Dataform

Google Dataform

- Dataform is a tool built into BigQuery that simplifies building, testing, and deploying SQL pipelines.
- Offers dependency management, testing, scheduling, and version control in one place.
- Lets you write SQL transformations in .sqlx files instead of long, monolithic SQL scripts

Benefits of Dataform

Centralized Workflow

- Instead of writing SQL transformations in multiple places or ad hoc scripts, Dataform allows you to keep all your SQL logic in one repository.
- Everything from raw data transformations to final dashboards is organized in a single project.
- Makes collaboration easier since the entire team works from the same codebase.
- Helps maintain consistency in naming conventions, business logic, and data definitions across the organization.
- Example. A marketing analytics team can store raw clickstream data, customer profiles, and campaign KPIs transformations all in one Dataform project, making it easy to track how raw data turns into reports.

Benefits of Dataform

Version Control (GitHub / GitLab Integration)

- Dataform integrates with GitHub or GitLab so every change to a transformation file (.sqlx) is versioned and tracked.
- Multiple developers can work on the same project without overwriting each other's changes.
- Use Git features like branches, pull requests, and code reviews before merging changes.
- You can roll back to previous versions if something breaks.
- Example: A data engineer creates a new table transformation on a separate branch. Another engineer reviews it via a pull request before merging into the main production branch.

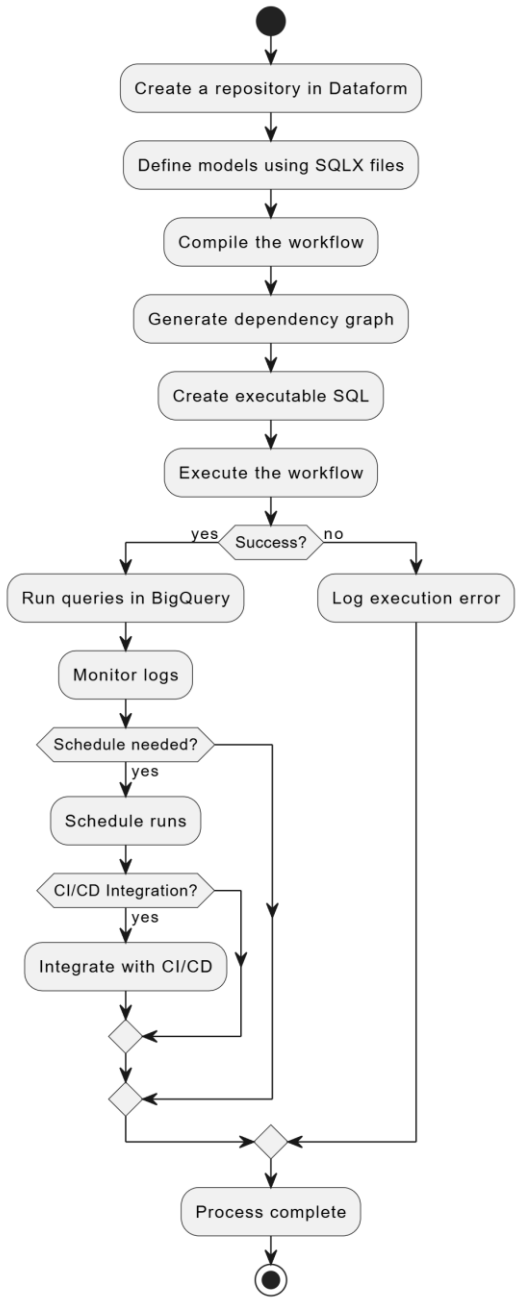
Benefits of Dataform

Dependency Tracking

- Dataform uses ${ref("table_name")} to automatically figure out which queries depend on which tables.
- No need to manually track the order of SQL scripts.
- Dataform builds a dependency graph so transformations run in the right sequence.
- If one table fails, dependent tables will not run, avoiding partial pipeline failures.
- Example: Table B depends on Table A because it aggregates A's results. Dataform ensures A is always built before B automatically.

Dataform end-to-end flow

- Create a repository in Dataform.
- Define models using SQLX files for tables, views, and operations.
- Compile the workflow → Dataform creates a dependency graph and generates executable SQL.
- Execute the workflow → Runs queries in BigQuery following dependencies.
- Monitor logs, schedule runs, and integrate with CI/CD if needed.



Dataform Repository

- Each Dataform repository contains a collection of **SQLX** and **JavaScript** files that define the workflow, along with configuration files and required packages.
- The repository contents are managed inside a development workspace, where changes are developed and tested before deployment.
- Dataform relies on Git for version control, ensuring that every change is tracked.
- A Dataform repository page typically includes several key components, such as the file explorer, development workspace, version control panel, and execution logs, all designed to support efficient workflow management.

Components of a Dataform Repository Page

- **Development Workspaces:** Lists all development workspaces created in the repository. Allows development of SQLX files and testing of workflows before deployment.
- **Release Configurations:** Provides options to inspect, create, edit, or delete release configurations. Releases define how and when workflow changes move to production.
- **Workflow Execution Logs:** Shows detailed logs of workflow executions. Useful for monitoring job status, performance, and errors.
- **Workflow Configurations:** Provides access to inspect, create, edit, or delete workflow configurations. Workflow configurations control schedules, triggers, and dependencies for pipelines.
- **Settings:** Displays repository details such as name and location. For repositories connected to third-party Git platforms, shows the Git source platform (GitHub, GitLab, or Bitbucket), Default branch name and Secret token used for authentication.

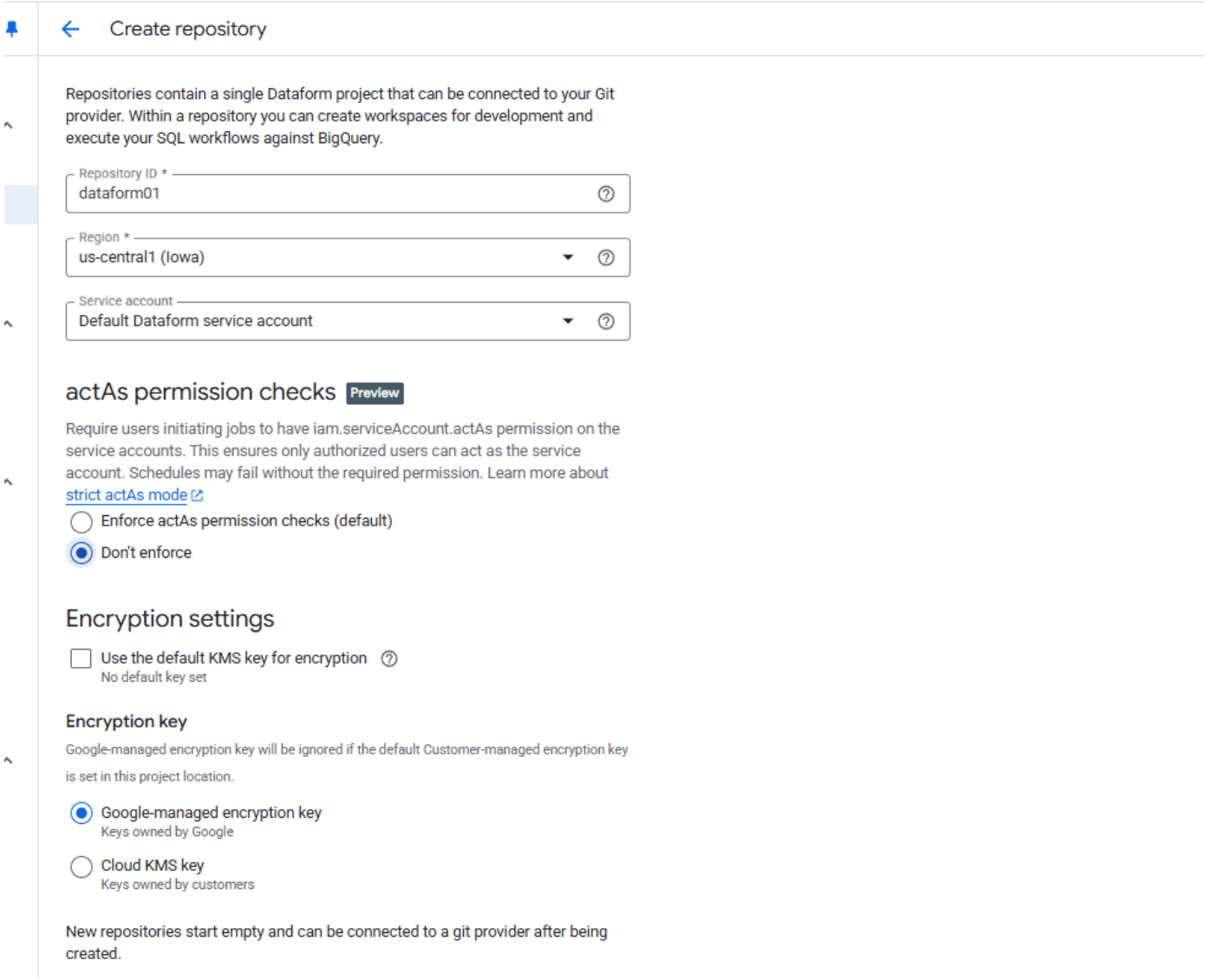
Repository Settings in Dataform

- **Repository ID:** A unique identifier for the repository. Can include only numbers, letters, hyphens, and underscores.
- **Region:** The storage region for the repository and its contents. This can differ from the processing region where workflows run, and results are stored. By default, the processing region matches the default BigQuery dataset region. It can be updated later in the workflow settings file.
- **Service Account:** The service account associated with the repository. Options include the default Dataform service account, a project-specific service account, or a manually specified one. The default service account manages all repository operations, while workflow execution can use a different service account if needed.
- **Strict Act-As Mode (Preview):** Adds an extra security check requiring the iam.serviceAccounts.actAs permission on the service account before workflows can be executed.
- **Encryption:** Controls how data in the repository is encrypted. Options include default encryption, a default Dataform CMEK key, or a customer-managed Cloud KMS key for organizations needing full control over encryption keys.

Required Roles in Dataform

- **Dataform Admin Role:** The roles / dataform.admin IAM role is required to create and delete a Dataform repository. This role can be granted on repositories by a project administrator.
- **Custom Roles or Predefined Roles:** Permissions may also be provided through custom roles or other predefined roles, depending on organizational policies.
- **Service Account Access:** A custom service account can be associated with a Dataform repository to execute its workflows. All other repository operations, such as creating or managing configurations, continue to use the default Dataform service account.
- **Automatic Role Assignment:** After a repository is created, the creator automatically receives the Dataform Admin role on that repository.
- **BigQuery Workflow Execution:** Additional roles might be required for executing workflows in BigQuery, depending on the datasets and resources involved.

Dataform – Create a Repository



Dataform – Settings

dataform01

Development Workspaces | Workflow Execution Logs | Releases & scheduling | Settings

Connect with Git | Configure private npm packages

Name	dataform01
Location	us-central1
Service account	Default Dataform service account
actAs permission checks Preview	
Enforcement	Don't enforce
Encryption Settings	
Type	Google-managed

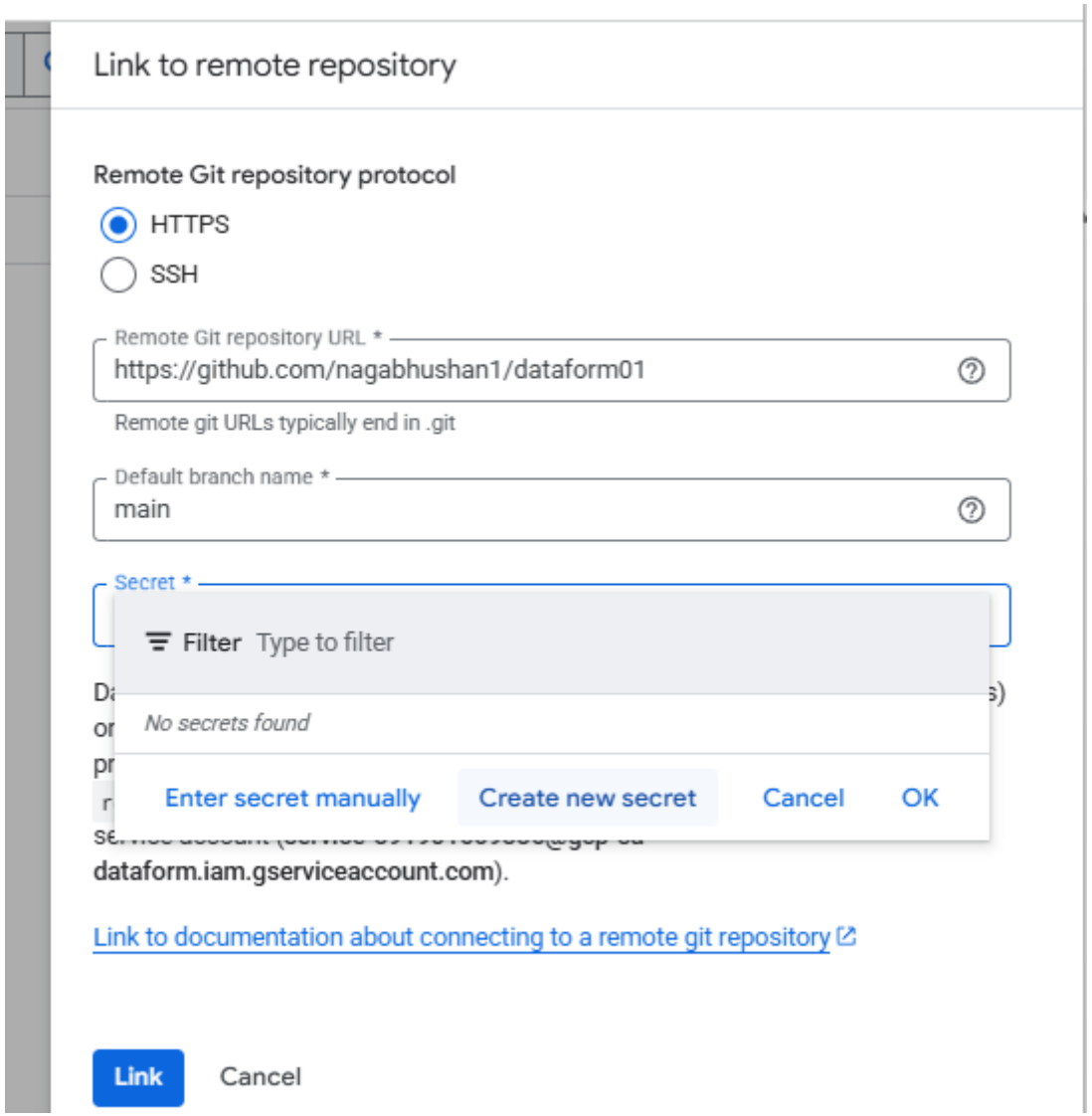
Workspace compilation overrides

Edit

Override the target project ID, table prefix, and schema suffix settings for manual executions of all workspaces in the repository. The default settings are stored in `workflow_settings.yaml`. Learn more about workspace compilation overrides.

No workspace compilation overrides.

Dataform – Connect with Git



Dataform – Connect with Git

dataform01

Development Workspaces | Workflow Execution Logs | Releases & scheduling | Settings

Edit Git connection | Configure private npm packages

Successfully linked to remote Git repository | Dismiss

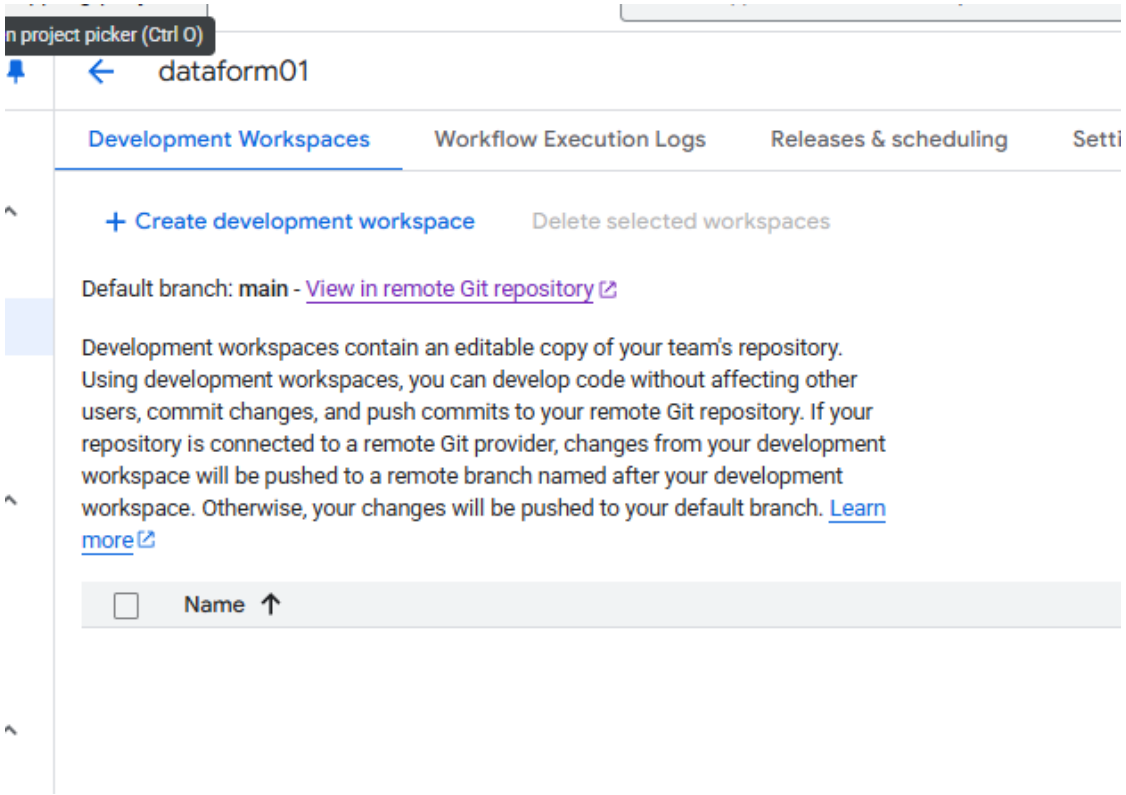
Name	dataform01
Location	us-central1
Service account	Default Dataform service account
actAs permission checks Preview	
Enforcement	Don't enforce
Git Connection Settings	
Repository source	https://github.com/nagabhushan1/dataform01
Default branch name	main
Secret Token	projects/391901669335/secrets/form01_secret/versions/latest
Encryption Settings	
Type	Google-managed

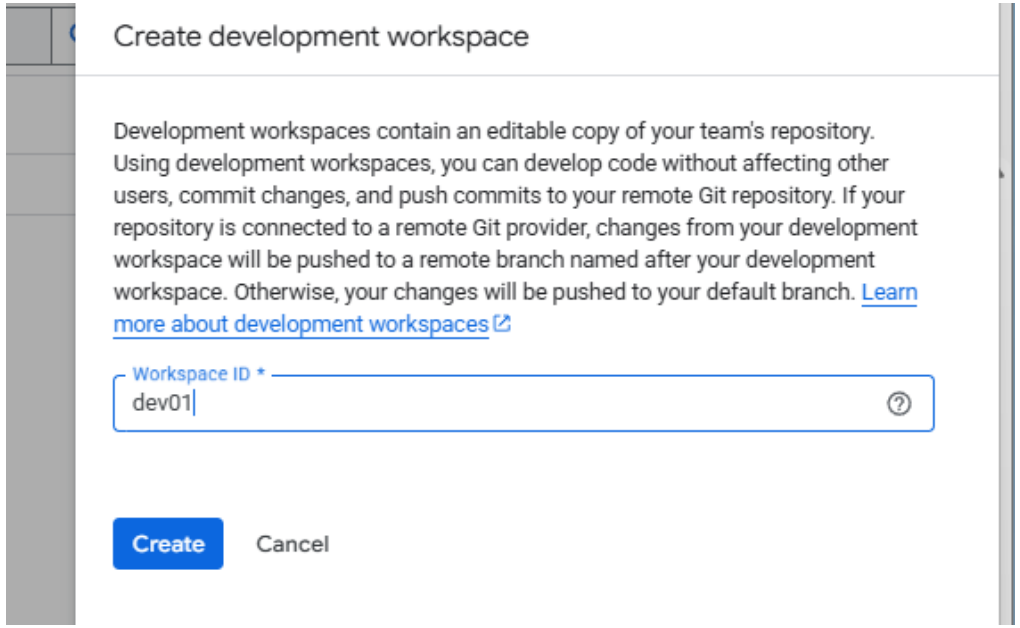
Workspace compilation overrides | Edit

Override the target project ID, table prefix, and schema suffix settings for manual executions of all workspaces in the repository. The default settings are stored in `workflow_settings.yaml`. Learn more about workspace compilation overrides.

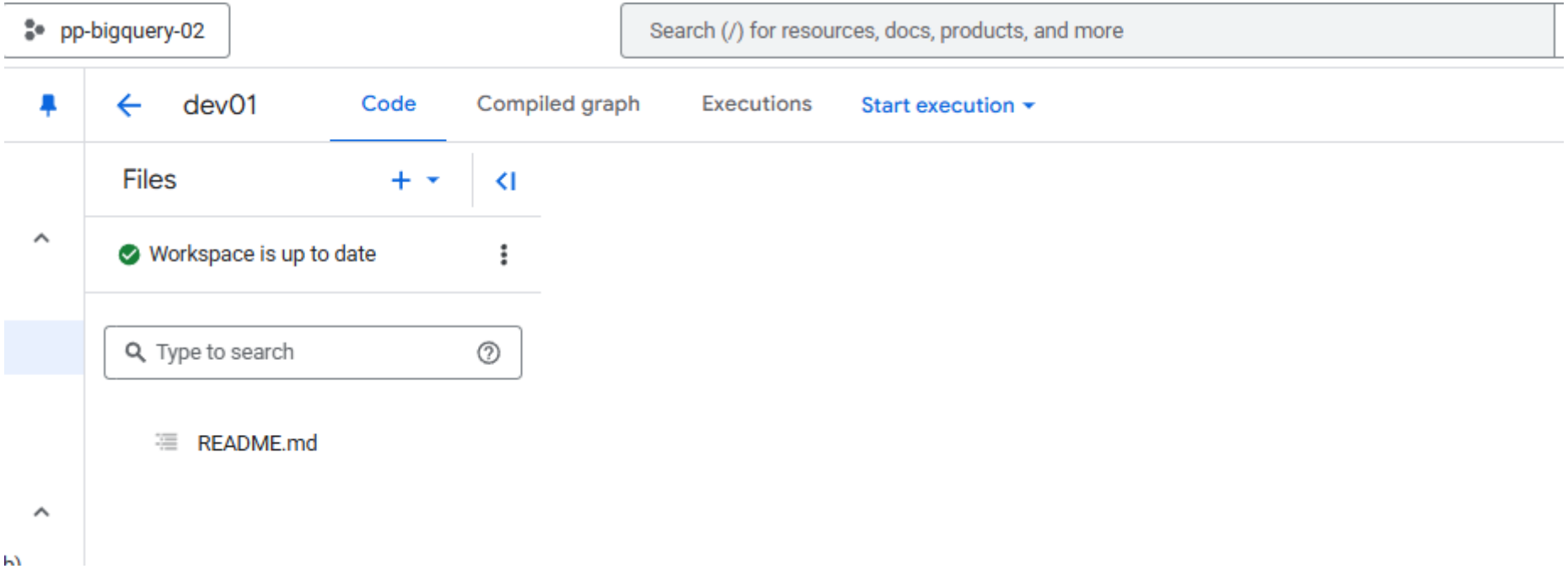
No workspace compilation overrides.

Dataform – Create a development workspace

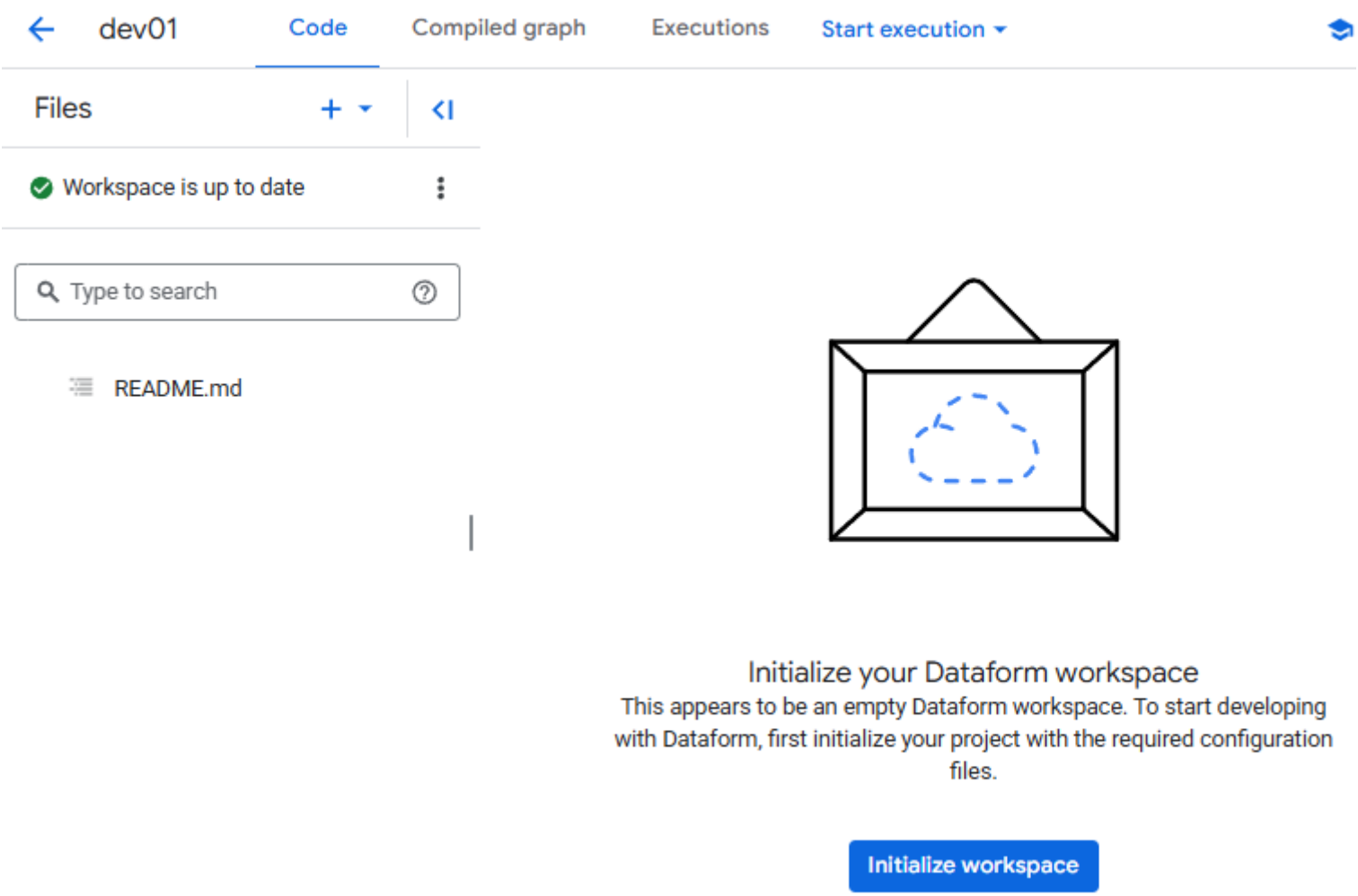




Dataform – Create a development workspace



Dataform – Initialize workspace



Dataform – Commit Changes

dev01 | Code | Compiled graph | Executions | Start execution

Files

Commit 4 changes

Type to search

- *definitions
- includes
- *.gitignore
- README.md
- *workflow_settings.yaml

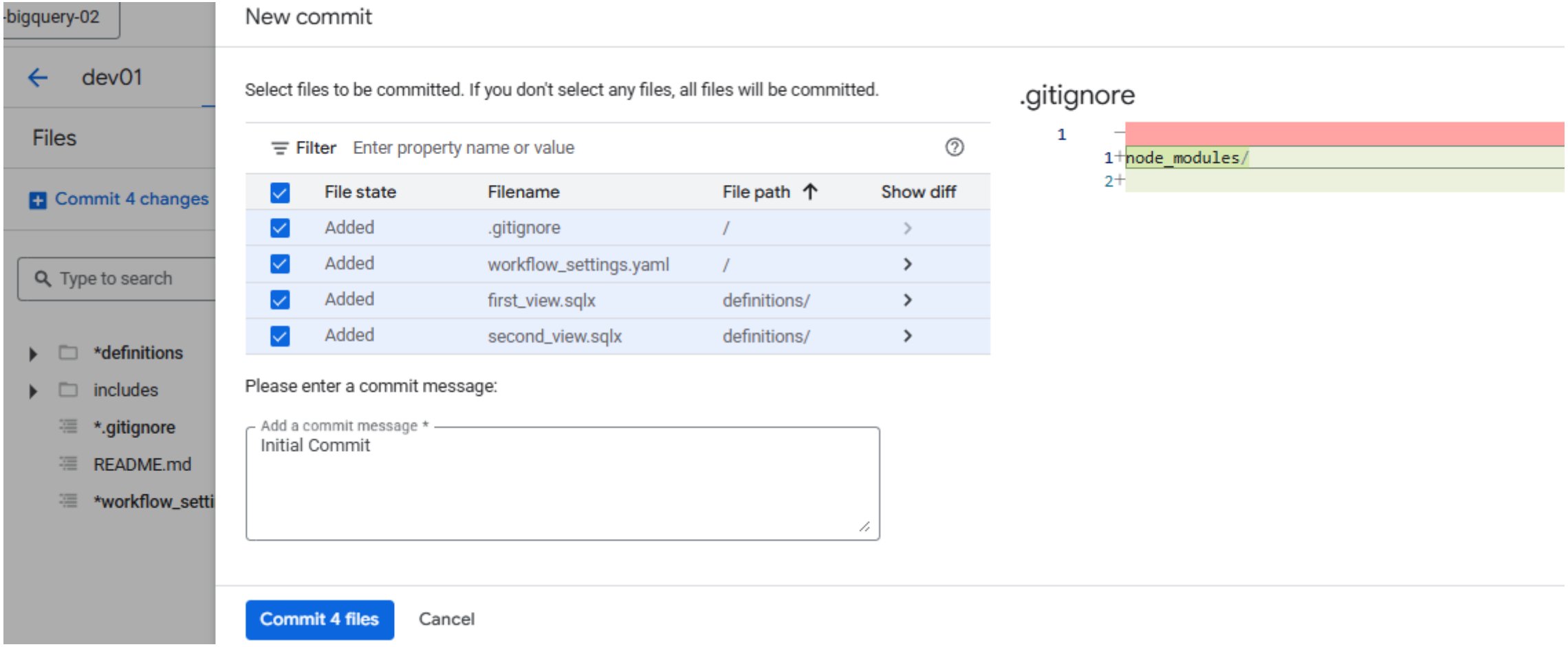
first_vie...qlx

definitions/first_view.sqlx

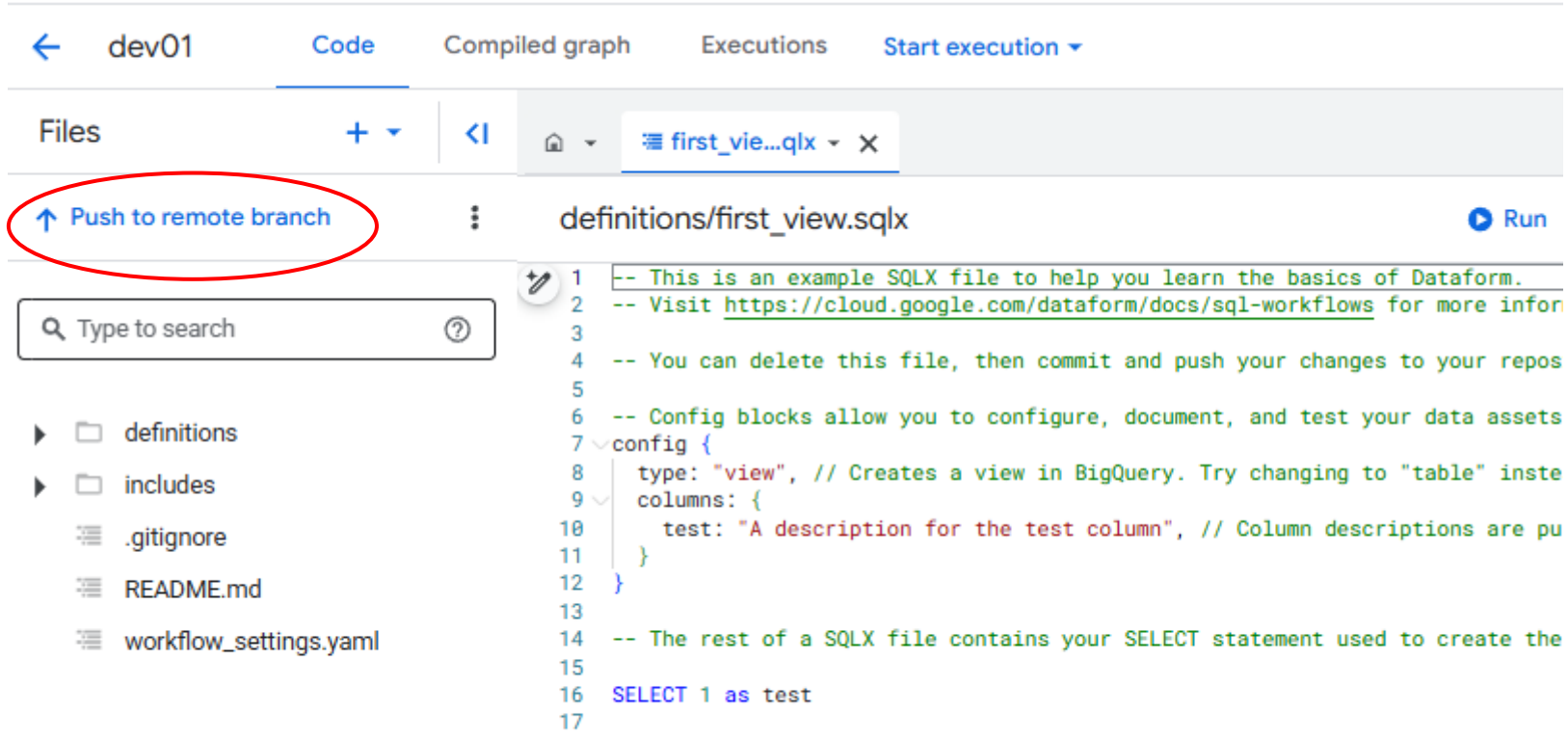
Run | Format

```
-- This is an example SQLX file to help you learn the basics of Dataform.
-- Visit https://cloud.google.com/dataform/docs/sql-workflows for more information on how to confi

-- You can delete this file, then commit and push your changes to your repository when you are rea

-- Config blocks allow you to configure, document, and test your data assets.
config {
  type: "view", // Creates a view in BigQuery. Try changing to "table" instead.
  columns: {
    test: "A description for the test column", // Column descriptions are pushed to BigQuery.
  }
}

-- The rest of a SQLX file contains your SELECT statement used to create the table.

SELECT 1 as test
```

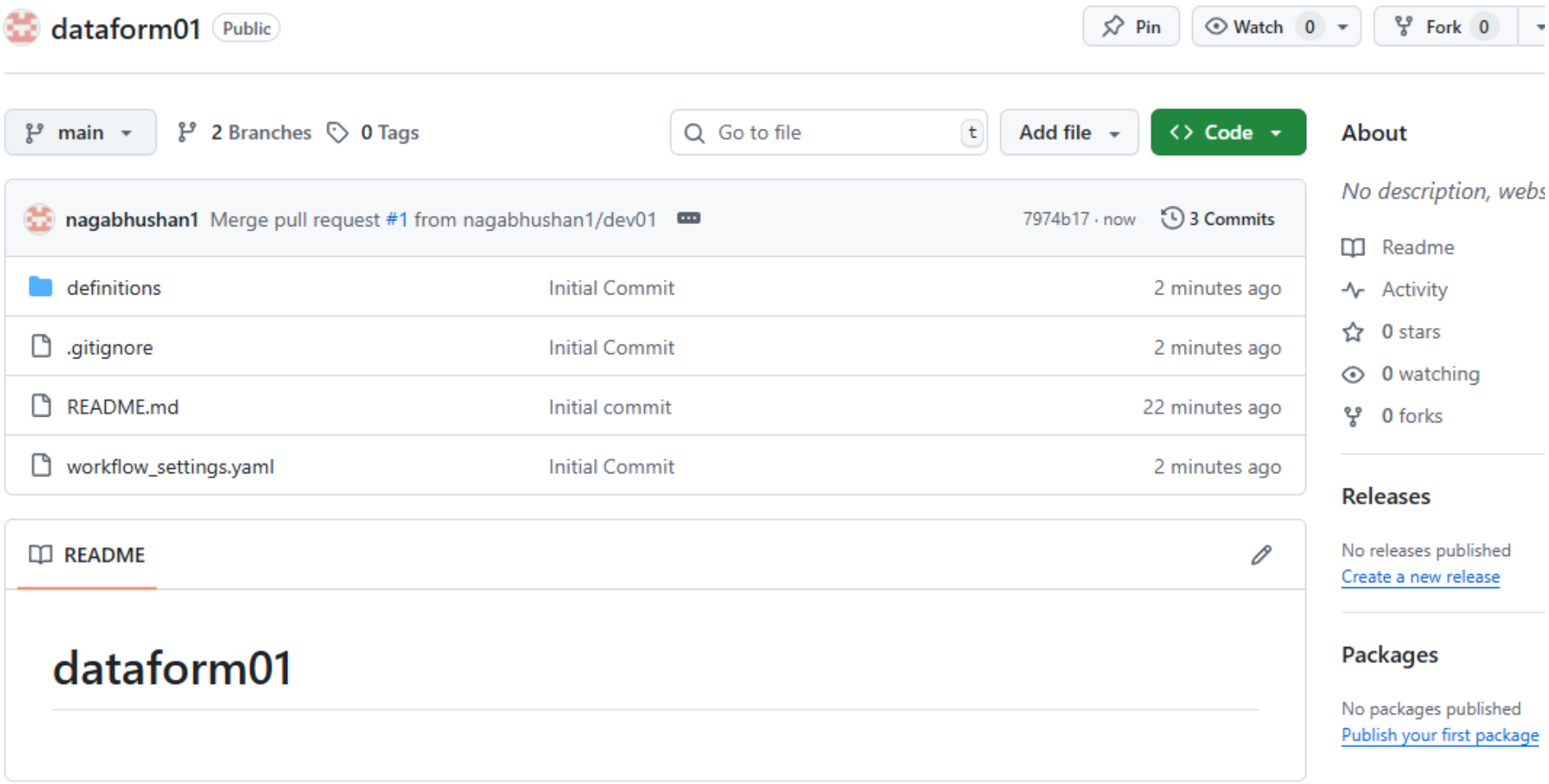
Dataform – Commit Changes



Dataform – Push to remote branch



Dataform – Confirm pull on GitHub



Dataform – Service Account IAM Roles