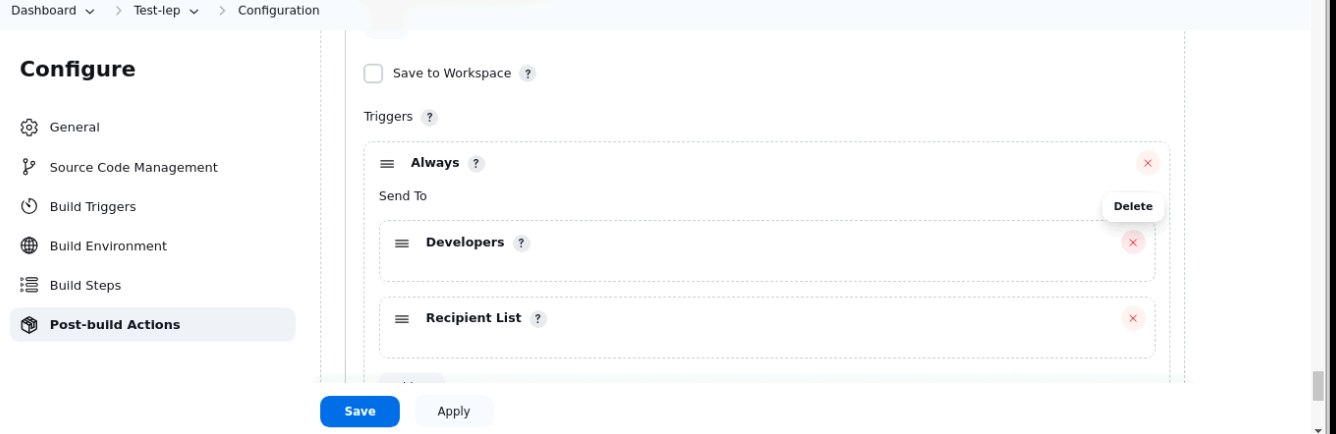
Description automatically generated**

* 1. Clickon **Add Trigger** and select **Always**

**A screenshot of a computer

Description automatically generated**

* 1. Delete the **Developers** and keep the **Recipient** list, then click on **Save**

****

* 1. Now, click on **Build Now** to build the job

**A screenshot of a computer

Description automatically generated**

* 1. Once the job is built, click on **Console Output** to see the output

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A white paper with black text

Description automatically generated**

You can see the build is successful.

* 1. Navigate to your email account to validate if the email has been sent

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



You can see the email has been sent successfully.

A screenshot of a computer

Description automatically generated

You can see the Junit report.

By following these steps, you have successfully configured a Jenkins pipeline to compile code, conduct JUnit tests, and send out test reports via email notifications.