**Lesson 01 Demo 01**

**Implementing the DevOps Model**

**Objective:** To implement DevOps using GitHub to store a Java program and Jenkins to build consistent code packages, enabling continuous integration

**Tools required:** Git, GitHub, and Jenkins

**Prerequisites:** None

Steps to be followed:

1. Create a GitHub repository
2. Add a Java program to the repository
3. Create a freestyle build job in Jenkins
4. Build the Java program with Jenkins

**Step 1: Create a GitHub repository**

1. Open the browser in your lab, go to [**https://github.com**](https://github.com), and click on the **Sign in** button

A screenshot of a computer

Description automatically generated

1. Enter the credentials of your GitHub account and click on **Sign in**

A screenshot of a computer

Description automatically generated

1. Click on **New** as shown in the screenshot below:

A screenshot of a computer

Description automatically generated

1. Add the **Repository name** as shown in the screenshot below:

A screenshot of a computer

Description automatically generated

1. Select the check box of **ADD a README file** and click on **Create repository**

A screenshot of a computer

Description automatically generated

1. Click on **<> Code**, then **HTTPS**, and finally copy the repository URL

A screenshot of a computer

Description automatically generated

**Step 2: Add a Java program to the repository**

1. Open the terminal, run the following commands to create a directory, navigate to the **hello-world** directory, and open the Java file in a text editor as shown in the screenshot below:

**mkdir hello-world**

**cd hello-world**

**nano HelloWorld.java**

A screenshot of a computer

Description automatically generated

1. Copy and paste the below code into the file, save the file, and exit from the text editor:

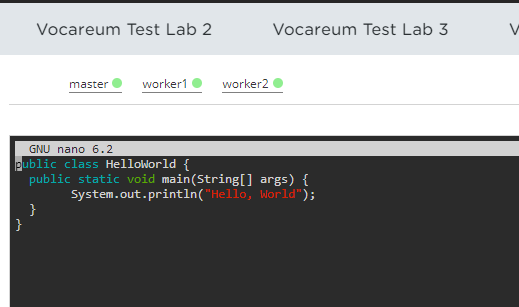
**public class HelloWorld {**

**public static void main(String[] args) {**

**System.out.println("Hello, World");**

**}**

**}**



1. Run the following commands:

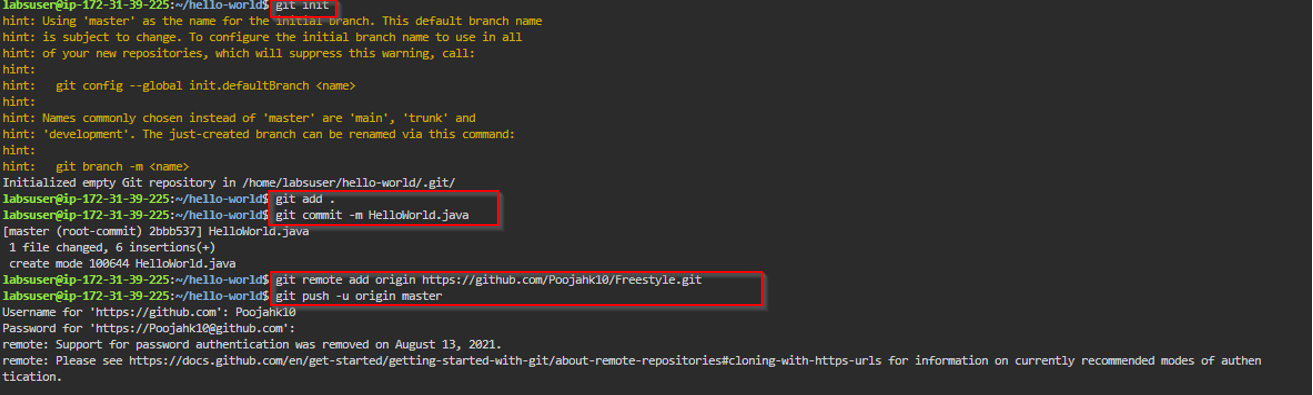
**git init**

**git add .**

**git commit -m “Add new files”**

**git remote add origin <Repository\_URL>**

**git push -u origin master**



|  |
| --- |
| **Note:** Ensure that the password to be added is your **GitHub** account **Token** |

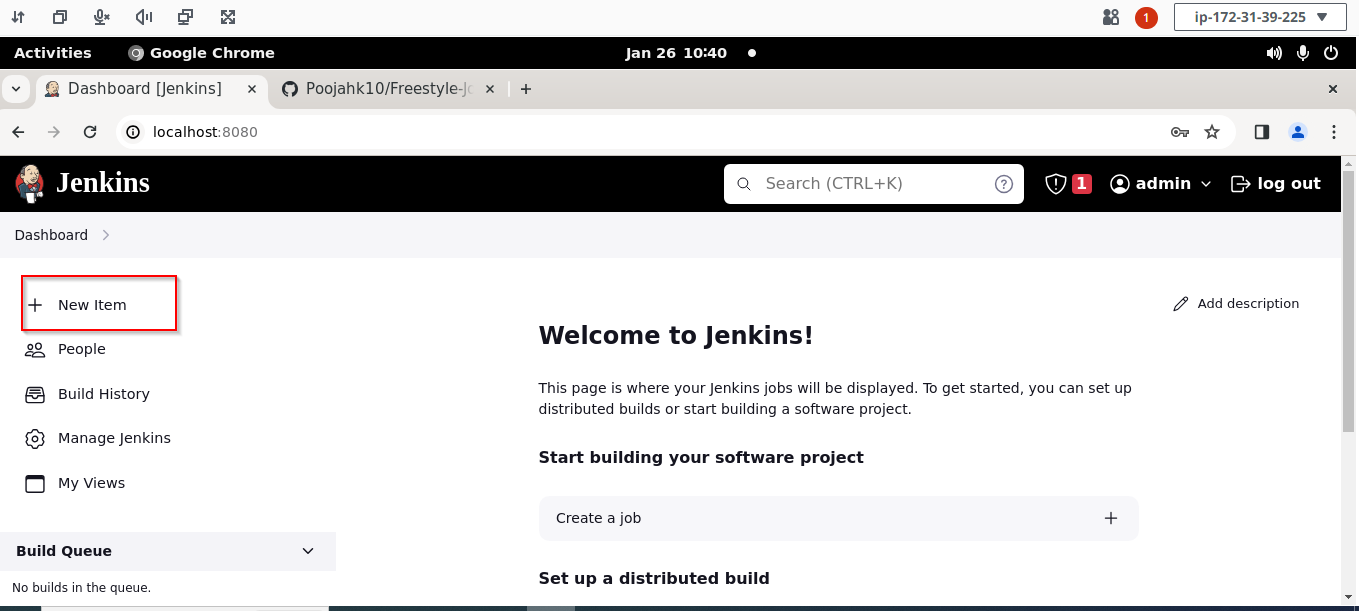
**Step 3: Create a freestyle build job in Jenkins**

1. Open the browser, type **localhost:8080**; this will open Jenkins. Provide the credentials and then click on **Sign in**

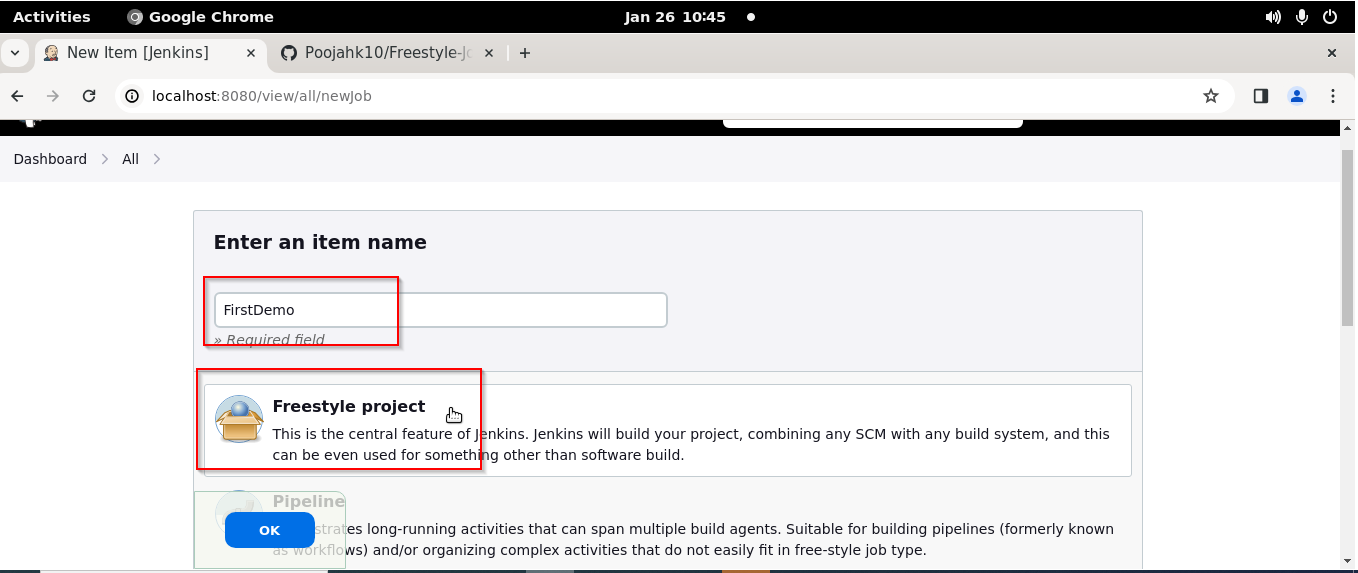
A screenshot of a login page

Description automatically generated

1. Click on **New Item** in the Jenkins Dashboard



1. Enter a name for your project, select **Freestyle project** as the build job type, and click on **OK**



1. Click on **Source Code Management**

A screenshot of a computer

Description automatically generated

1. Select **Git**, enter the **Repository URL**,and then click on **Save**

A screenshot of a computer

Description automatically generated

**Step 4: Build the Java program with Jenkins**

1. Click on **Build Now** to build your project

A screenshot of a computer

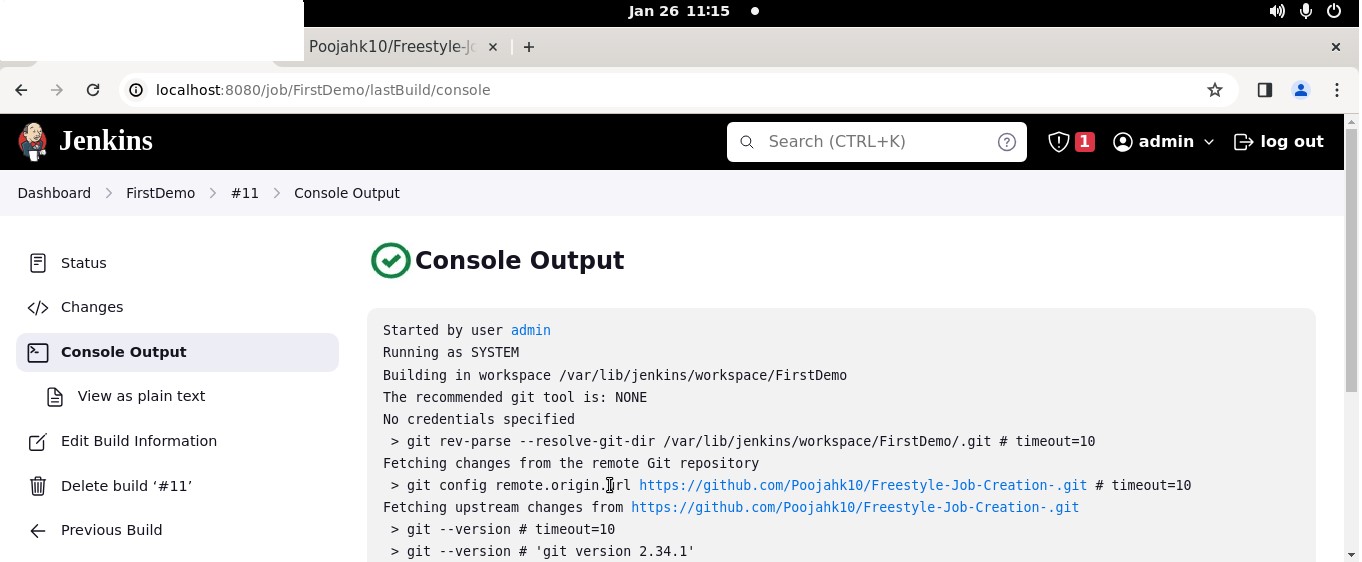
Description automatically generated

1. Click on **Build History** to view the build results

A screenshot of a computer

Description automatically generated

1. Click on the **Console Output** to view the build logs



By following these steps, you have successfully implemented DevOps using GitHub to store a Java program and Jenkins to build consistent code packages, enabling continuous integration.

.