

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.

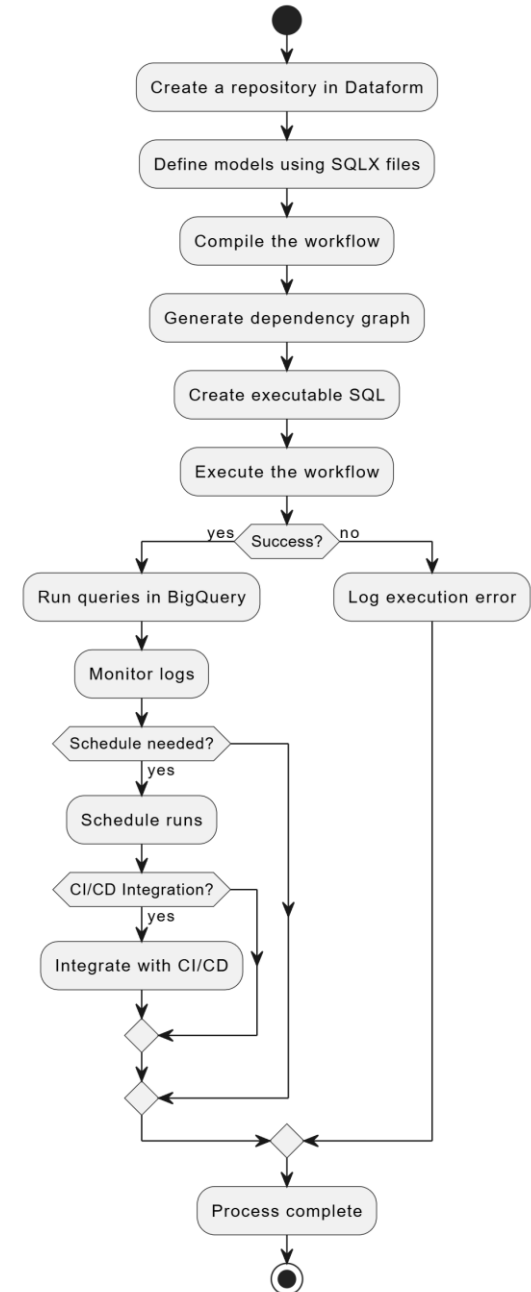
Benefits of Dataform

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

  Create repository

^

Repositories contain a single Dataform project that can be connected to your Git provider. Within a repository you can create workspaces for development and execute your SQL workflows against BigQuery.

Repository ID *

dataform01

?

Region *

us-central1 (Iowa)

▼

?

Service account

Default Dataform service account

▼

?

actAs permission checks Preview

Require users initiating jobs to have iam.serviceAccount.actAs permission on the service accounts. This ensures only authorized users can act as the service account. Schedules may fail without the required permission. Learn more about [strict actAs mode](#)

☐ Enforce actAs permission checks (default)

☒ Don't enforce

Encryption settings

☐ Use the default KMS key for encryption ?
No default key set

Encryption key

Google-managed encryption key will be ignored if the default Customer-managed encryption key is set in this project location.

☒ Google-managed encryption key
Keys owned by Google

☐ Cloud KMS key
Keys owned by customers

New repositories start empty and can be connected to a git provider after being created.

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

Link to remote repository

Remote Git repository protocol

☒ HTTPS

☐ SSH

Remote Git repository URL * ?

Remote git URLs typically end in .git

Default branch name * ?

Secret *

Type to filter

No secrets found

[Enter secret manually](#) [Create new secret](#) [Cancel](#) [OK](#)

dataform.iam.gserviceaccount.com).

[Link to documentation about connecting to a remote git repository](#)

[Link](#) [Cancel](#)

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	<div>Preview ✎</div>
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

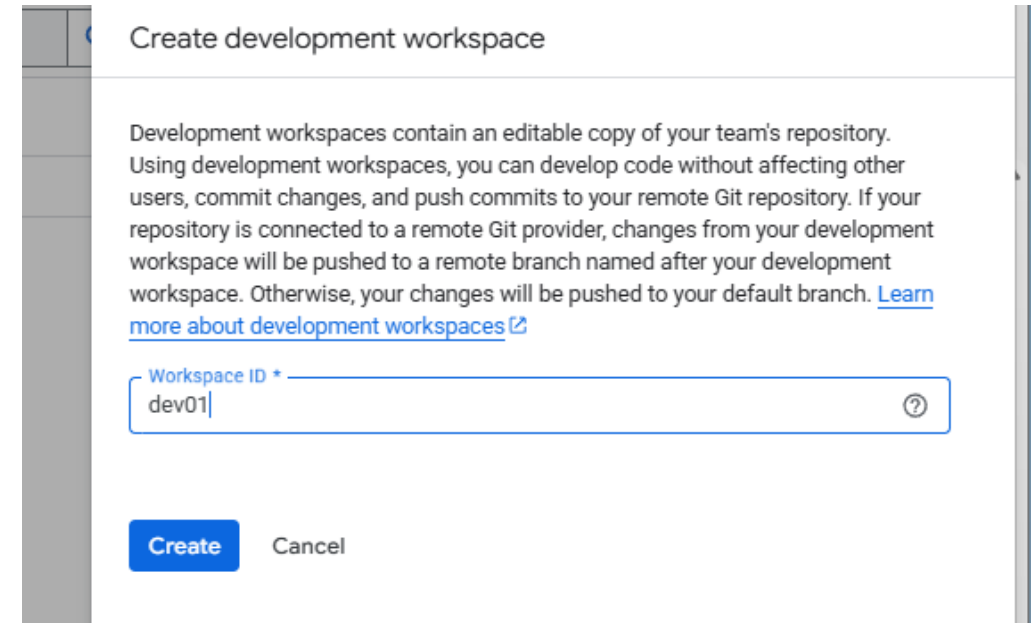
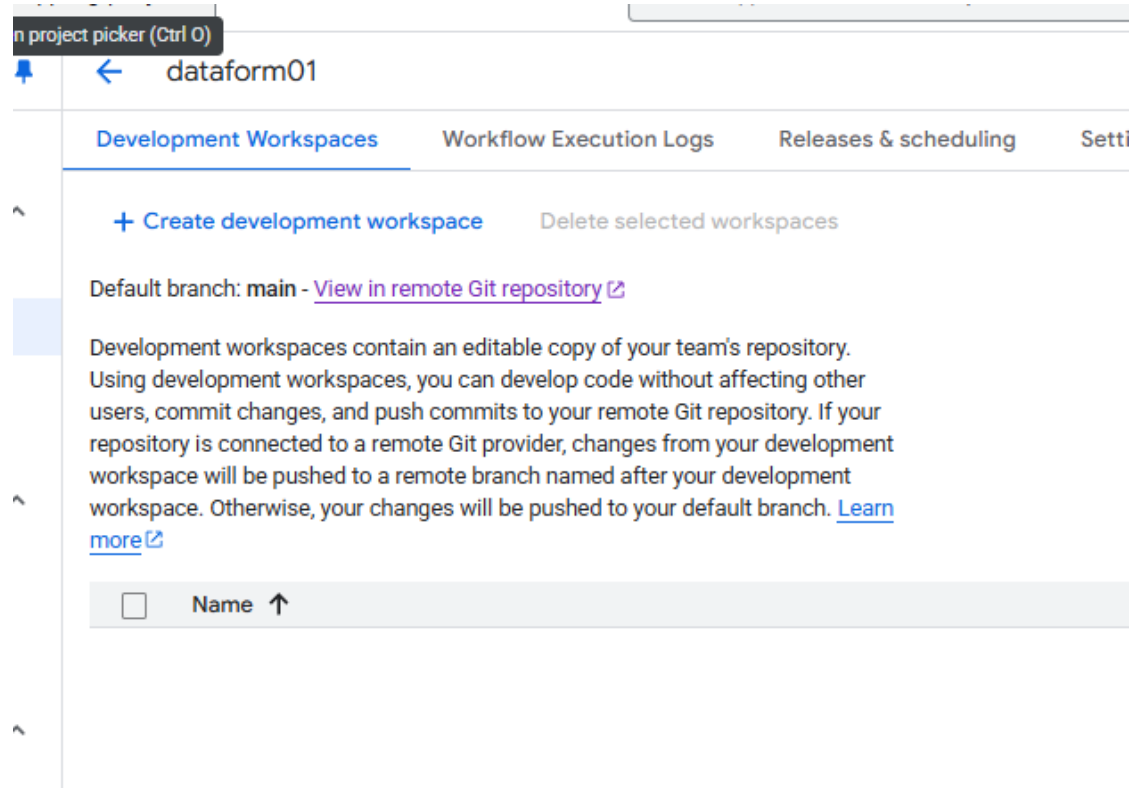
Type	Google-managed
------	----------------

Workspace compilation overrides [✎ Edit](#)

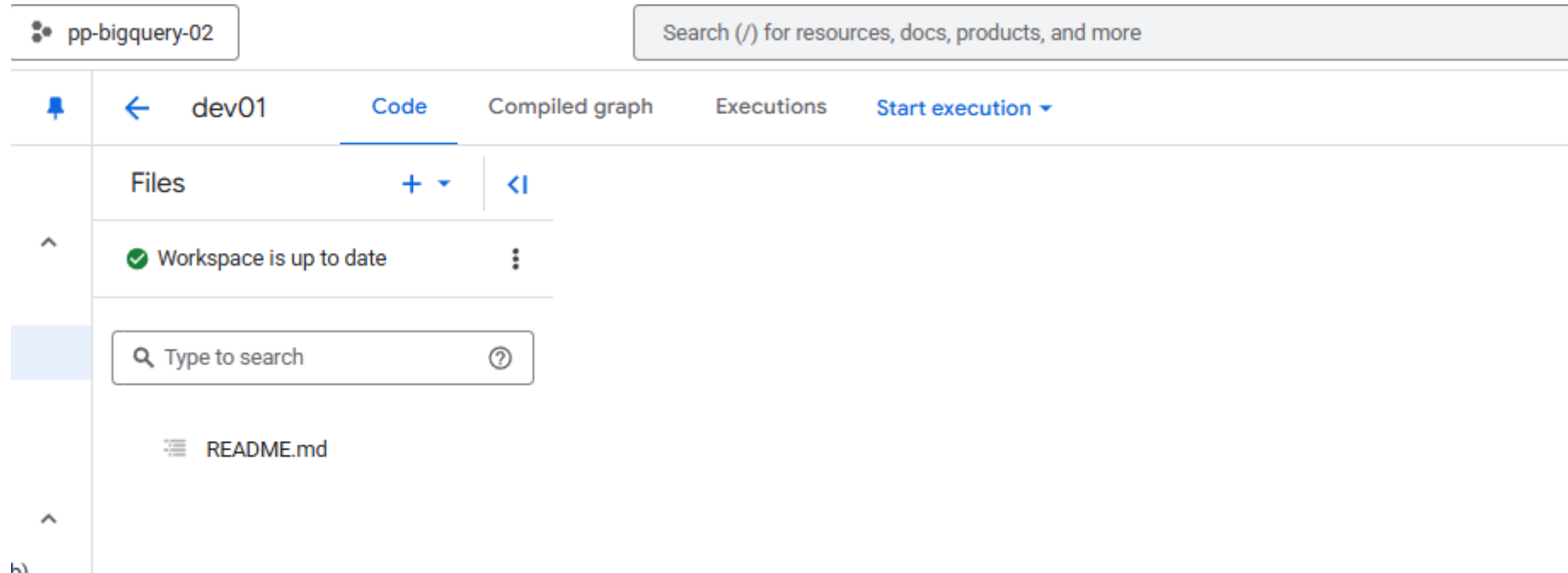
Override the target project ID, table prefix, and schema suffix settings for manual executions of all workspaces in the repository. The default settings are stored in `workflow_settings.yaml`. Learn more about [workspace compilation overrides](#).

No workspace compilation overrides.

Dataform – Create a development workspace



Dataform – Create a development workspace



Dataform – Initialize workspace

←

dev01

Code

Compiled graph

Executions

Start execution ▾

Files

+ ▾

<|

✓ Workspace is up to date

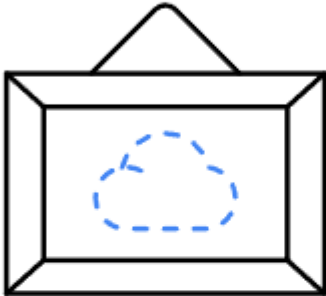
⋮

🔍 Type to search

?

☰

README.md

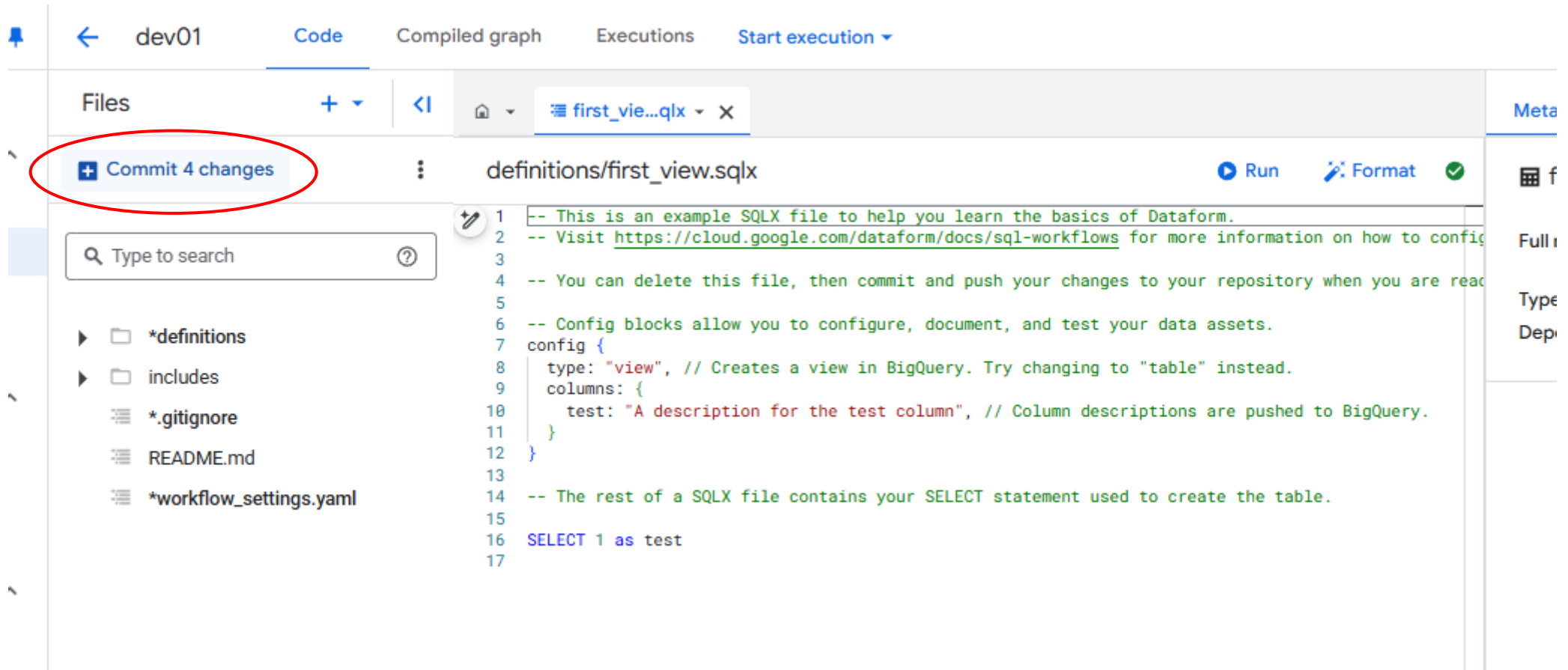


Initialize your Dataform workspace

This appears to be an empty Dataform workspace. To start developing with Dataform, first initialize your project with the required configuration files.

Initialize workspace

Dataform – Commit Changes



The screenshot displays the Dataform web interface for a project named 'dev01'. The 'Code' tab is active, showing a file named 'first_view.sqlx' in the 'definitions/' directory. A red circle highlights the 'Commit 4 changes' button in the top left of the code editor area. The file content is a SQLX file with comments and a configuration block. The left sidebar shows a file explorer with folders like '*definitions' and '*includes', and files like '*.gitignore', 'README.md', and '*workflow_settings.yaml'. The right sidebar shows a 'Meta' panel with a 'Full' view option.

dev01 Code Compiled graph Executions Start execution ▾

Files + ▾ <| 🏠 ▾ first_view...qlx ▾ ×

Commit 4 changes ⋮ definitions/first_view.sqlx ▶ Run ▶ Format ✓

🔍 Type to search (?)

- ▶ 📁 *definitions
- ▶ 📁 includes
- 📄 *.gitignore
- 📄 README.md
- 📄 *workflow_settings.yaml

```
1 -- This is an example SQLX file to help you learn the basics of Dataform.
2 -- Visit https://cloud.google.com/dataform/docs/sql-workflows for more information on how to config
3
4 -- You can delete this file, then commit and push your changes to your repository when you are ready
5
6 -- Config blocks allow you to configure, document, and test your data assets.
7 config {
8   type: "view", // Creates a view in BigQuery. Try changing to "table" instead.
9   columns: {
10     test: "A description for the test column", // Column descriptions are pushed to BigQuery.
11   }
12 }
13
14 -- The rest of a SQLX file contains your SELECT statement used to create the table.
15
16 SELECT 1 as test
17
```

Meta

Full

Type

Dep

Dataform – Commit Changes

bigquery-02

← dev01

Files

+ Commit 4 changes

🔍 Type to search

▶

📁

*definitions

▶

📁

includes

📄

*.gitignore

📄

README.md

📄

*workflow_setti

New commit

Select files to be committed. If you don't select any files, all files will be committed.

Filter

Enter property name or value

?

<input checked="" type="checkbox"/>	File state	Filename	File path ↑	Show diff
<input checked="" type="checkbox"/>	Added	.gitignore	/	>
<input checked="" type="checkbox"/>	Added	workflow_settings.yaml	/	>
<input checked="" type="checkbox"/>	Added	first_view.sqlx	definitions/	>
<input checked="" type="checkbox"/>	Added	second_view.sqlx	definitions/	>

Please enter a commit message:

Add a commit message *

Initial Commit

//

Commit 4 files

Cancel

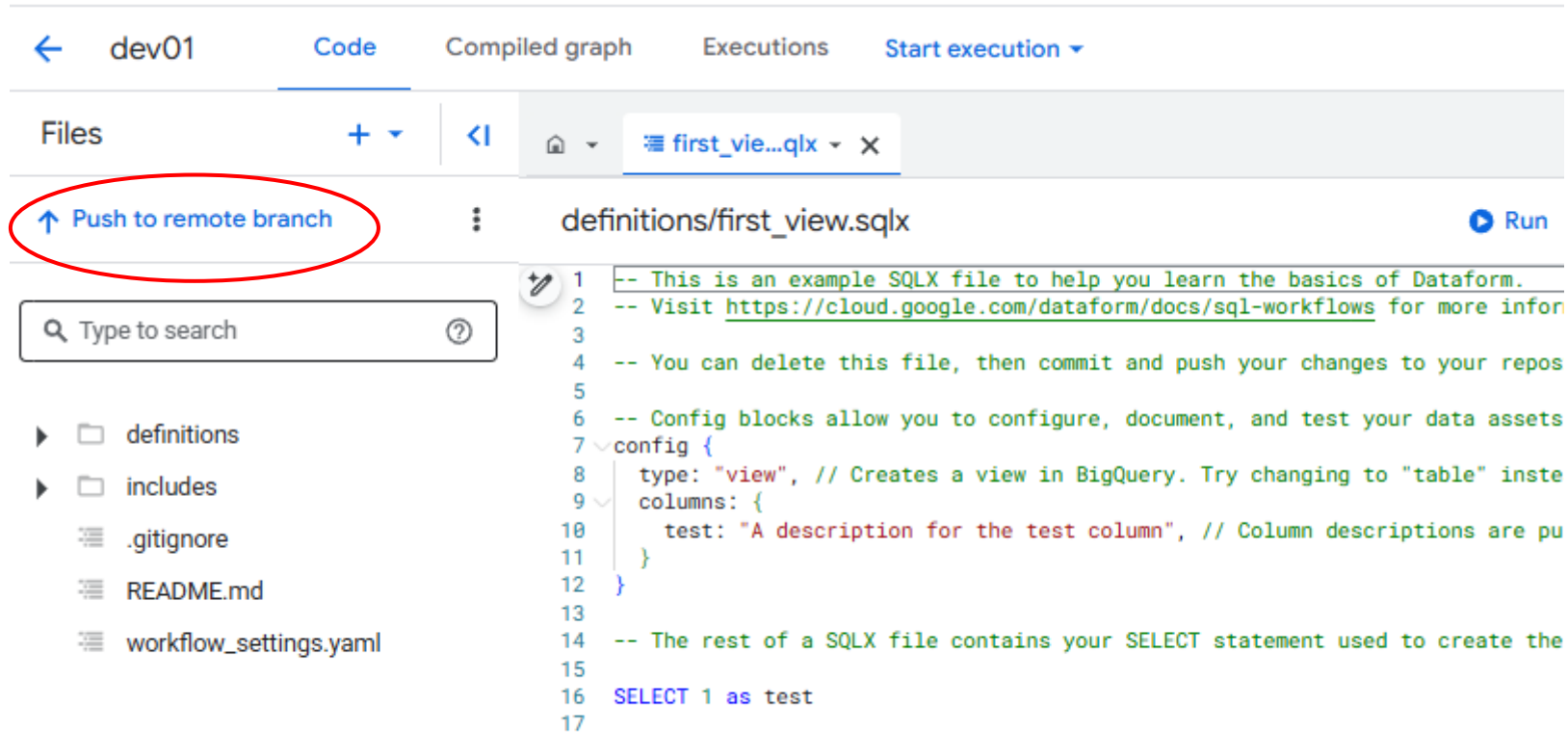
.gitignore

1 -

1+node_modules/

2+

Dataform – Push to remote branch



The screenshot displays the Dataform web interface for a project named 'dev01'. The 'Code' tab is selected, showing a file explorer on the left and a code editor on the right. The file explorer lists the following files and folders: 'definitions', 'includes', '.gitignore', 'README.md', and 'workflow_settings.yaml'. The 'definitions' folder is expanded, showing the file 'first_view.sqlx'. The code editor displays the content of 'first_view.sqlx', which is a SQLX file. The file contains comments and a configuration block for a view in BigQuery. The 'Push to remote branch' button is circled in red.

dev01 Code Compiled graph Executions Start execution ▾

Files + ▾ <I

first_vie...qlx

↑ Push to remote branch

Type to search ?

definitions

includes

.gitignore


README.md




workflow_settings.yaml




definitions/first_view.sqlx Run


```
1 -- This is an example SQLX file to help you learn the basics of Dataform.
2 -- Visit https://cloud.google.com/dataform/docs/sql-workflows for more information.
3
4 -- You can delete this file, then commit and push your changes to your repository.
5
6 -- Config blocks allow you to configure, document, and test your data assets
7 config {
8   type: "view", // Creates a view in BigQuery. Try changing to "table" instead.
9   columns: {
10     test: "A description for the test column", // Column descriptions are pushed to the BigQuery table.
11   }
12 }
13
14 -- The rest of a SQLX file contains your SELECT statement used to create the table.
15
16 SELECT 1 as test
17
```


Dataform – Confirm pull on GitHub






 dataform01 Public




 Pin  Watch 0  Fork 0





 main  2 Branches  0 Tags



 Add file

 Code

About
No description, website or files provided.
 Readme
 Activity
 0 stars
 0 watching
 0 forks

 nagabhushan1 Merge pull request #1 from nagabhushan1/dev01  7974b17 · now  3 Commits

 definitions	Initial Commit	2 minutes ago
 .gitignore	Initial Commit	2 minutes ago
 README.md	Initial commit	22 minutes ago
 workflow_settings.yaml	Initial Commit	2 minutes ago

 README 

dataform01

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)




Dataform – Service Account IAM Roles

Service account: `service-165926949701@gcp-sa-dataform.iam.gserviceaccount.com` | Project: `My GCP Course Project`

Edit access to "My GCP Course Project"

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. [Learn more](#)

<div>Role</div> <div>BigQuery Admin</div> <div>Administer all BigQuery resources and data</div>	<div>IAM condition (optional) ?</div> <div>+ Add IAM condition</div>	
<div>Role</div> <div>BigQuery Job User</div> <div>Access to run jobs</div>	<div>IAM condition (optional) ?</div> <div>+ Add IAM condition</div>	
<div>Role</div> <div>Dataform Service Agent</div> <div>Gives permission for the Dataform API to access a secret from Secret Manager</div>	<div>IAM condition (optional) ?</div> <div>+ Add IAM condition</div>	
<div>Role</div> <div>Secret Manager Admin</div> <div>Full access to administer Secret Manager resources.</div>	<div>IAM condition (optional) ?</div> <div>+ Add IAM condition</div>	