2. Click **Set up Git sync**.
3. On the **Configure service credentials** page, select a service and protocol, add your GitHub App, SSH private key, or GitHub token, then click **Next**.
    - To connect to GitLab or Bitbucket, use your SSH private key.

Working with Git Sync

The Git sync extension syncs the following resources from Segment to your Git repository:

- Sources and Destinations
- Warehouses
- Destination Filters and Mappings for Connections
- Tracking Plans
- Functions
- Transformations

- Reverse ETL
- Users and User groups
- Labels

The Git sync extension doesn't support the following resources:

- Spaces
- Audiences and Journeys
- Data Graph
- Mappings for Linked Audiences

Reach out to Segment support to request support for additional Git Sync resources.

After you set up the Git sync extension for the first time, Segment performs an initial sync that sends the current state of your Segment workspace to the Git repository you connected. Segment automatically tracks all following workspace updates.

You can manually trigger syncs at any time by clicking **Full Sync** on the Git Sync page. To disable Git Sync from the Git Sync page, switch the **Enabled** toggle to off.

Git Sync architecture and data model

Because a Segment workspace can represent a distinct environment (testing, staging, production), each workspace is mapped directly to a single Git repository. This direct mapping ensures a clear and organized relationship between workspace resources and a Git repository.

Segment uses its Terraform provider to manage key functions like tracking changes and retrieving information about those changes in Segment. Segment stores changes in HashiCorp Configuration Language (HCL), the format used by Terraform. To learn more about HCL and how it compares to JSON or YAML, visit HashiCorp's HCL repository on GitHub.

Using HCL makes it easier to document Segment's data model, especially for users managing versioning and Git Sync with Terraform. It also helps manage Segment configurations directly from Git. For more details on the Git Sync data model, read Segment's Terraform provider documentation.

Managing your Segment workspace with Terraform and Git Sync

Segment supports one-way synchronization from Segment to Git, but you can set up two-way synchronization using the Segment Terraform provider.

Terraform offers an open-source way to manage Segment resources through a Git repository as an alternative to a fully managed two-way sync. This method requires third-party tools like Atlantis for CI integration.

To manage Segment resources using Git and Terraform, follow these steps:

1. Copy the generated Terraform configuration for the resources you want to manage into a separate Git repository dedicated to Terraform.
2. Include the following provider configuration blocks:

```
# providers.tf

terraform {
required_providers {
    segment = {
    source  = "segmentio/segment"
    version = "1.0.4"
    }
}
}

provider "segment" {
# Provide the token directly or load it from an environment variable
}
```

2. Apply configuration changes by running Terraform locally or using a tool like Atlantis to run it directly from your Git provider.

For more information on using Terraform, visit Terraform's documentation.

Git Connections

Git Connections enable Segment to sync data with your preferred Git repository through supported like SSH and token-based authentication.

> Git Sync and the dbt integration operate independently. You don't need to set up Git Sync to use dbt, and dbt Cloud can trigger its own syncs without relying on Git Sync.

Supported connection types

Segment supports the following credential types for setting up a Git Connection:

- **SSH**: Compatible with GitHub, GitLab, and Bitbucket, SSH provides a secure method for connecting to your repository.
- **Git token**: Git tokens are also supported across GitHub, GitLab, and Bitbucket, enabling token-based authentication..
- **GitHub App**: For GitHub users, GitHub App integrations offer enhanced security and functionality. This method is exclusive to GitHub and supports additional features, like CI checks.

Reusing Git Connections

Segment lets you set up multiple Git Connections, allowing you to reuse credentials across both dbt and Git Sync. You can either use the same credential for multiple configurations or create separate Git Connections for each product and environment as needed.

If you plan to reuse a Git token across both dbt and Git Sync, ensure it has the necessary read and write permissions for both integrations.

Troubleshooting Git Sync

When setting up Git Sync, you may run into an access error with the following message:

`"Unable to create Git Sync due to Git connection issues. Please check your configuration and try again.`

This error can occur if there are issues with your Git connection settings or permissions. To resolve the error, verify that:

- Your credentials have write access to the Git repository, as Segment requires this to sync changes.
- Your repository is hosted by GitHub, GitLab, or Bitbucket (Segment doesn't support self-hosted repositories).

Branch protections are disabled on the repository.

This page was last modified: 07 Nov 2024

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

Visit our Support page

Help improve these docs!

Edit this page

Request docs change

Was this page helpful?

Yes

No

Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

Your work e-mail

Request Demo

or

Create free account