

Managing Pipeline Sources

Overview

A **Pipeline Source** represents a source control repository (such as GitHub or BitBucket) where Pipelines definition files can be found. A pipeline source connects to the repository through an [integration](#). After a pipeline source is added, Pipelines automatically loads all config files from the repository that matches the specified filename filter.



Administrator users can create, view, edit and sync pipeline sources. Non-administrator users can only view and sync pipeline sources.

Page Contents

- [Overview](#)
- [Administering Pipeline Sources](#)
 - [Requirements](#)
 - [Adding a Pipeline Source](#)
 - [From YAML](#)
 - [From Template](#)
 - [Validating YAML](#)
 - [Viewing Pipeline Sources](#)
 - [Syncing a Pipeline Source](#)
 - [Viewing Sync Logs](#)
 - [Deleting a Pipeline Source](#)
 - [Fetching Branches](#)

Administering Pipeline Sources

Only users with [administrator](#) privileges can manage pipeline sources.

Requirements

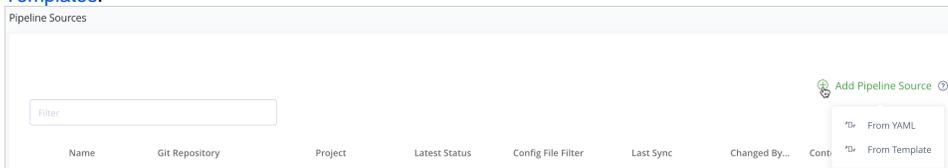
To successfully add a pipeline source, ensure the following:

- The credentials used in the source control integration should have admin access to the specified source control repository. Admin privileges are used to add a webhook that allows Pipelines to be notified of changes in the repository.
- **Pipelines 1.26.0 and lower:** The repository path is valid. To know the exact path to specify, inspect the git clone URL of that repository in your source control system and copy the value. Here are some examples:
 - **GitHub / GitHub Enterprise:** <https://github.com/myuser/basic-pipeline.git> - use *myuser/basic-pipeline*
 - **Bitbucket Server (Private repository):** <https://git.mycompany.com/scm/project-id/repo-name.git> - use *project-id/repo-name*
 - **Bitbucket Server (Individual repository):** <https://git.mycompany.com/scm/~myuser/myfirstrepo.git> - use *~myuser/myfirstrepo*
 - **Bitbucket:** <https://username@bitbucket.org/team-space/test-repo.git> - use *team-space/test-repo*
 - **GitLab:** <https://gitlab.com/user1/repo1.git> - use *user1/repo1*
- Branch name is valid and the credential used in the Git integration for the pipeline source has access to the branch.
- [Add an integration](#) for the source control system where your pipeline file repository is (or will be) located. The integration can be one of these source control system providers:
 - [GitHub Integration](#)
 - [GitHub Enterprise Integration](#)
 - [GitLab Integration](#)
 - [Bitbucket Integration](#)
 - [Bitbucket Server Integration](#)

Adding a Pipeline Source

To add a source control repository as a Pipeline Source:

1. Go to **Administration | Pipelines | Pipeline Sources**.
2. In the resulting **Pipeline Sources** display, click **Add Pipeline Source** and click one of the following:
 - **From YAML:** Add the pipeline source from a YAML configuration file, which contains the declarations of all resources and workflow steps.
 - **From Template:** Add the pipeline source from a pre-defined template by providing a values file. For more information, see [Pipelines Templates](#).



From YAML

- In the **Add YAML Pipeline Source** page, click one of the following:
 - **Single Branch**
 - **Multi Branch**
- Click to select the protocol type to be used for cloning the repository when the pipeline source is synced:
 - **SSH**
 - **HTTPS**
- Complete the **Single/Multi Branch** form:

Field	Description
Name	Enter a unique name for the pipeline source.
SCM Provider Integration	Click the <i>SCM Provider Integration</i> field and select your source control integration from the dropdown list. Only integrations that are compatible for use as a Pipeline Source will be included in the dropdown list.
Repository Full Name	Based on the information you have provided for the selected integration, such as API token, all the relevant repositories are listed in the <i>Repository Full Name</i> field. Select the path of the repository where your pipeline files are stored. If the name of the repository is not auto-fetched, enter the full name of the repository where your pipeline files are stored.
Branch (Single Branch only)	Based on the SCM provider and Repository Full Name you have provided, all the available branches are automatically fetched (for example, <code>main</code>). Select the required branch. If the name of the branch is not auto-fetched, enter the name of the branch.
Exclude (Multi Branch only)	Specify the <i>Exclude Branch Pattern</i> as a regular expression for the matching branch names to exclude.
Include (Multi Branch only)	Specify the <i>Include Branch Pattern</i> as a regular expression for the matching branch names to include.
Folder Name (Pipelines 1.27.0 and higher)	<p>Provide the name of the directory where the YAML config is available.</p> <ul style="list-style-type: none"> • To use Folder Name, in the SCM repository, place your YAML files in a directory named <code>.jfrog-pipelines</code>. This directory can be the root or one level below the root. Directories beyond this level are not supported. • Following are the possible directory structures for monorepos: <pre> - root ----- .jfrog-pipelines Or - root ---- .jfrog-pipelines ----- service1 ----- service2 Or - root ---- service1 ----- .jfrog-pipelines ---- service2 ----- .jfrog-pipelines </pre> • The <code>.jfrog-pipelines</code> directory can contain any number of YAML files. • To parse all YAML files in the root directory, enter <code>.</code>. However, other pipeline sources cannot point to this repository. • You can create multiple pipeline sources pointing to the same SCM repository as long as the directory names are different (and none of them point to root).

Recommended Directory Structure for using Folder Name

Here are some examples of the supported directory structure:

YAML Files Location	Folder Name Path
YAML files are in the <code>.jfrog-pipelines</code> root directory	Enter <code>.</code> (dot) or <code>.jfrog-pipelines</code> in the Folder Name field to fetch all the YAML files
YAML files are in the <code>.jfrog-pipelines/project1</code> directory	Enter <code>.jfrog-pipelines/project1</code> in the Folder Name field to fetch the YAML files in the <code>.jfrog-pipelines/project1</code> directory
YAML files are in the following directories: <ul style="list-style-type: none"> <code>.jfrog-pipelines</code> <code>.jfrog-pipelines/project1</code> <code>.jfrog-pipelines/project2</code> 	Enter <code>.</code> (dot) or <code>.jfrog-pipelines</code> in the Folder Name field to fetch all the YAML files in all directories or Enter <code>.jfrog-pipelines/project1</code> in the Folder Name field to fetch the YAML files in the <code>.jfrog-pipelines/project1</code> directory Enter <code>.jfrog-pipelines/project2</code> in the Folder Name field to fetch the YAML files in the <code>.jfrog-pipelines/project2</code> directory
YAML files are in the following directories: <ul style="list-style-type: none"> <code>.jfrog-pipelines</code> <code>service1/.jfrog-pipelines</code> <code>service2/.jfrog-pipelines</code> 	Enter <code>.</code> (dot) or <code>.jfrog-pipelines</code> in the Folder Name field to fetch the YAML files in the <code>.jfrog-pipelines</code> directory, but YAML files in the <code>service1/.jfrog-pipelines</code> and <code>service2/.jfrog-pipelines</code> directories will not be included. or Enter <code>service1/.jfrog-pipelines</code> in the Folder Name field to fetch specific YAML files in the <code>service1/.jfrog-pipelines</code> directory Enter <code>service2/.jfrog-pipelines</code> in the Folder Name field to fetch specific YAML files in the <code>service2/.jfrog-pipelines</code> directory

If you have a monorepo with multiple services within a single repo, a directory structure such as the following is recommended:

Existing Directory Structure	Recommended Directory Structure
Root Build/ci - Service1 - Build/ci/pipe.yaml - Service2 - Build/ci/pipe.yaml	Root .jfrog-pipelines - Service1/pipe.yaml - Service2/pipe.yaml

Migrating from Config File Filter to the New Folder Structure

In Pipelines version 1.27.0, **Config File Filter** has been replaced with a new field called **Folder Name** for adding your pipeline sources. Though **Config File Filter** is still supported and available while editing a pipeline source.

- **This change does not affect pipeline sources added before 1.26.0.** However, it is highly recommended that you use the recommended folder structure for new pipeline sources and move your existing sources to the recommended folder structure. Using the folder structure significantly improves the performance of your pipeline sync.
- You can take advantage of the **Folder Name** feature by editing the pipeline source and then providing the new directory path. In this case, the entry in the **Config File Filter** is ignored and the pipeline source in the directory path is used.
- If both the fields have values, then the **Folder Name** field takes precedence. However, if no `.jfrog-pipelines` directory is found in the directory path, then information in the **Config File Filter** field is used to find the YAML files.

Config File Filter (Pipelines 1.26.0 and lower)	Enter a regular expression. Any filenames that match the expression will be loaded. Examples: <ul style="list-style-type: none"> • If you plan to declare your resources in a file named <code>pipelines.resources.yml</code> and your pipelines in <code>pipelines.workflows.yml</code>, then you might enter <code>pipelines.*.yml</code>. • If you have multiple yaml config files, then enter <code>.*yml</code>. • In addition to the <code>pipelines.yml</code>, if you are using a <code>values.yml</code>, ensure both files are added to the filter, using <code>(pipelines values).yml</code>.
---	--

From Template

1. Complete the **Template Properties** form:

- Click the *Select Template Namespace* field and select the namespace.
 - Click the *Select Template Names* field and select the template.
 - Click the *Select Template Version* field and select the version for the template.
2. Click **Next**.
In the resulting **Specify values file** page, click one of the following from the **Read YAML values from** drop-down:
 - **Source Control Repository**: Use this when you want to fetch the `values.yml` file from an SCM provider.
 - **Enter Manually**: Use this when you want to manually the define the values for the pipeline source.
 3. **Source Control Repository**
Click one of the following:
 - **Single Branch**
 - **Multi Branch**
 4. Click to select the protocol type to be used for cloning the repository when the pipeline source is synced:
 - **SSH**
 - **HTTPS**
 5. Complete the **Single/Multi Branch** form:

Field	Description
Name	Enter a unique name for the pipeline source.
SCM Provider Integration	Click the <i>SCM Provider Integration</i> field and select your source control integration from the dropdown list. Only integrations that are compatible for use as a Pipeline Source will be included in the dropdown list.
Repository Full Name	Based on the information you have provided for the selected integration, such as API token, all the relevant repositories are listed in the <i>Repository Full Name</i> field. Select the path of the repository where your pipeline files are stored. If the name of the repository is not auto-fetched, enter the full name of the repository where your pipeline files are stored.
Branch (Single Branch only)	Based on the SCM provider and Repository Full Name you have provided, all the available branches are automatically fetched (for example, <code>main</code>). Select the required branch. If the name of the branch is not auto-fetched, enter the name of the branch.
Exclude (Multi Branch only)	Specify the <i>Exclude Branch Pattern</i> as a regular expression for the matching branch names to exclude.
Include (Multi Branch only)	Specify the <i>Include Branch Pattern</i> as a regular expression for the matching branch names to include.

Values File Path	Enter the path to the <code>.values.yml</code> file, which contains the values for the template.
-------------------------	--

6. **Enter Manually**

- **Name:** Enter a unique name to identify a pipeline source definition created from the template.
- **Values.yml:** Manually enter or paste the Values definition.

7. Click **Create Source** to complete adding the Pipeline Source.

If JFrog Pipelines successfully syncs with the repository, the **Status** field displays "Success". Otherwise, it displays "Failed".

Pipeline Sources

+ Add Pipeline Source ⓘ

Filter

Name	Git Repository	Project	Latest Status	Config File Filter	Last Sync	Changed By...	Context
myPipeline	jfrog/Swampup2021 Branch: master		✔ Success	yml	16-08-21 12:52:5...	admin	02d0e04 Update values.yml

All pipeline config files committed to the repository will be automatically loaded into JFrog Pipelines, and all defined pipelines from that source are listed in the pipeline list.

Validating YAML

Before adding a pipeline source, you have the option of validating your YAML and then committing it to the SCM. This enables you to get instant feedback on pipeline sync errors. The YAML validator will validate your YAML for both semantic and syntactic errors.

To validate your YAML:

1. Click **Validate YAML**.
2. Either enter the YAML and paste the YAML contents in the Pipelines and Resources fields. For more information about the Pipelines DSL, see [Defining a Pipeline](#).
As the YAML content is entered, it is validated and syntactic errors, if any, are displayed.
3. After fixing all the syntactic errors, click **Validate** to validate the YAML for semantic errors.

Pipeline Sources

+ Add Pipeline Source ⓘ

Validate YAML

Filter

Name	Git Repository	Project	Latest Stat...	Config File Filt...	Last Sync ...	Chang...	Context
pipe28...	nitind123/pipelines-local- Branch: pipe28189		❌ Failure	*.yml	18-05-22 1...	kunal...	15578ed Update pipeli...
jenkins...	raksha78/bash-step Branch: main		✔ Success	jenkins.yml	04-07-22 1...	raksha...	7d393c2 Update jenkins...
IN/miss...	IN/mission-control-install. Branch: test-4,x-mono		❌ Failure	pipelines.yml	05-12-20 1...	Ambris...	e720ed4 JFI-0 - Remove...
vishnur...	vishnurao12/pipelines Branch: test9		✔ Success	pipelines.yml	22-11-21 1...	vishnur...	fb4566f ci 2
~bens/...	~bens/hello-changes Branch: master		❌ Failure	pipelines.yml	20-03-21 0...	Ben Ste...	ef53261 use custom im...
vishnur...	vishnurao12/pipelines Branch: master		✔ Success	pipelines.yml	29-05-22 1...	vishnur...	57a6327 Update pipeli...
~IDANS...	~IDANS/clamav-tests Branch: master		❌ Failure	pipeline.yml	10-03-21 0...	Idan So...	408f0e3 pipeline

Viewing Pipeline Sources

To view the list of pipeline sources already added to Pipelines, go to **Administration | Pipelines | Pipeline Sources** or **Application | Pipelines | Pipeline Sources**.

The page displays the list of pipelines sources that are available to you according to the permission targets defined in the JFrog Platform. Your user account must be granted [permissions](#) for a pipeline source for it to be shown.

Each row of the pipeline sources list includes the following:

Name	<ul style="list-style-type: none"> • For new pipeline sources: This is the name you provided while creating the pipeline source • For existing pipeline sources: This defaults to <code>repositoryName/branch</code> for single branch and <code>repositoryName</code> for multi-branch
Git Repository	The source repository path of the pipeline source
Project	The Project that the pipeline source belongs to
Latest Status	The success/failure status of the last sync
Config File Filter	The filter string for the pipeline config files
Last Sync	The time and date of the last sync
Changed By	Name of the user who made the last update to the pipeline source
Context	The commit SHA that triggered the last sync
Logs	Click the Logs link to view the log from the last sync. Use this to diagnose a failure to sync a pipeline source.

If the pipeline source is a multi branch source, the row presents aggregate information for all branches, and can be expanded/collapsed to show the sync status of each branch.

Syncing a Pipeline Source

When any of the pipeline config files have changed, you will need to sync the pipeline source to reload:

1. Go to **Administration | Pipelines | Pipeline Sources** and click the **Actions** button located at the far right.
2. Click **Sync**.

Name	Git Repository	Project	Latest Status	Config File Filter	Last Sync	Changed By...	Context
myPipeline	jfrog/Swampup2021 Branch: master		✔ Success	.yml	16-08-21 12:52:5...	admin	02d0e04 Update values.yr

⌵

- 📄 Logs
- ↻ Sync
- ✎ Edit
- ✖ Delete

Viewing Sync Logs

To view the logs of all synchronized config files:

1. Go to **Administration | Pipelines | Pipeline Sources** and click the **Actions** button located at the far right.
2. Click **Logs**.

Sync Logs

These sync logs were generated at 2021-08-17T05:41:21.714Z

```
Successfully synced pipeline source
```

```
SUMMARY:
```

```
Updated 2 pipelines
```

```
DETAILS:
```

```
Processing pipeline source with local templates.
```

```
Pipeline config file filter: .yaml
```

```
pipelines.yaml: successfully updated pipeline: app_prod_pipeline
```

```
pipelines.yaml: successfully updated pipeline: app_dev_pipeline
```

Pipeline Source Sync Recovery

Pipelines provides an option to enable auto-sync for pipelines sources that are out of sync. For more information, see [Pipeline Source Sync Recovery](#).

Deleting a Pipeline Source

When a pipeline source is deleted, it deletes all the pipelines associated with the source.

To delete a pipeline source:

1. Go to **Administration | Pipelines | Pipeline Sources** and click the **Actions** button located at the far right.
2. Click **Delete**.

Name	Git Repository	Project	Latest Status	Config File Filter	Last Sync	Changed By...	Context
myPipeline	jfrog/Swampup2021 Branch: master		 Success	.yaml	17-08-21 11:11:2...	admin	02d0e04 Update values.y...



3. Click **Confirm** on the confirmation window that appears.

 A pipeline source cannot be deleted if any of its associated pipeline is still running. You can either cancel the run or wait for the run to complete and then delete the pipeline source.

Fetching Branches

When one or more branches are created for a Multibranch pipeline source, the webhooks may not be processed correctly, resulting in those branches not being added to the pipeline source.

Pipelines can automatically fetch and sync missing branches in a multibranch pipeline source based on the selected SCM integration and the full name of the repository.

To do this:

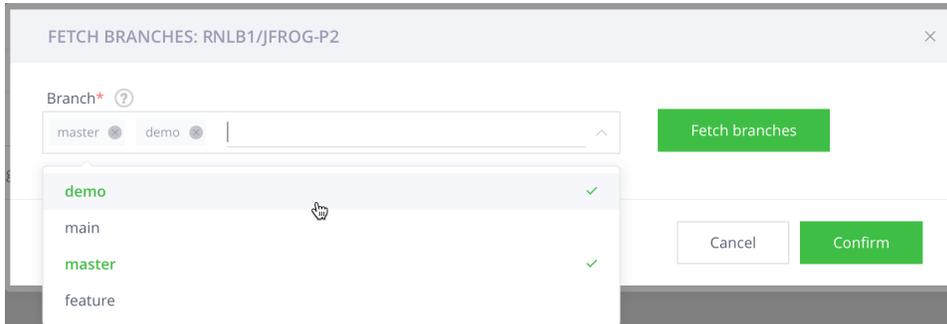
1. Click the **Actions** button for the multibranch pipeline, and then click **Fetch Branches**.

Repository	Branch	Latest Status	Last Sync	Changed By	Context	Logs...
mlb1/jfrog-p2	Multibranch	4 Success				  

The **Fetch Branches** window appears.

- In the **Branch** field, you can either:
 - Enter the name of the branch you want to fetch and sync.
 - or
 - Click the **Fetch Branches** button to try to automatically fetch all the branches based on the configured SCM integration and repository.If one or more branches are found, they are listed in the drop-down menu.

- Click to select the relevant branches.



- Click **Confirm** to sync the pipeline source.
The newly found branches are now listed on the Pipeline Sources page.