

Demo Project: Configure Webhook to trigger CI Pipeline automatically on every change

This guide walks you through setting up webhooks to enable automatic triggering of Jenkins jobs for both Pipeline and Multibranch Pipeline projects when changes are pushed to a GitLab repository.

Why Use Webhooks?

Configure Webhook for Pipeline Job

Step 1: Configure Jenkins to Authenticate with GitLab API

Step 2: Configure GitLab Webhook

Configure Webhook for Multibranch Pipeline

Step 1: Enable Multibranch Webhook Plugin

Step 2: Add Webhook in GitLab

Testing the Setup

Pipeline Job:

Multibranch Pipeline:

Why Use Webhooks?

Webhooks allow GitLab to notify Jenkins of repository events (e.g., pushes, merges). Jenkins then uses this notification to trigger builds automatically, removing the need for manual intervention.

Configure Webhook for Pipeline Job

Step 1: Configure Jenkins to Authenticate with GitLab API

1. Install the GitLab Plugin:

- Navigate to **Manage Jenkins → Plugins**.
- Search for and install the **GitLab** plugin.

2. Add GitLab Connection in Jenkins:

- Go to **Manage Jenkins → System**.
- Scroll down to the **GitLab** section and:
 - Add a connection name, e.g., `gitlab-conn`.
 - Set the GitLab host URL to `https://gitlab.com/`.
 - Click `+ ADD` to create an **API Token**.

3. Create a GitLab API Token:

- In GitLab:
 - Go to **User Profile → Preferences → Access Tokens**.
 - Add a token name, set an expiration date, and choose the `api` scope.
- Copy the token and paste it in Jenkins under:
 - **Kind:** GitLab API token
 - **API token:** GitLab API token.
 - **ID and Description:** `gitlab-token`.

4. Save the GitLab Connection:

- Select the credentials created above and save the configuration.

5. Set Build trigger in pipeline:

- Choose the pipeline to intergrate
- Go to **Configuration**.
- Scroll down to the **Build Triggers** section and:
 - Click `Build when a change is pushed to GitLab`.
 - Also click on `Push Events`, as well as `Opened Merge Request Events`.

Build Triggers

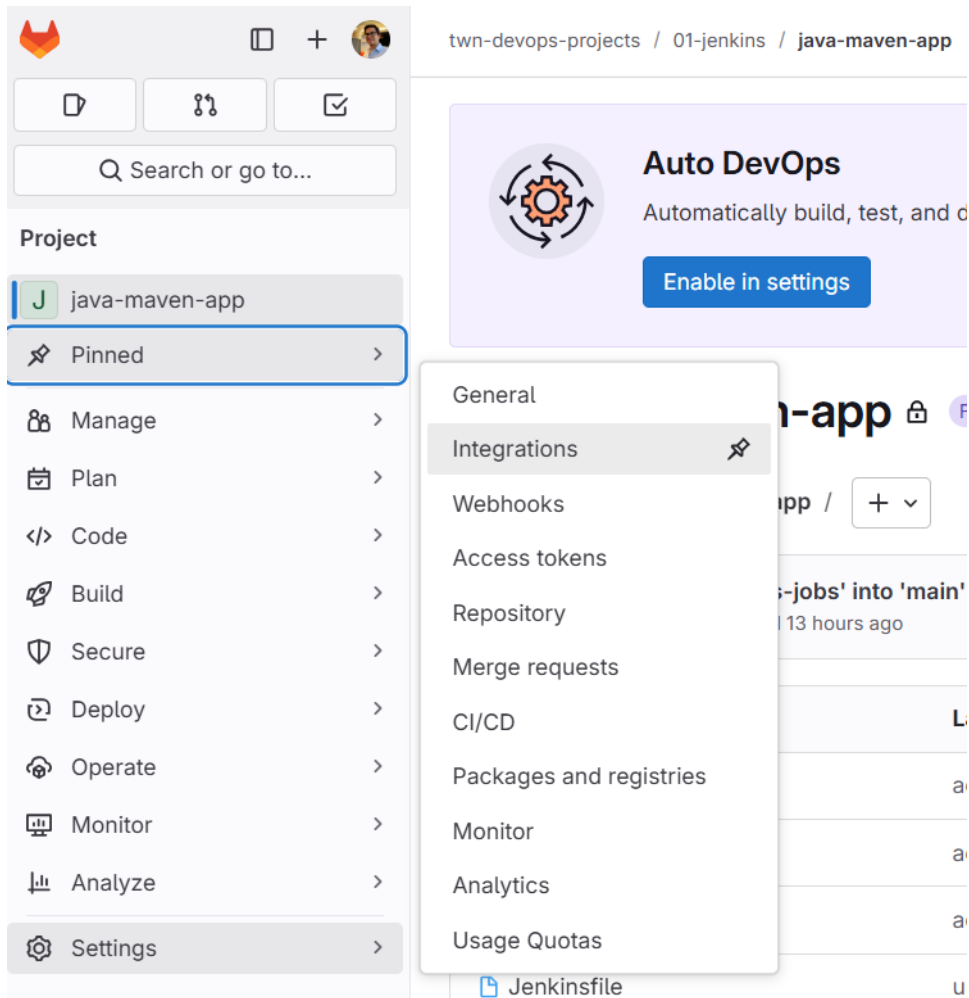
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ Build when a change is pushed to GitLab. GitLab webhook URL: http://159.223.150.154:8080/project/my-pipeline

Enabled GitLab triggers

- ☒ Push Events ?
- ☐ Push Events in case of branch delete ?
- ☒ Opened Merge Request Events ?
- ☐ Build only if new commits were pushed to Merge Request ?
- ☐ Accepted Merge Request Events ?
- ☐ Closed Merge Request Events ?

Step 2: Configure GitLab Webhook

1. In GitLab, go to the project and click **Settings → Integrations**.



2. Scroll down to **Jenkins** then and click **Configure**.

3. Add a new **Webhook**:

- Jenkins server URL: `<JENKINS_URL>/git/notifyCommit?url=<YOUR_REPO_URL>`.
- Example: `http://159.223.150.154:8080/git/notifyCommit?url=https://gitlab.com/my-repo.git`.
- Disable **SSL verification** if needed.

4. Add the **Project Name**

- Example: `my-pipeline`.

5. Add your **Username**, and **Password** for Jenkins.

6. Save the webhook and click **Test Settings** to verify connectivity.

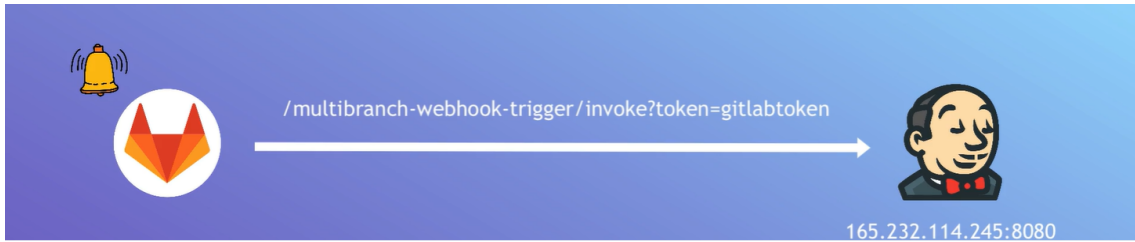
Configure Webhook for Multibranch Pipeline

Step 1: Enable Multibranch Webhook Plugin

1. Install the **Multibranch Scan Webhook** plugin in Jenkins:
 - Navigate to **Manage Jenkins → Plugins**.
 - Search for and install **Multibranch Scan Webhook**.
 2. Configure Multibranch Pipeline:
 - Go to your multibranch pipeline job.
 - Under **Configure → Scan Multibranch Pipeline Triggers**, enable **Scan by webhook**.
 - Add a Trigger Token, e.g., `gitlabtoken`.
 - Take note of the Trigger token since its needed to set the webhook in GitLab.
 3. Save the configuration.
-

Step 2: Add Webhook in GitLab

1. In GitLab, go to **Settings → Webhooks**.
2. Add a new webhook with the following:
 - URL: `<JENKINS_URL>/multibranch-webhook-trigger/invoke?token=<TRIGGER_TOKEN>`.
 - Example: `http://159.223.150.154:8080/multibranch-webhook-trigger/invoke?token=gitlabtoken`.
 - Enable **Push Events** as the trigger.
 - Note: This URL is where Jenkins is accepting calls from GitLab. Therefore GitLab will send a notification to this URL, and will send the token indicated in the request.



3. Save the webhook.
-

Testing the Setup

Pipeline Job:

1. Push code to the repository.
2. Check Jenkins for a triggered build under the Pipeline job.
3. Verify the build log for success.

Multibranch Pipeline:

1. Push changes to a branch in the repository.
 2. Jenkins should automatically scan the branch and trigger a build.
 3. Review the console output for any issues.
-