Jenkins Automation with Gitlab

1.Update the pakages

```
apt-get update
```

2.Install Required Java

Jenkins is a Java-based application, so you need to install Java on your system. You can install OpenJDK using the following command:

```
apt-get install openjdk-11-jdk -y
```

Verify the installation by checking the Java version

```
java -version
```

3.Add Jenkins Repository and Key

```
https://www.jenkins.io/doc/book/installing/linux/#debianubuntu
```

Go to jenkins website in browser, --> "Linux Section" --> "Debian/Ubuntu" --> " Copy the Long Term Support release" --> Do that steps

```
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
```

4.Start the Jenkins service and enable it to start on boot

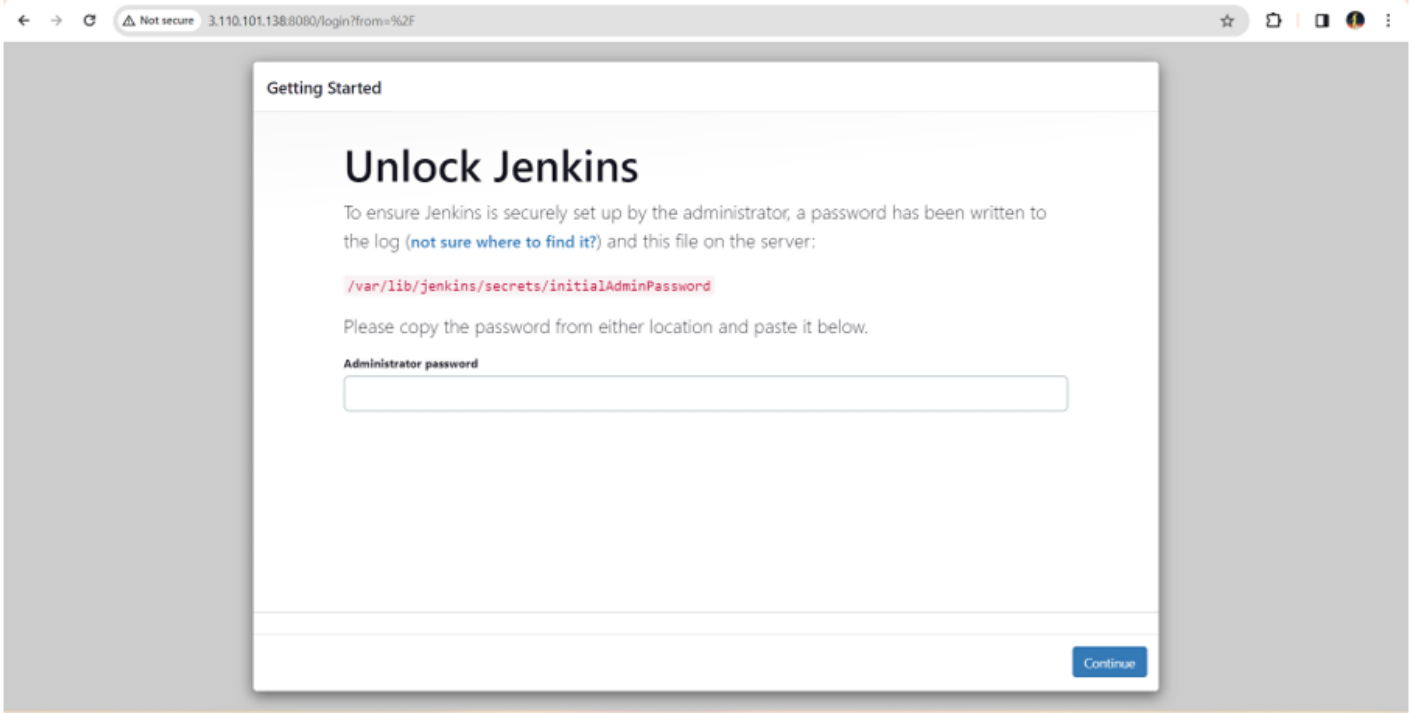
```
sudo systemctl start jenkins
```

```
sudo systemctl enable jenkins
```

5.Access Jenkins Web Interface

By default, Jenkins runs on port 8080. Open your web browser and navigate to

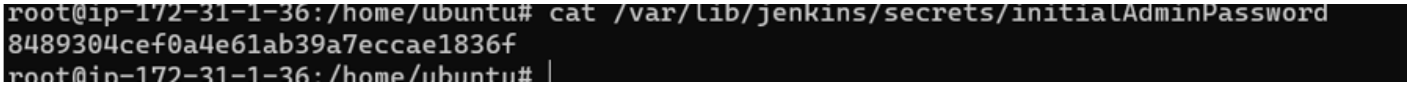
```
http://your_server_ip_or_domain:8080
```



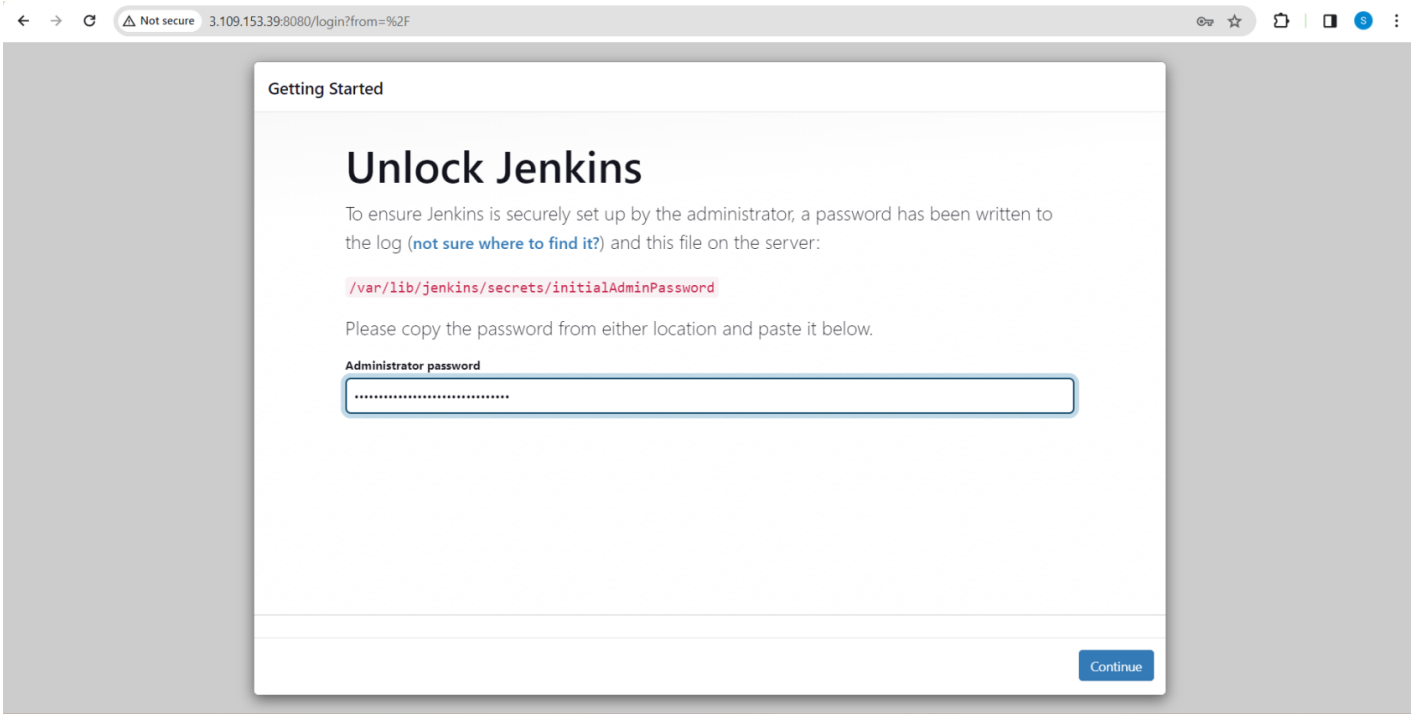
6.You will be prompted to unlock Jenkins.

Retrieve the initial admin password from the following location:

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



Copy and paste this password into the Jenkins web interface to unlock it.



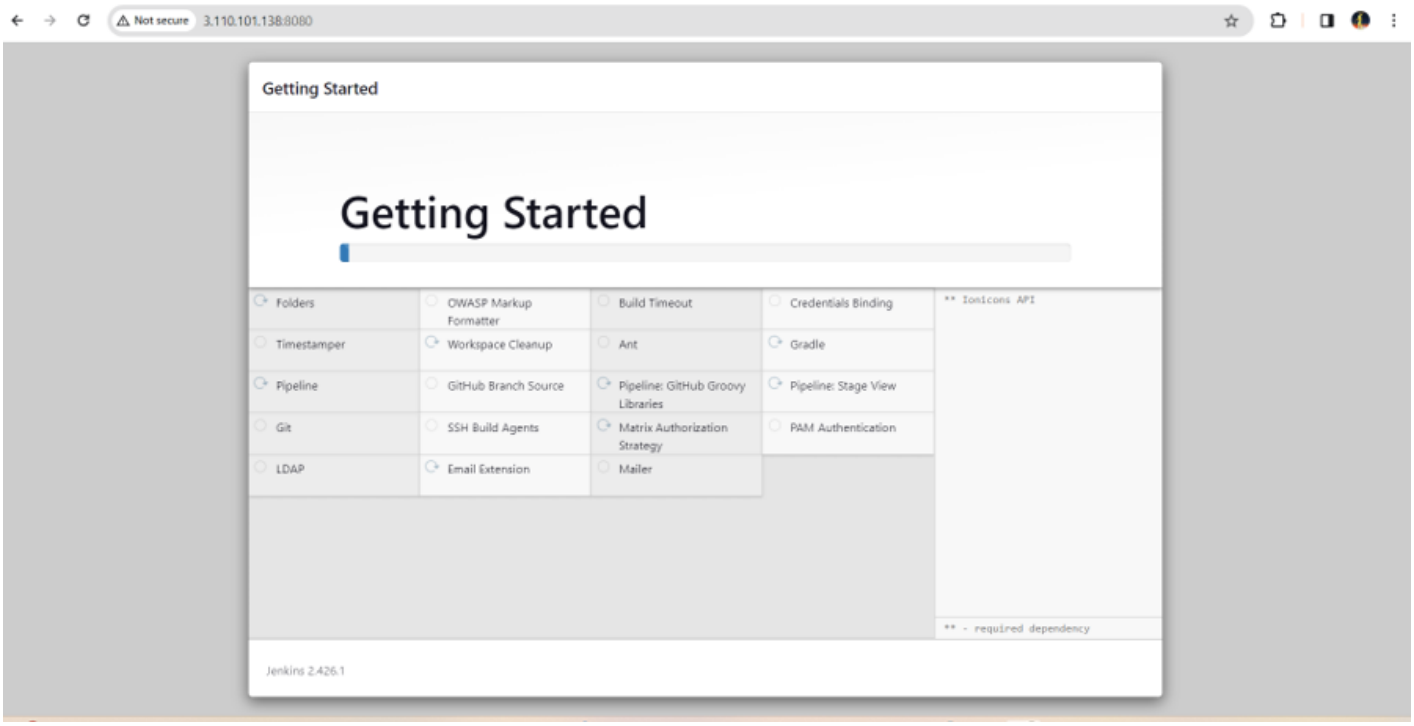
7.Install Plugins

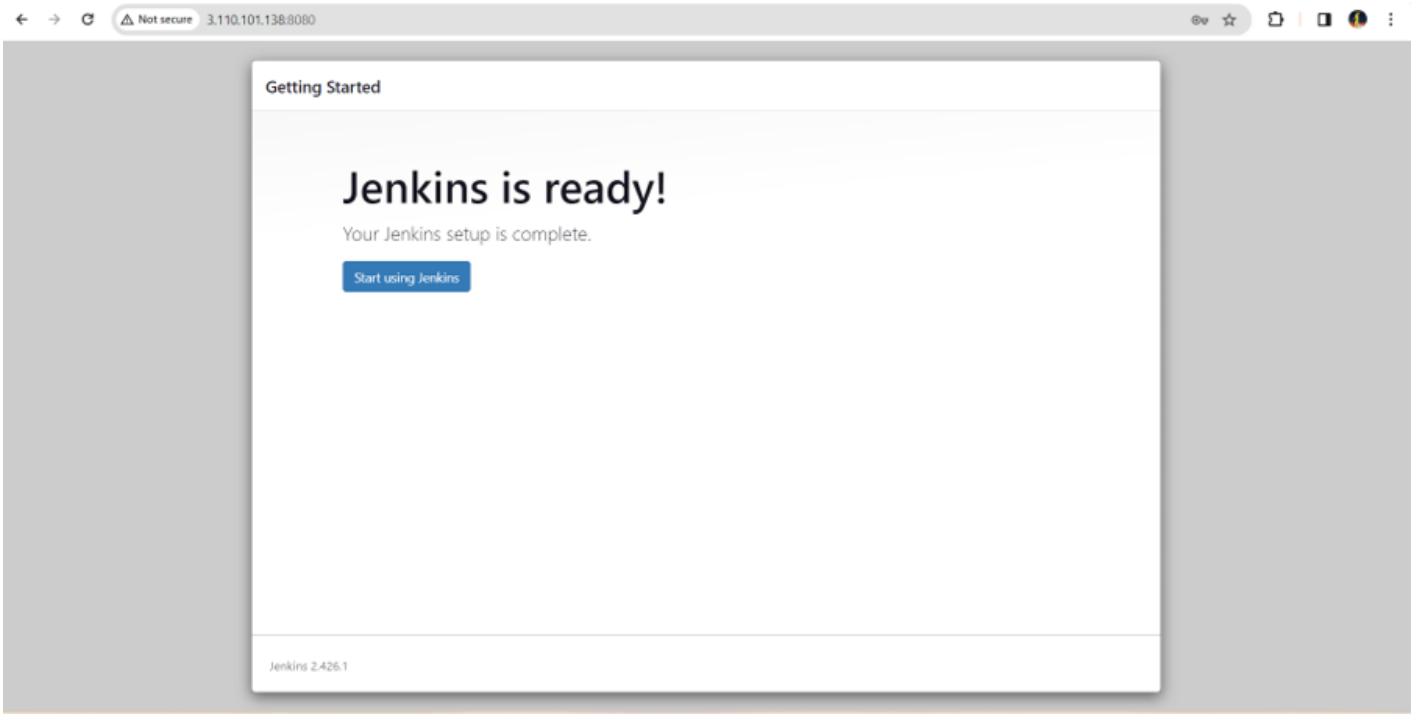
After unlocking Jenkins, you can choose to install recommended plugins or select specific plugins based on your requirements.

-- Create an admin user and provide the necessary details.

-- Specify the Jenkins URL and click "Save and Finish."

-- After completing the setup, you can start using Jenkins.



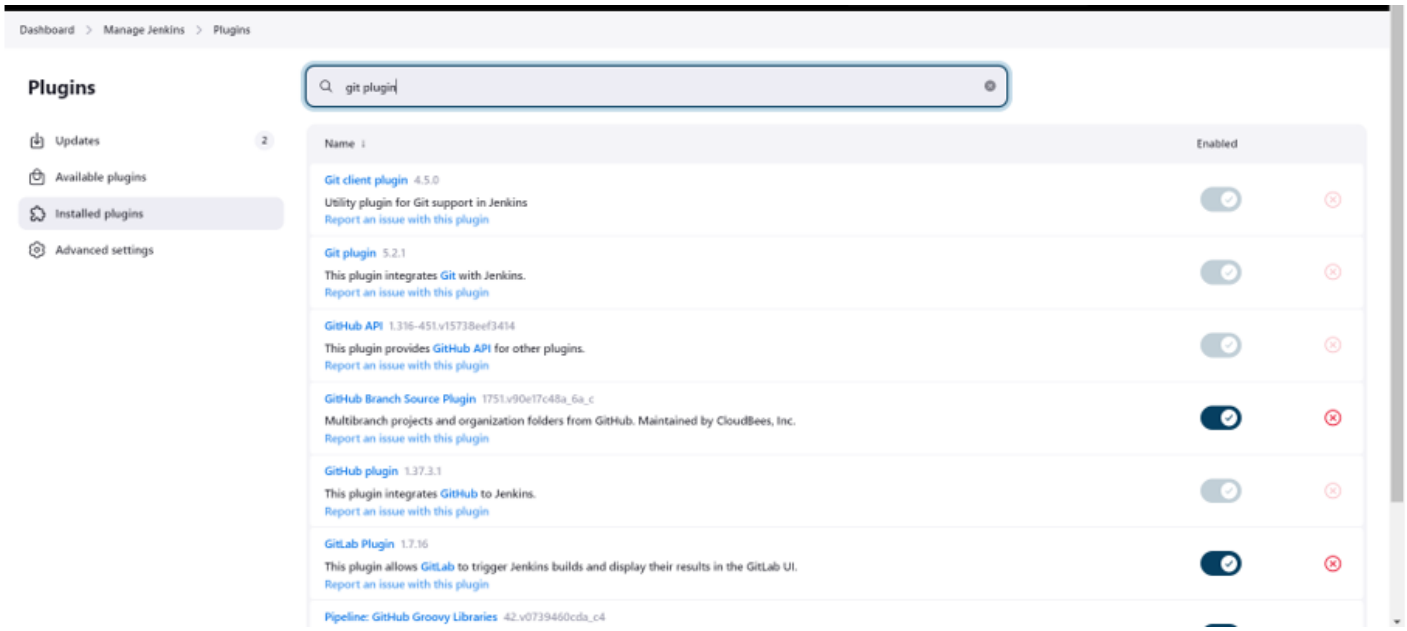


Configure the Jenkins server

Install and configure the Jenkins plugin to authorize the connection to GitLab.

1.On the Jenkins server, select **"Manage Jenkins"** -> **"Manage Plugins".**

2.Select the Available tab. Search for `gitlab-plugin` and select it to install.



3.Select **Manage Jenkins** -> **Configure System.**

4.In the GitLab section, select **Enable authentication for '/project' end-point.**

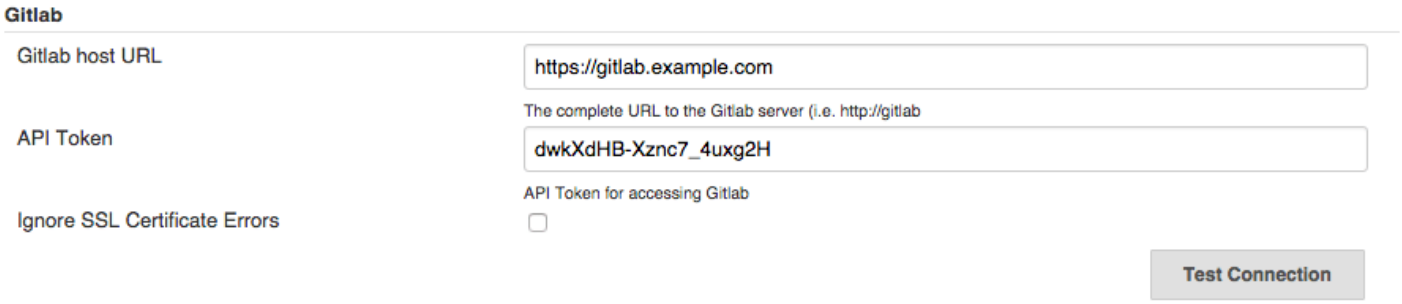
5.Select **Add**, then choose **Jenkins Credential Provider.**

6.Select **GitLab API token** as the token type.

7.In **API Token**, paste the access token value you copied from GitLab and select Add.

8.Enter the GitLab server's URL in **GitLab host URL.**

9.To test the connection, select Test Connection



Configure the Jenkins project

Set up the Jenkins project you intend to run your build on.

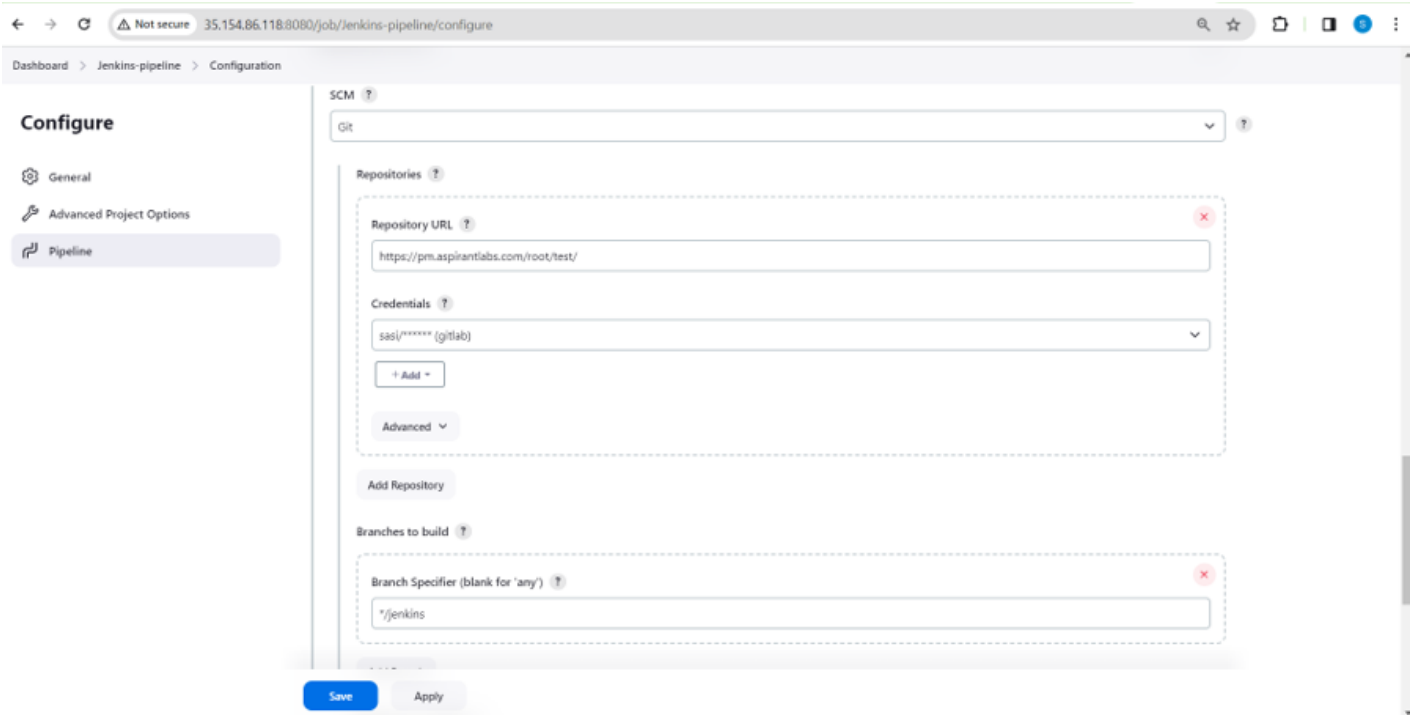
1.On your Jenkins instance, select New Item.

2.Enter the project's name.

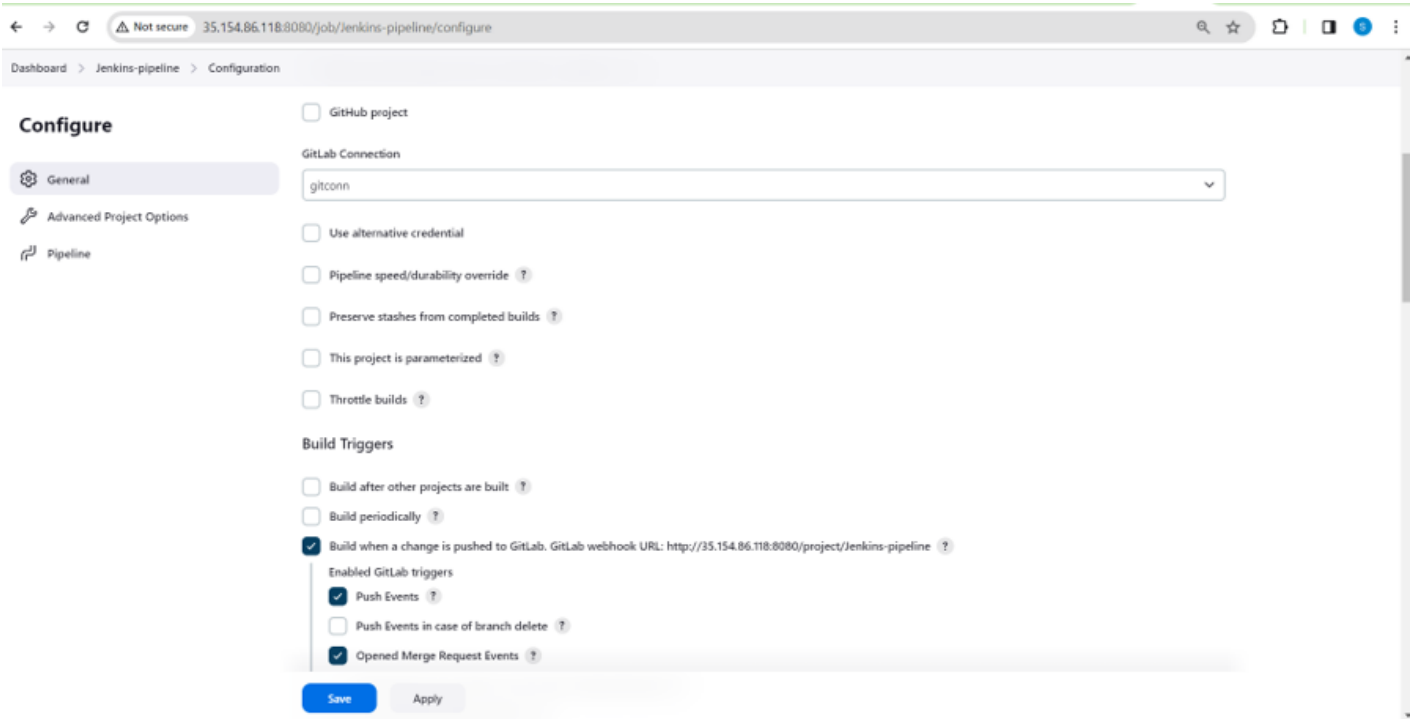
3.Select Freestyle or Pipeline and select OK. You should select a freestyle project, because the Jenkins plugin updates the build status on GitLab. In a pipeline project, you must configure a script

to update the status on GitLab.

4.Choose your GitLab connection from the dropdown list.



5.Select Build when a change is pushed to GitLab.



6.Select the following checkboxes:

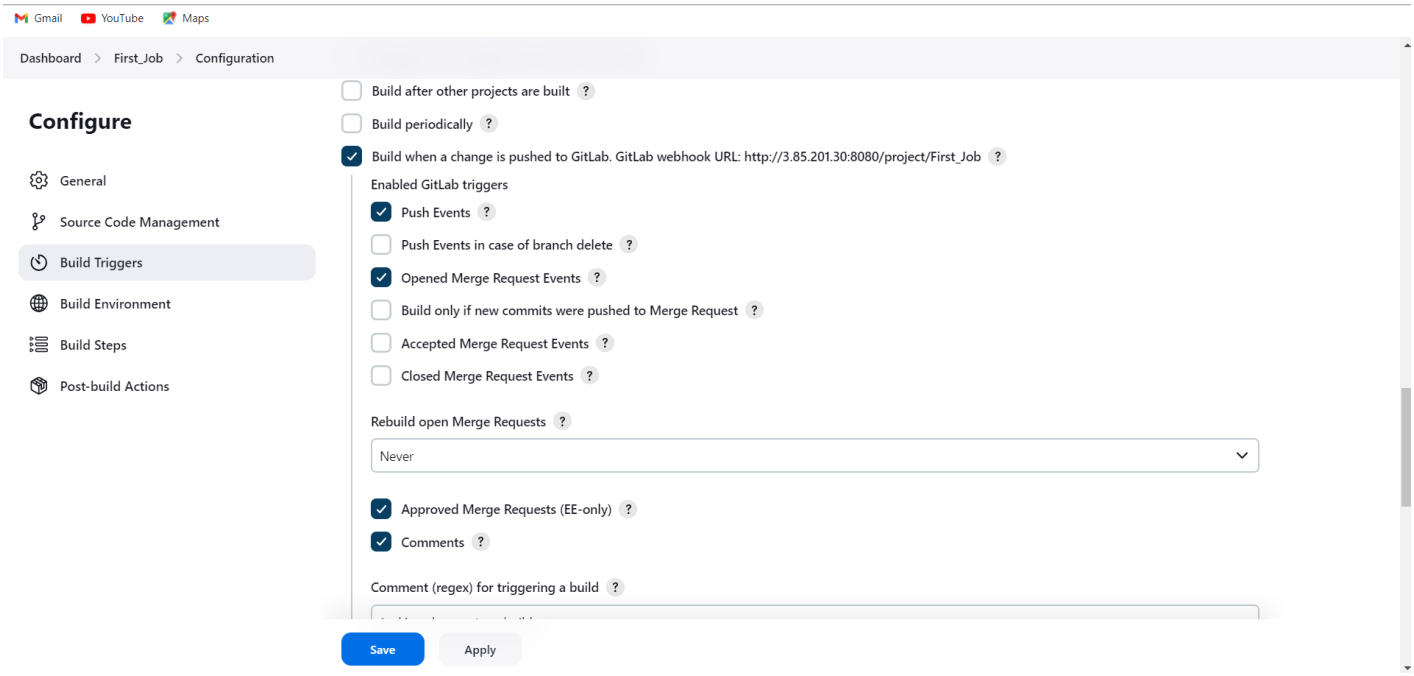
Accepted Merge Request Events

Closed Merge Request Events

7.Specify how the build status is reported to GitLab:

*If you created a freestyle project, in the **Post-build Actions section**, choose **Publish build status to GitLab**.*

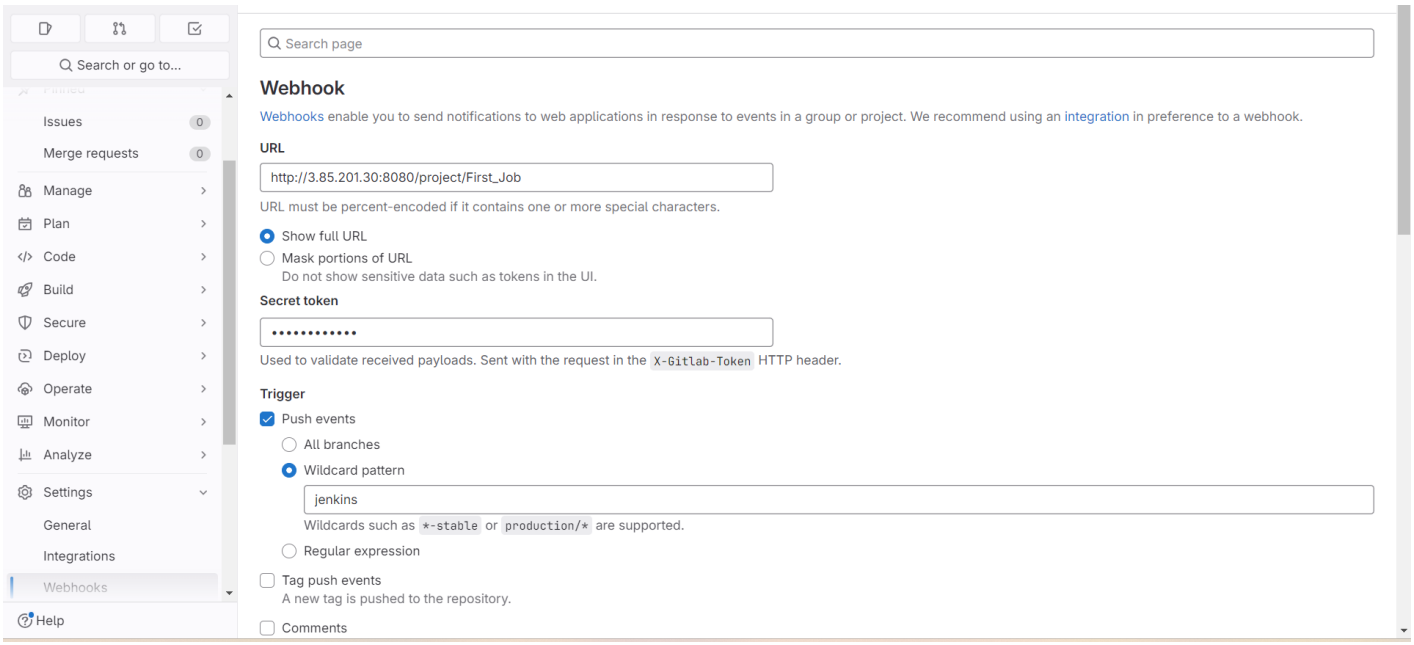
If you created a pipeline project, you must use a Jenkins Pipeline script to update the status on GitLab.



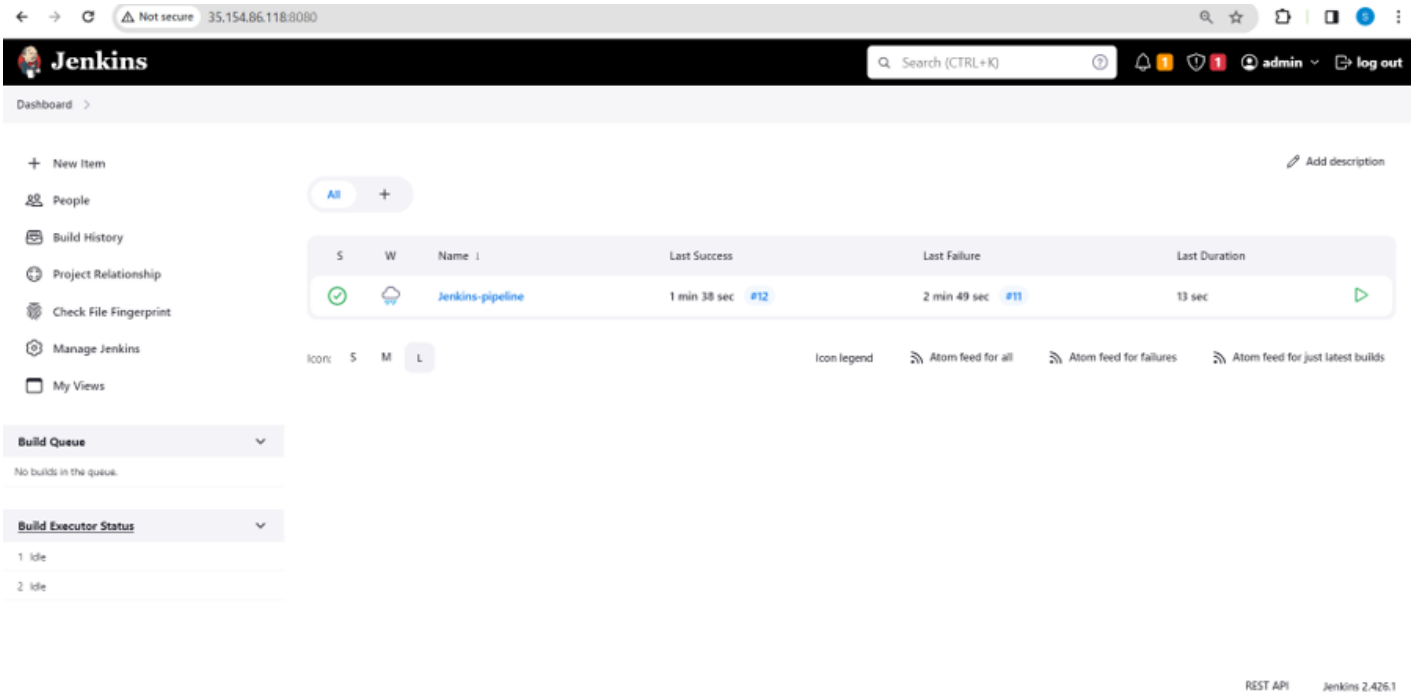
8.Generate a token

copy and save the job

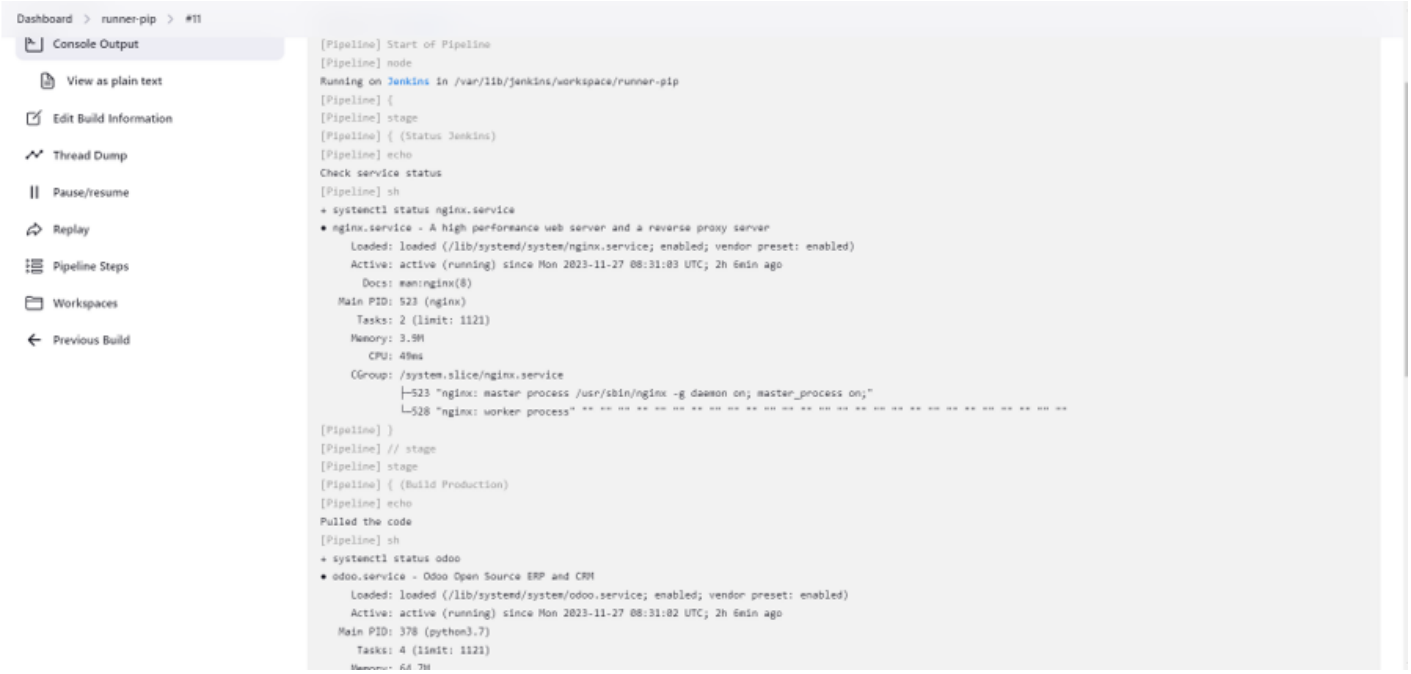
9.Go to webhook paste the url and secret token



10.Now make the changes in gitlab and check the job will or not



11.If the job will trigger , check the console output



Revision #10

Created Fri, Jan 5, 2024 4:57 AM by sasi

Updated Wed, Apr 3, 2024 7:44 AM by sasi