**Setting up Jenkins Pipeline Job from Git**

**Objective:** To set up a Jenkins pipeline job from Git version control system to enable automated CI for building, testing, and potentially deploying software upon code changes

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

**Prerequisites:** None

Steps to be followed:

1. Create a Git repository
2. Push the pipeline script into the Git repository
3. Create a pipeline script-based freestyle job

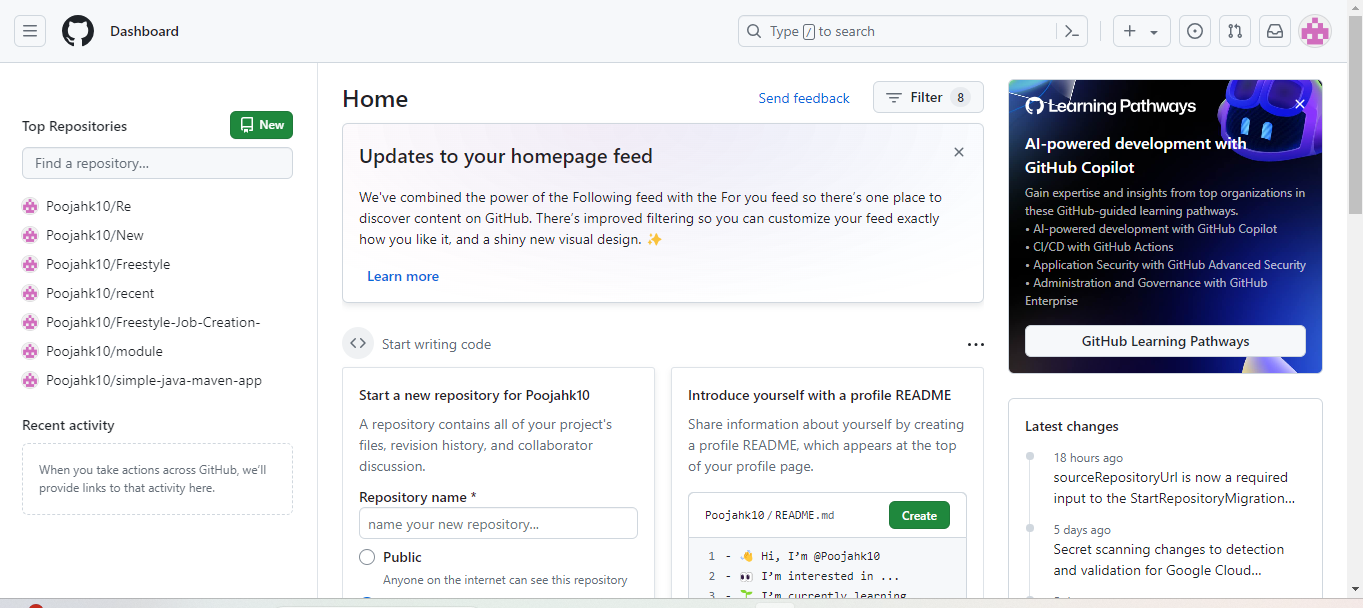
**Step 1: Create a Git repository**

|  |
| --- |
| **Note**: If you do not have a GitHub account, visit the official website at **https://github.com/signup** and create a new account |

1. Open the browser in your lab, go to **github.com**, and **Sign in** to your account  
     
   A screenshot of a login form

   Description automatically generated

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

****

1. Enter a desired name for your repository and choose **Public** as shown in the screenshot below:

A screenshot of a computer

Description automatically generated

1. Click on **Add a README** **file** and then click on **Create repository** as shown in the screenshot below:

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. Copy the repository URL as shown in the screenshot below:

A screenshot of a computer

Description automatically generated

**Step 2: Push the pipeline script into the Git repository**

1. Open the Linux terminal in your lab and clone the repository using the below command:  
   **git clone RepositoryURL**

A computer code with black text

Description automatically generated

1. Navigate inside the repository that you had created using the below command:  
   **cd RepositoryName**
2. Initialize the Git using the below command:  
   **git init**



1. Create a file using the below command:  
   **nano demofile**



1. Paste the below pipeline script inside the file as shown in the screenshot below:

**pipeline {**

**agent any**

**stages {**

**stage('Checkout') {**

**steps {**

**// Checkout your source code from version control**

**git 'https://github.com/your/repository.git'**

**}**

**}**

**stage('Build') {**

**steps {**

**// Use Maven to build your project**

**sh 'mvn clean package'**

**}**

**}**

**stage('Test') {**

**steps {**

**// Run tests if applicable**

**sh 'mvn test'**

**}**

**}**

**stage('Deploy') {**

**steps {**

**// Deploy your artifact, if necessary**

**// Example: sh 'mvn deploy'**

**}**

**}**

**}**

**post {**

**success {**

**// This block will be executed if the pipeline runs successfully**

**echo 'Pipeline executed successfully!'**

**}**

**failure {**

**// This block will be executed if the pipeline fails**

**echo 'Pipeline failed!'**

**}**

**}**

|  |
| --- |
| **Note**: Ensure you provide your Git repository URL on line7, save, and exit the page by clicking on **ctrl+S** to save and **ctrl+X** to exit |

**}**

A screenshot of a computer

Description automatically generated

1. Stage and commit the changes using the below commands:  
   **git add .**

**git commit -m “initial commit”**  
  
A screenshot of a computer

Description automatically generated

1. Push the file to the Git repository using the below command:

**git push**  
A screenshot of a computer code

Description automatically generated

1. Navigate to your Git repository to check for the file that is pushed as shown in the screenshot below:  
     
   A screenshot of a computer

   Description automatically generated

**Step 3: Create a pipeline script-based freestyle job**

|  |
| --- |
| **Note**: Use the given credentials to access Jenkins in the lab: **Username** is admin and **Password** is admin |

1. Open the browser, go to the Jenkins **Dashboard** by typing **localhost:8080** in your browser, provide the credentials, and click the **Sign in** button   
     
   A screenshot of a login page

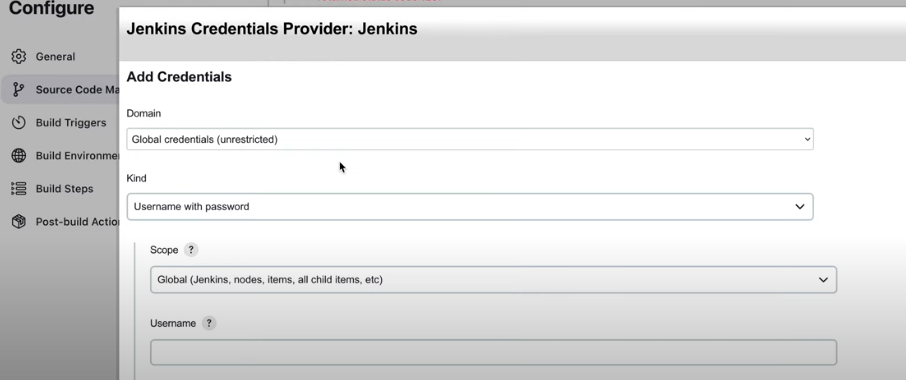
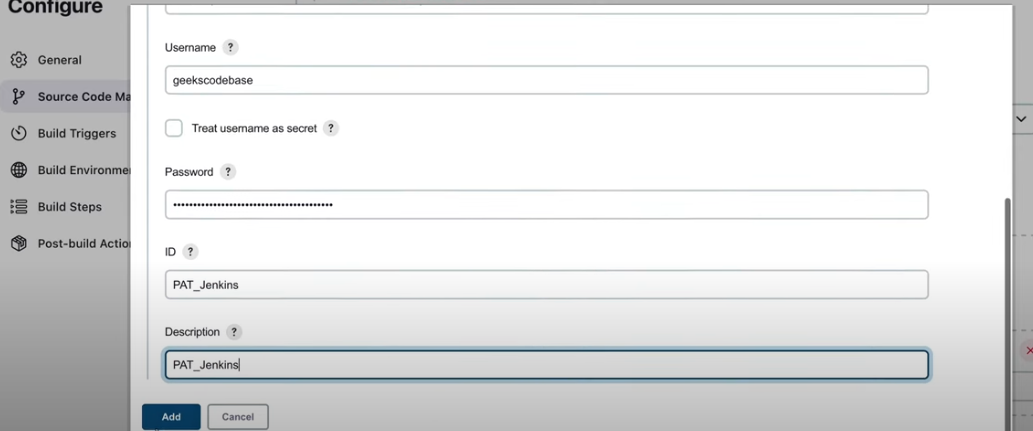
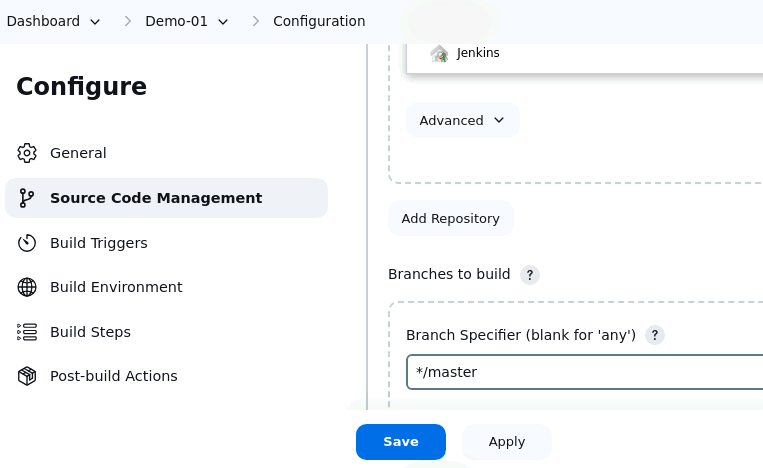
   Description automatically generated
2. Click on **New Item** as show in the screenshot below:  
     
   A screenshot of a computer

   Description automatically generated
3. Enter a desired name for the project, select **Freestyle project**, and then click on **OK** as shown in the screenshot below: A screenshot of a computer

   Description automatically generated
4. Click on **Source Code Management** asshown in the screenshot below:A screenshot of a computer

   Description automatically generated
5. Click on **Git** as shown in the screenshot below:  
     
   A screenshot of a computer

   Description automatically generated
6. Enter the repository URL and click on **Add** as shown in the screenshot below:  
   A screenshot of a computer

   Description automatically generated
7. Enter a desired name for the **Username** as shown in the screenshot below:  
   
8. Paste the Git token under **Password** section and then click on **Add** as shown in the screenshot below**:**
9. Click on **Save** as shown in the screenshot below:  
   
10. Click on **Build Now** in the left section after the creation of the job as shown in the below screenshot:  
      
    A screenshot of a computer

    Description automatically generated

1. Click on **Build History** as shown in the screenshot below:  
   A screenshot of a computer

   Description automatically generated
2. Verify that the output indicates the status (successful) as shown in the screenshot below:  
   A screenshot of a computer

   Description automatically generated  
   By following these steps, you have successfully set up the Jenkins freestyle job from the Git version control system to enable automated CI for building, testing, and potentially deploying software upon code changes.