

Multibranch Pipeline with a Webhook on Jenkins

Sometimes it may be necessary to create a pipeline on Jenkins for each Git branch. In this case it could be difficult to create independent pipeline for each branch. Besides that what if we create or delete a branch in future? So someone has to take care of the pipelines on Jenkins whenever there is change in branches. that's where Multibranch pipeline comes into picture.

Through Multibranch pipeline we can create a pipeline for each branch in a repository and it also create or remove when there is a change in the branches. Lets see how it works !!

Pre-Requisites:

1. Expected a Jenkins server is already up and running
2. A Git repository with more than one branch

In my case I am using a repository called <https://github.com/ravdy/test-nodejs-app.git>

Create a Multibranch Pipeline

1. Login to Jenkins GUI
2. Click on “New Item” → Specify a job name → Select “Multibranch Pipeline option”

Enter an item name

* Required field

- Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

If you want to create a new item from other existing, you can use this option:

Copy from

OK

3. Give Display Name and Description

4. Under Branch Sources → Add source → Chose Git → and provide GitHub URL and Credentials (Credentials are optional if it is public repo)

Branch Sources

Add source

Git

Single repository & branch

Branch Source Image

Branch Sources

Git

Project Repository

Credentials **Add**

Behaviors

Discover branches

Add

Property strategy

Add property

Add source

Git Repo Information

5. Under Build Configuration → chose Jenkinsfile path. Most of the case it will be under Repo root directory.

A screenshot of the Jenkins 'Build Configuration' page. It shows a 'Mode' dropdown menu set to 'by Jenkinsfile' and a 'Script Path' text input field containing 'Jenkinsfile'. There is a blue help icon to the right of the script path field.

Build Configuration

Mode: by Jenkinsfile

Script Path: Jenkinsfile

Jenkinsfile path

6. Apply and Save the job

Now Jenkins automatically scans the repository and create a job for each branch wherever it finds a Jenkinsfile and initiate first build.

A screenshot of the Jenkins 'Multibranch-pipeline-example' page. It shows a table of branches with columns for status, warnings, name, last success, last failure, and last duration. There are three branches: dev, main, and master. A 'Disable Multibranch Pipeline' button is in the top right. A legend at the bottom explains the status icons.

Multibranch-pipeline-example

Disable Multibranch Pipeline

Branches (3)

S	W	Name	Last Success	Last Failure	Last Duration
		dev	3 min 19 sec - #1	N/A	21 sec
		main	3 min 19 sec - #1	N/A	21 sec
		master	3 min 19 sec - #1	N/A	27 sec

Icon: S M L

Legend: Atom feed for all Atom feed for failures Atom feed for just latest builds

Multibranch Pipeline

Using Webhook

If you wish to automate the build process in the multibranch pipeline we can use Webhook. This feature is not enabled until we install “Multibranch Scan Webhook Trigger”. This enables an option “scan by webhook” under “Scan Multibranch Pipeline Triggers”. Here we should give a token. I am giving it as “mytoken”. by this time your job looks something like below.

Branch Sources

Git

Project Repository:

Credentials:

Behaviors:

Property strategy:

Build Configuration

Mode:

Script Path:

Scan Multibranch Pipeline Triggers

☐ Periodically if not otherwise run

☒ Scan by webhook

Trigger token:

The token to match with webhook token. Receive any HTTP request `JENKINS_URL/multibranch-webhook-trigger/invoke?token=[Trigger token]` if a token match, then a multibranch scan will be triggered. (from [Multibranch Scan Webhook Trigger](#))

Jenkins Multibranch pipeline with webhook

Now to enable auto build process we should provide Jenkins URL with token in the GitHub. in this case like should be <http://65.0.130.108:8080/multibranch-webhook-trigger/invoke?token=mytoken>

for this log into GitHub → settings → Webhooks → Add webhook

[ravdy / test-nodejs-app](#)

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights **Settings**

Options Manage access Security & analysis Branches **Webhooks**

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

Provide Payload URL as “<http://65.0.130.108:8080/multibranch-webhook-trigger/invoke?token=mytoken>” and Content type as “application/json” and click on Add webhook

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Actions

Environments

Secrets

Moderation settings

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

http://65.0.130.108:8080/multibranch-webhook-trigger/invoke?token=...

Content type

application/json

Secret

Which events would you like to trigger this webhook?

☒ Just the push event.

☐ Send me everything.

☐ Let me select individual events.

☒ Active

We will deliver event details when this hook is triggered.

Add webhook

Once this is done you can see a new webhook and its time to do some changes in the repository to test the webhook connection.

Webhooks

[Add webhook](#)

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

● [http://65.0.130.108:8080/multibr...](#) (push)

[Edit](#)[Delete](#)

In this demonstration, I am going to create a new branch called “stage” and push the changes onto remote repo.


```









valaxy (main) nodejs
$ ls
Jenkinsfile package.json README.md server.js
valaxy (main) nodejs
$ git checkout -b stage
Switched to a new branch 'stage'
valaxy (stage) nodejs
$ git status
On branch stage
nothing to commit, working tree clean
valaxy (stage) nodejs
$ git push origin stage
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote: This repository moved. Please use the new location:
remote:   https://github.com/ravdy/test-nodejs-app.git
remote:
remote: Create a pull request for 'stage' on GitHub by visiting:
remote:   https://github.com/ravdy/test-nodejs-app/pull/new/stage
remote:
To https://github.com/ravdy/nodejs.git
 * [new branch]      stage -> stage
valaxy (stage) nodejs
$ |





```




New branch creation on Git and pushed changes onto GitHub

Now you could see a build has been triggered automatically on Jenkins and it scan and create a job for the new branch as well.

 **multibranch-pipeline-example** Disable Multibranch Pipeline

Branches (4)					
S	W	Name ↕	Last Success	Last Failure	Last Duration
		dev	32 sec - #1	N/A	21 sec
		main	32 sec - #1	N/A	13 sec
		master	32 sec - #1	N/A	13 sec
		stage	32 sec - #1	N/A	21 sec

Icons:    

Legend:  Always build for all  Always build for failures  Always build for just latest builds

Hope this has given fair idea about how Jenkins multibranch pipelines work with webhook.