Automated Deployment (CICD) of SnapLogic assets with GitHub

Introduction

This guide is a reference document for the deployment of SnapLogic assets to a GitHub repository. It also includes sample YAML code for a GitHub Actions workflow which can be used to automate the deployment of assets across Environments (Dev -> Stg / Stg -> Prod, etc.)

This guide is targeted towards SnapLogic *Environment Administrators* (Org Administrators) and users who are responsible for the deployment of SnapLogic assets / Release management operations.

Section B covers automated deployment with GitHub Actions, and *Section A* illustrates a manual deployment flow using the Manager interface.

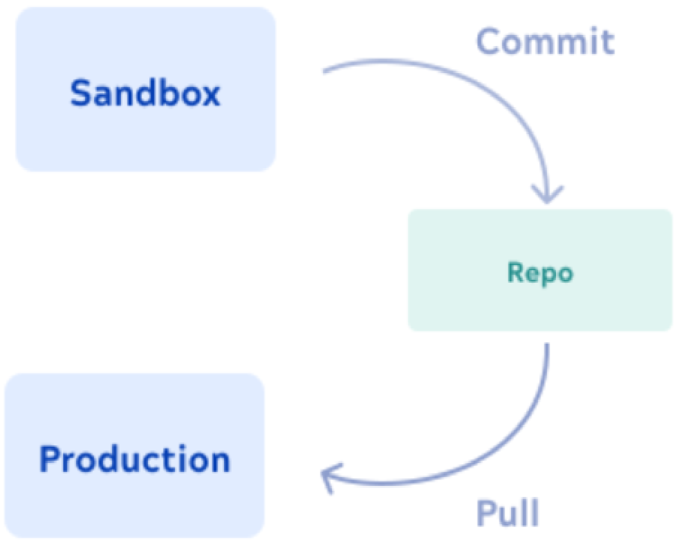
Author:

Ram Bysani
SnapLogic Enterprise Architecture team

SnapLogic Git Integration

Git Integration allows you to track, update, and manage versions of SnapLogic assets using the graphical interface or the public APIs. The following asset types can be tracked in a GitHub repository:

Accounts
Files
Pipelines
Tasks



Git model

A) Asset deployment across environments - an example

The example in this document illustrates a sample deployment of SnapLogic assets from the ***Dev*** environment (org) to the ***Prod*** environment (org). A similar methodology can be adopted to deploy assets from Dev -> Stg -> Prod environments. The environments should be configured for Git integration with GitHub. Please refer to the steps in the documentation.

Git Integration

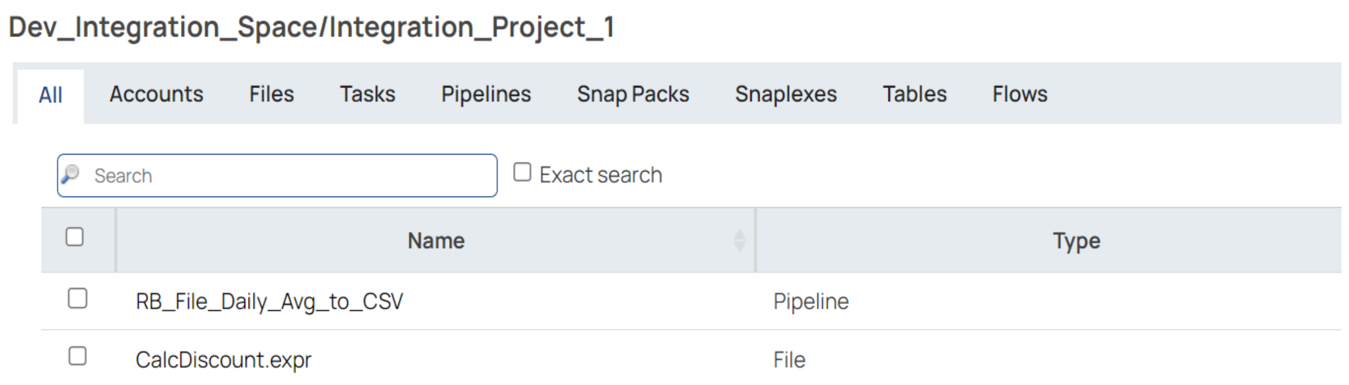
Git operations

The assets in this example are tracked at a project space level, i.e. one *Project Space in Dev* is associated with a single branch in the GitHub repository. A **single** GitHub repository is used to maintain the branches for Dev, Stg, Prod, etc. Repository branches can also be deleted and re-created for specific deployment needs.

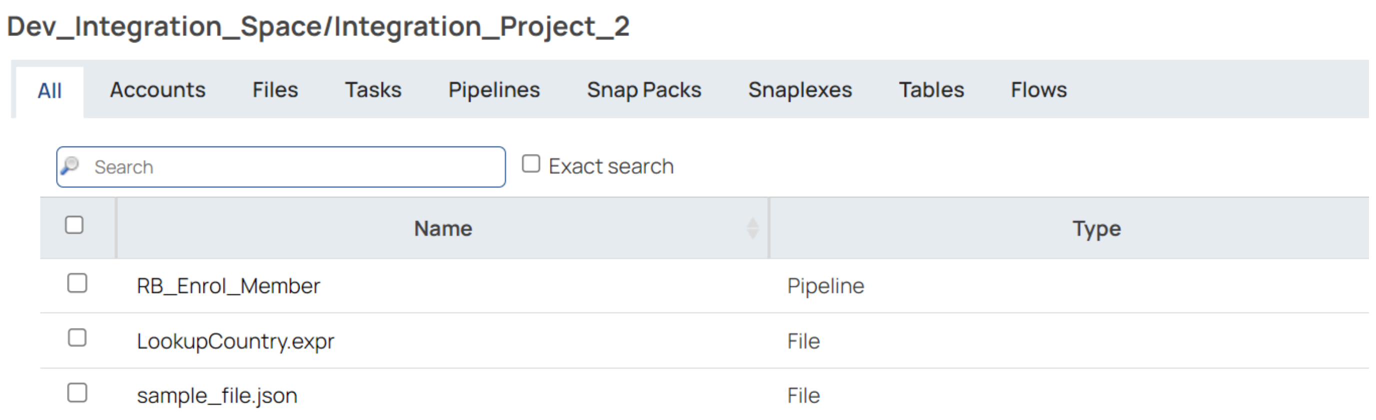
New / Modified Assets in the Dev Environment

Project Space: *Dev_Integration_Space* with the below project folders having SnapLogic assets.

Integration_Project_1, Integration_Project_2, share


Dev_Integration_Space/Integration_Project_1

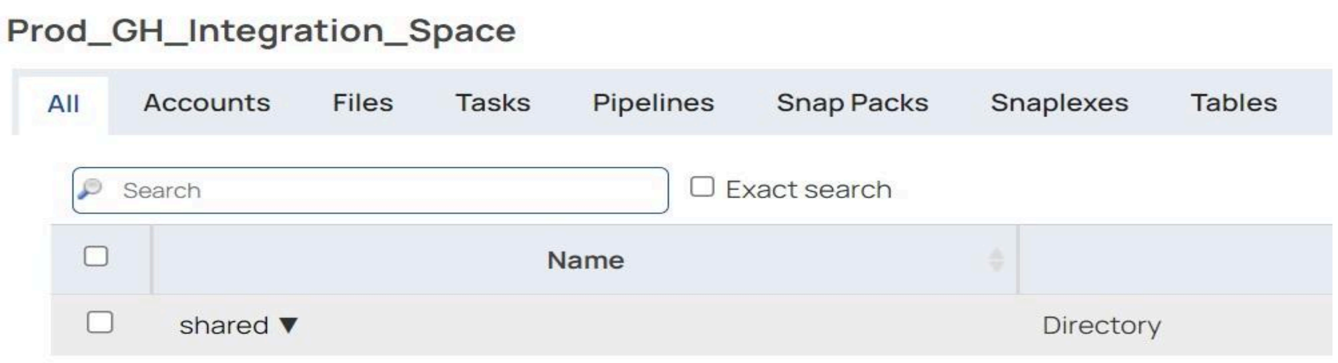
Name	Type
RB_File_Daily_Avg_to_CSV	Pipeline
CalcDiscount.expr	File


Dev_Integration_Space/Integration_Project_2

Name	Type
RB_Enrol_Member	Pipeline
LookupCountry.expr	File
sample_file.json	File

Prod Environment

We have already defined an empty project space named *Prod_GH_Integration* in the Prod env. This step can also be done by using the SnapLogic public API Project APIs.



Prod_GH_Integration_Space

Name	
shared ▼	Directory

Define branches in the GitHub repository

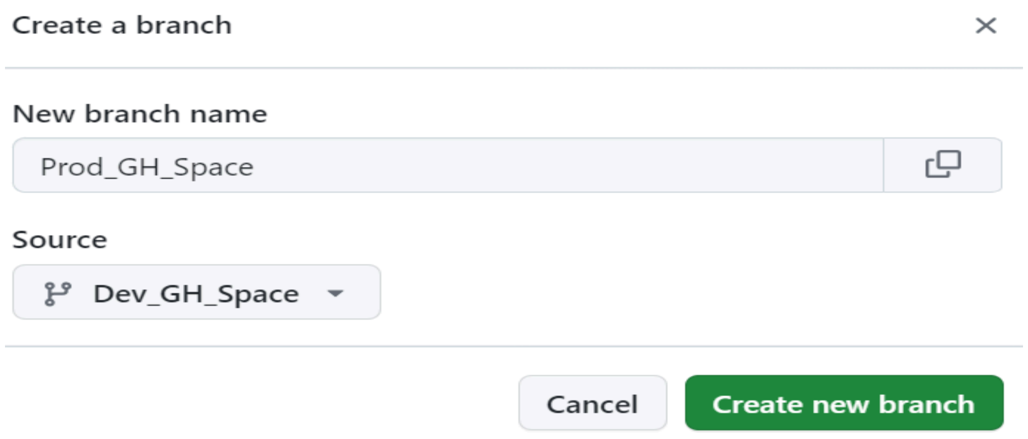
Create individual branches in the GitHub repository for the Dev and Prod project space assets. You can choose the **main** branch as the default branch while creating *Dev_GH_Space*. Choose the *Dev_GH_Space* branch as the source when creating the *Prod_GH_Space* branch.

Each branch in the GitHub repository corresponds to a Project Space in SnapLogic.

e.g:

Dev_GH_Space

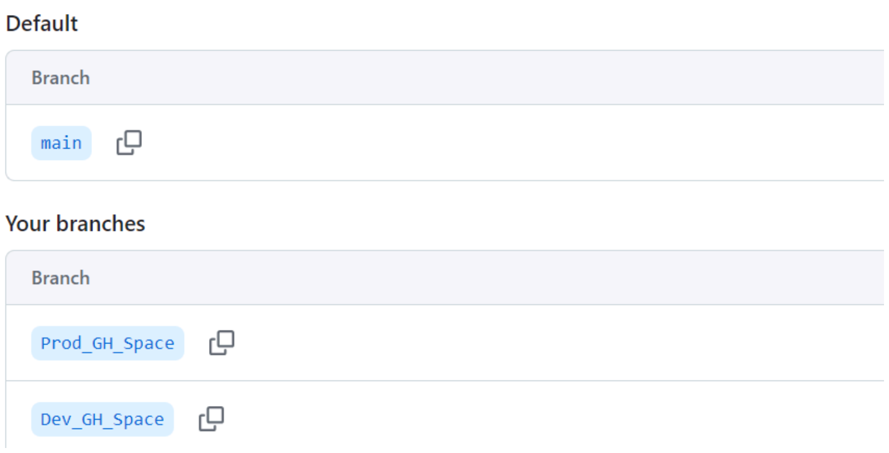
Prod_GH_Space



Create a branch

New branch name: Prod_GH_Space

Source: Dev_GH_Space



Default

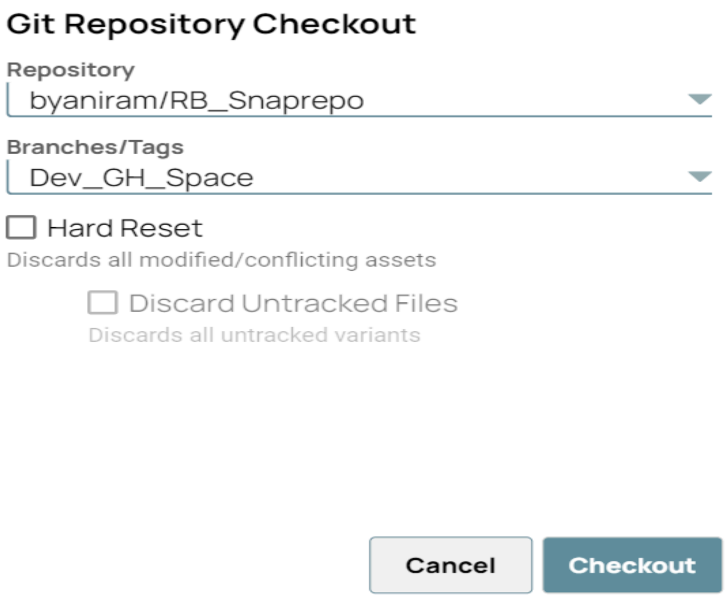
Branch
main

Your branches

Branch
Prod_GH_Space
Dev_GH_Space

Commit Dev assets to GitHub

Connect to the Dev (source) environment in the SnapLogic Manager interface, and navigate to the project space named *Dev_GH_Integration_Space.* Right click and select *Git Repository Checkout.* Choose the Git repository branch *Dev_GH_Space.*



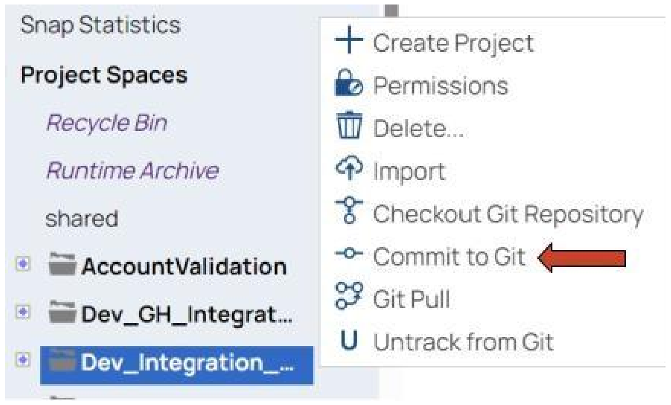
Git Repository Checkout

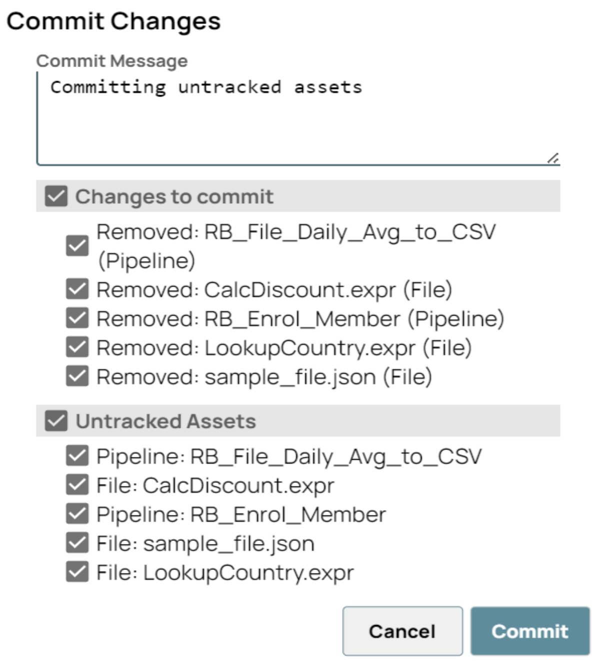
Repository: byaniram/RB_Snaprepo

Branches/Tags: Dev_GH_Space

☐ Hard Reset
Discards all modified/conflicting assets

☐ Discard Untracked Files
Discards all untracked variants





Commit Changes

Commit Message

```
Committing untracked assets
```

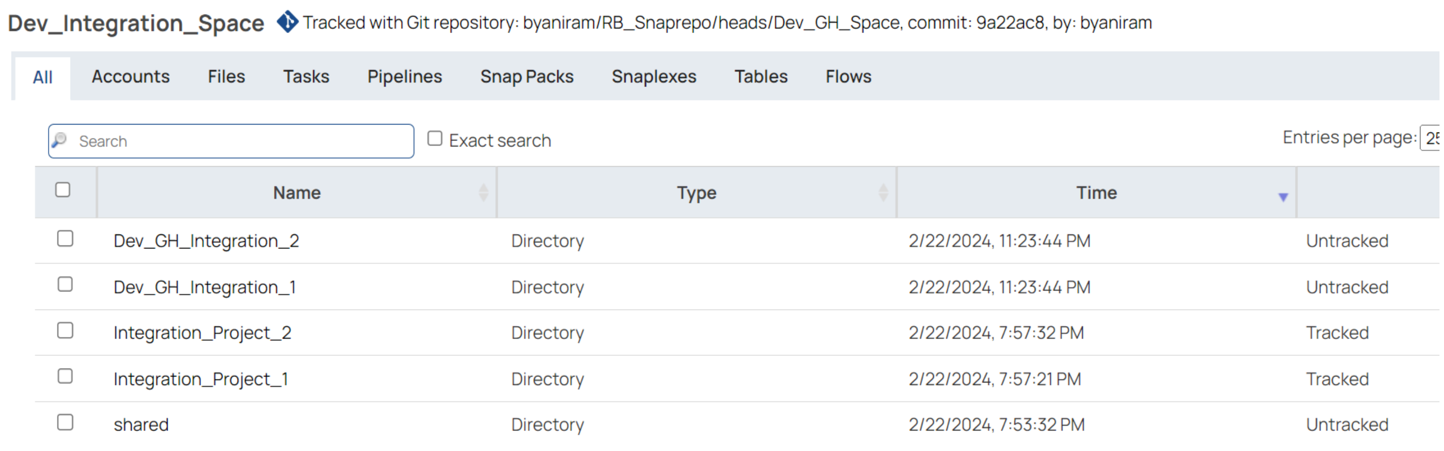
☑ Changes to commit

- ☑ Removed: RB_File_Daily_Avg_to_CSV (Pipeline)
- ☑ Removed: CalcDiscount.expr (File)
- ☑ Removed: RB_Enrol_Member (Pipeline)
- ☑ Removed: LookupCountry.expr (File)
- ☑ Removed: sample_file.json (File)

☑ Untracked Assets

- ☑ Pipeline: RB_File_Daily_Avg_to_CSV
- ☑ File: CalcDiscount.expr
- ☑ Pipeline: RB_Enrol_Member
- ☑ File: sample_file.json
- ☑ File: LookupCountry.expr

You can see that the Git status has changed to *Tracked* for all assets under the child projects. Note that some assets appear with status *Untracked* as these were already existing in the *main* branch. These assets would not be committed to the Git repository.



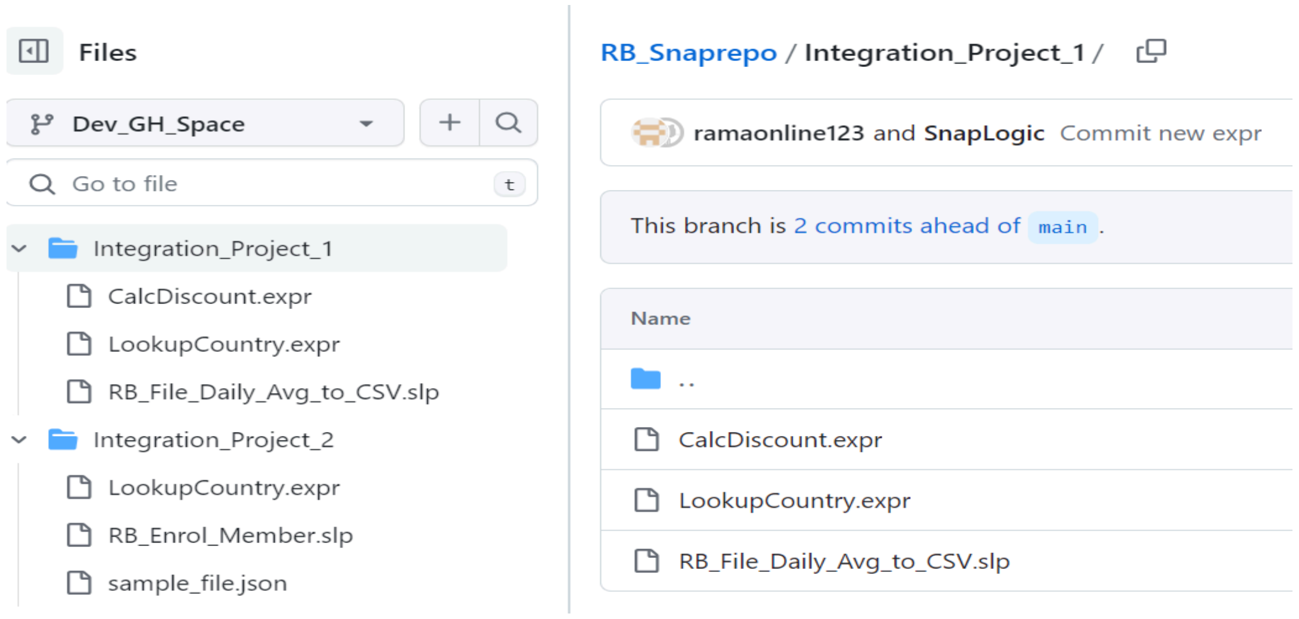
Dev_Integration_Space Tracked with Git repository: byaniram/RB_Snaprepo/heads/Dev_GH_Space, commit: 9a22ac8, by: byaniram

	Name	Type	Time	
☐	Dev_GH_Integration_2	Directory	2/22/2024, 11:23:44 PM	Untracked
☐	Dev_GH_Integration_1	Directory	2/22/2024, 11:23:44 PM	Untracked
☐	Integration_Project_2	Directory	2/22/2024, 7:57:32 PM	Tracked
☐	Integration_Project_1	Directory	2/22/2024, 7:57:21 PM	Tracked
☐	shared	Directory	2/22/2024, 7:53:32 PM	Untracked

Notice the tracking message with the branch name and commit id next to the project space name:

Tracked with Git repository: byaniram/RB_Snaprepo/heads/Dev_GH_Space, commit: 9a22ac8

Connect to the GitHub repository and verify the commit status for the branch *Dev_GH_Space.*

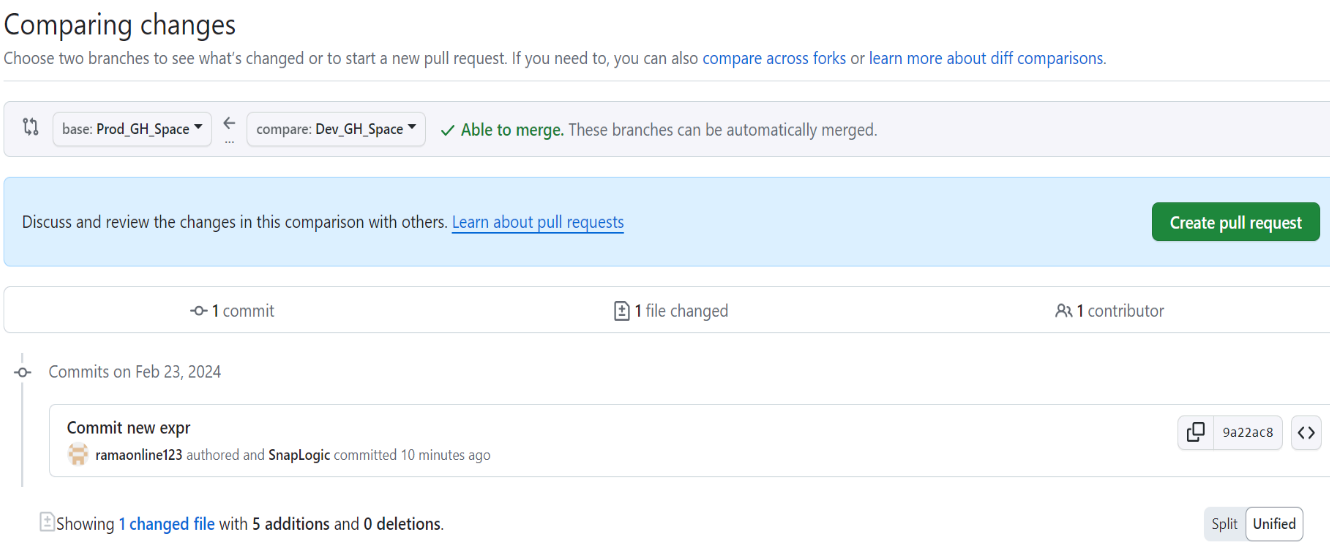


ramaonline123 and SnapLogic Commit new expr

This branch is 2 commits ahead of `main`.

Create Pull Request in GitHub

At this step, you would need to create a Pull Request in GitHub. Choose *Prod_GH_Space* as the **base** branch, and *Dev_GH_Space* as the **compare** branch, and create the Pull request. This action would merge the assets contained in the *Dev_GH_Space* branch into the *Prod_GH_Space* branch.


Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base: Prod_GH_Space ← compare: Dev_GH_Space ✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. Learn about pull requests

Create pull request

1 commit | 1 file changed | 1 contributor

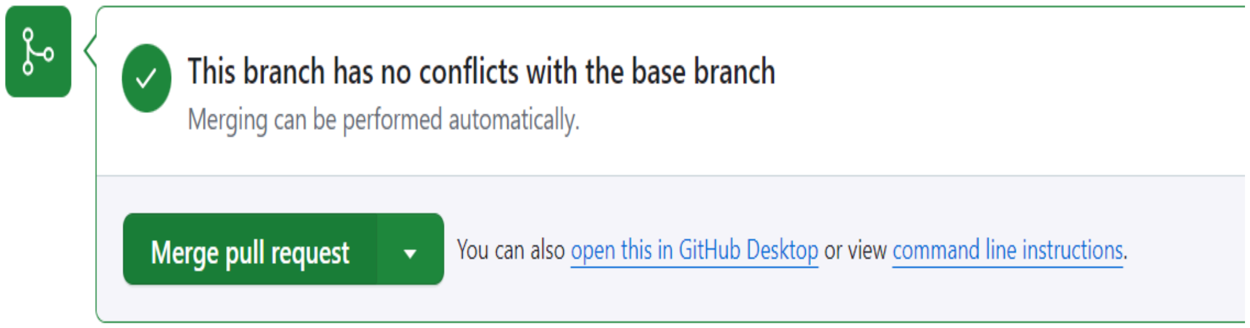
Commits on Feb 23, 2024

Commit new expr

ramaonline123 authored and SnapLogic committed 10 minutes ago — 9a22ac8

Showing 1 changed file with 5 additions and 0 deletions.

Split | Unified

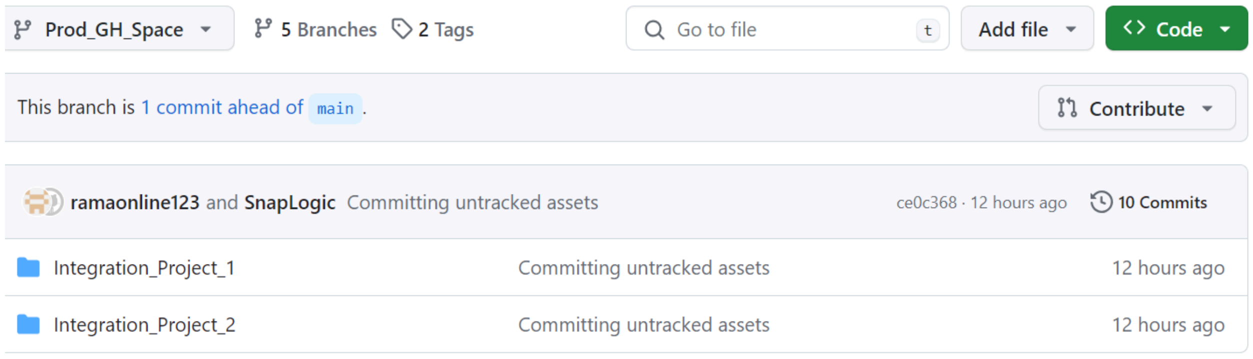

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Connect to the GitHub repository and verify the commit status for the branch *Prod_GH_Space.* The assets have now been committed to the Prod environment and are tracked in the GitHub repository under the branch - *Prod_GH_Space.*

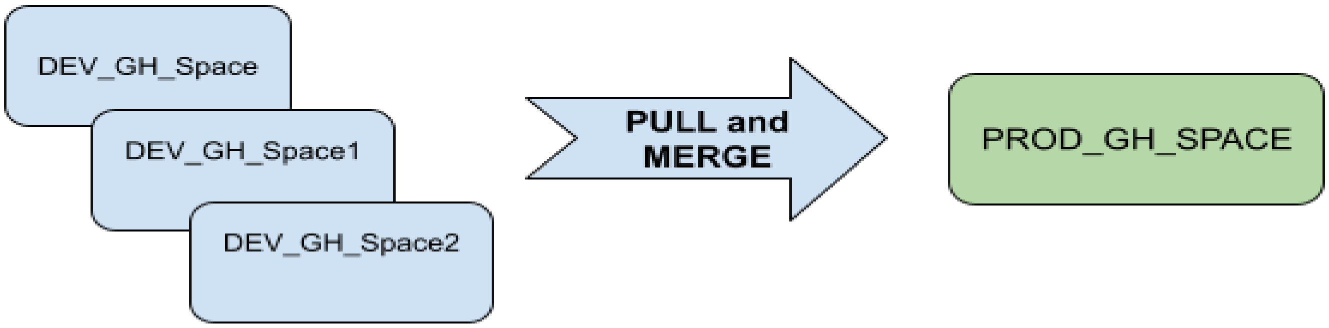

Prod_GH_Space | 5 Branches | 2 Tags | Go to file | Add file | Code

This branch is 1 commit ahead of `main`. Contribute

ramaonline123 and SnapLogic Committing untracked assets — ce0c368 · 12 hours ago — 10 Commits

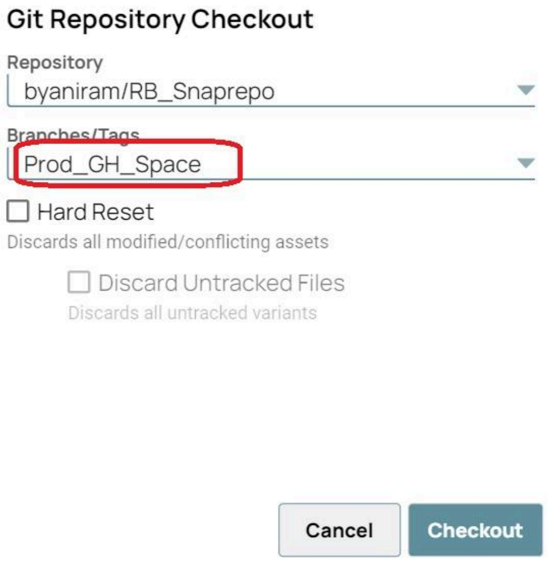
Integration_Project_1	Committing untracked assets	12 hours ago
Integration_Project_2	Committing untracked assets	12 hours ago

It is also possible to merge and pull from additional branch(es) into a single *Prod_GH_Space* if you have a need for it. You would need to repeat the Pull / Merge process as above with the **base** branch being Prod_GH_Space, and the **compare** branch being one of Dev_GH_Space, Dev_GH_Space_1, or Dev_GH_Space_2.



Pulling / Committing assets into the Prod Environment

Connect to the Prod (target) environment in the SnapLogic Manager interface, and navigate to the project space named *Prod_GH_Integration_Space.* Right click and select *Git Repository Checkout.* Choose the Git repository branch *Prod_GH_Space.*



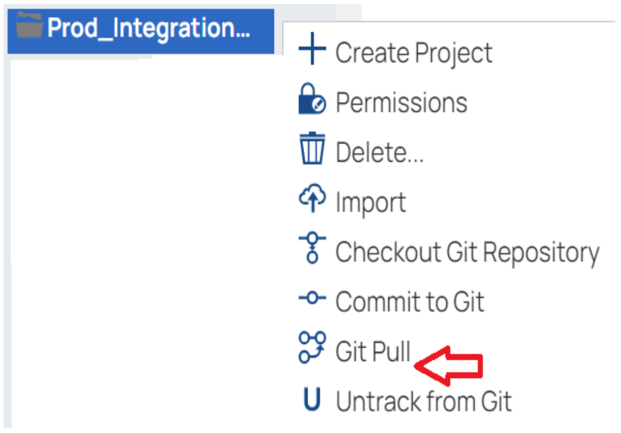
Git Repository Checkout

Repository: byaniram/RB_Snaprepo

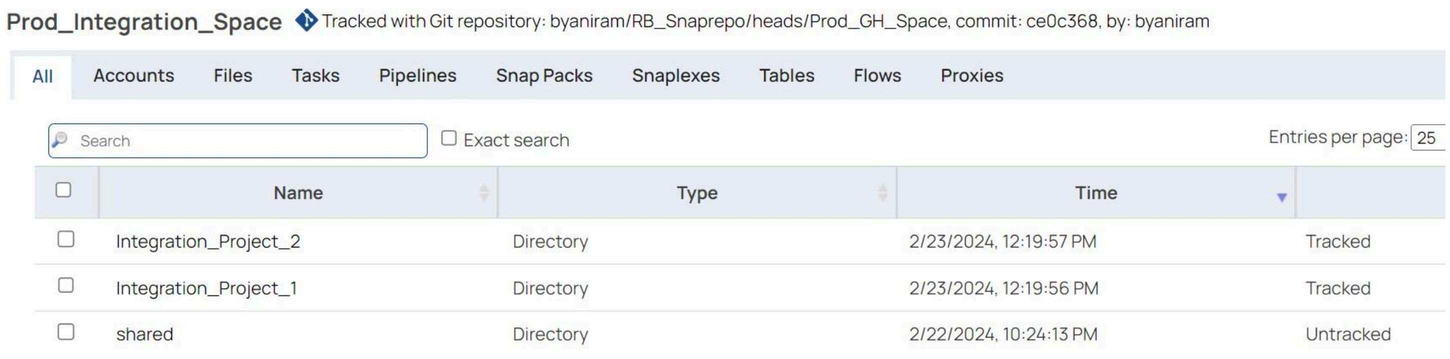
Branches/Tags: Prod_GH_Space

Hard Reset
Discards all modified/conflicting assets
Discard Untracked Files
Discards all untracked variants
Cancel
Checkout

Choose *Git Pull* to pull the assets into the Project space.



The assets from the *Dev_Integration_Space* project space *of the Dev* environment are deployed to the *Prod_Integration_Space* project space of the *Prod* environment.



Prod_Integration_Space Tracked with Git repository: byaniram/RB_Snaprepo/heads/Prod_GH_Space, commit: ce0c368, by: byaniram

	Name	Type	Time	
☐	Integration_Project_2	Directory	2/23/2024, 12:19:57 PM	Tracked
☐	Integration_Project_1	Directory	2/23/2024, 12:19:56 PM	Tracked
☐	shared	Directory	2/22/2024, 10:24:13 PM	Untracked

Notice the tracking message with the branch name and commit id next to the project space name:

Tracked with Git repository: byaniram/RB_Snaprepo/heads/Prod_GH_Space, commit: ce0c368

For subsequent deployments of changed assets, you would first do a *Commit to Git* for the project space in the SnapLogic *Dev* environment, followed by the above steps. Changed assets would be visible with a Git status of '*Tracked, Modified locally*' in the SnapLogic Manager.

☐	RB_Enrol_Member	Pipeline	2/22/2024, 8:07:50 PM	Tracked, Modified locally

B) Deployment Automation using a GitHub Actions Workflow

Actions workflow YAML sample

A *GitHub Actions* workflow can be used to automate the deployment of assets across SnapLogic environments (such as Dev to Stg, Stg to Prod, etc.). A workflow is a configurable automated process made up of one or more jobs. You must create a YAML file to define your workflow configuration.

Here’s a complete YAML file for the Dev -> Prod deployment example described in *Section A* above. The complete YAML file is attached for your reference. Please create a new Workflow from the *Actions* tab, and paste the contents of the file in the workflow editor and commit changes.

```
# Actions workflow for automated deployment of SnapLogic assets

name: SnapLogic CICD Sample
on:
  push:
    branches:
      - Dev_GH_Space
# Uncomment the below line if you need to execute the workflow manually.
# workflow_dispatch:
jobs:
  pull_merge_branches:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4
      - name: Merge Dev to Prod
```

```
        uses: devmasx/merge-branch@master
        with:
          type: now
          from_branch: Dev_GH_Space
          target_branch: Prod_GH_Space
          github_token: ${{ secrets.ACTIONS_TOKEN }}

      - name: Checkout project assets to Prod project space
        run: |
         curl -s -X POST \
${{vars.SNAP_URL}}/api/1/rest/public/project/pull/${{vars.SNAP_ORG}}/${{vars.PROJECT_SPACE}} \
        -H "Content-Type:application/json" -H "Authorization:Basic ${{secrets.BASE64_TOKEN}}" \
        -d '{"use_theirs":"true"}'
```

Please refer to the GitHub documentation for information related to Workflow usage and syntax:

GitHub Workflows
Workflow syntax

The following table provides clarification on certain aspects of the sample workflow for better understanding.

Section	Comments
runs-on: ubuntu-latest	*runs-on* defines the runner (type of machine) to use to run the job. ubuntu-latest specifies a GitHub hosted runner image. GitHub hosted runners
uses: actions/checkout@v4	*checkout* is an action which is available in the GitHub marketplace. This action checks out the repository for use. v4 is the version number of the action.

	https://github.com/marketplace/actions/checkout
uses: devmasx/merge-branch@master	*merge-branch* is an action from the GitHub marketplace. This action runs a Git merge operation. https://github.com/marketplace/actions/merge-branch It also requires you to define a personal access token (classic) under Developer Settings -> Personal access tokens. Select both the repo and workflow checkboxes.

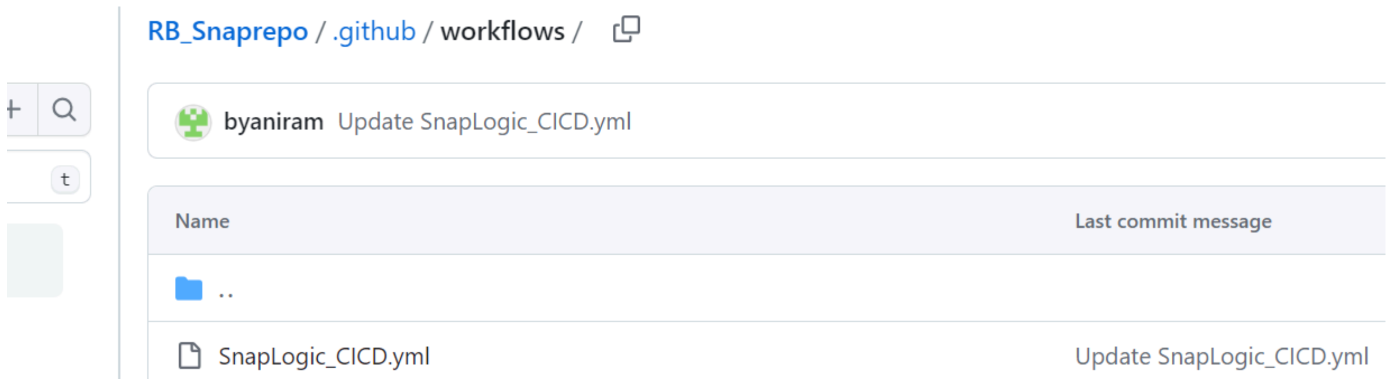
`curl -s -X POST \`
`${{vars.SNAP_URL}}/api/1/rest/public/project/pull/${{vars.SNAP_ORG}}/${{vars.PROJECT_SPACE}} \`
`-H "Content-Type:application/json" -H "Authorization:Basic ${{secrets.BASE64_TOKEN}}" \`
`-d '{"use_theirs":"true"}'`
	This is a CURL command that executes the SnapLogic public API to pull the latest project files from Git.
	See Pull the latest project files from Git
	The referenced variables are defined on the GitHub repository under Settings -> *Secrets and variables* -> *Actions*. The **vars** context is used to reference those variables. (e.g. SNAP_ORG, PROJECT_SPACE)
	You can also define encrypted Secrets for sensitive data and reference them using the **secrets** context as in the example.
	(e.g. BASE64_TOKEN has the base64 encoded string for username and password)
	Workflow Variables

Table 1.0 - Workflow Actions

Workflow execution

The above Actions workflow will be automatically executed whenever there is a “Push” / “Git Commit” operation to the Dev_GH_Space branch. i.e. whenever a commit is done from the **Dev** SnapLogic environment project space.

The workflow will execute the pull-merge operation to the Prod_GH_Space branch, and pull the latest project assets into the **Prod** SnapLogic environment. The YAML file must be created under the .github/workflows folder of the *Dev_GH_Space* branch in the GitHub repository.

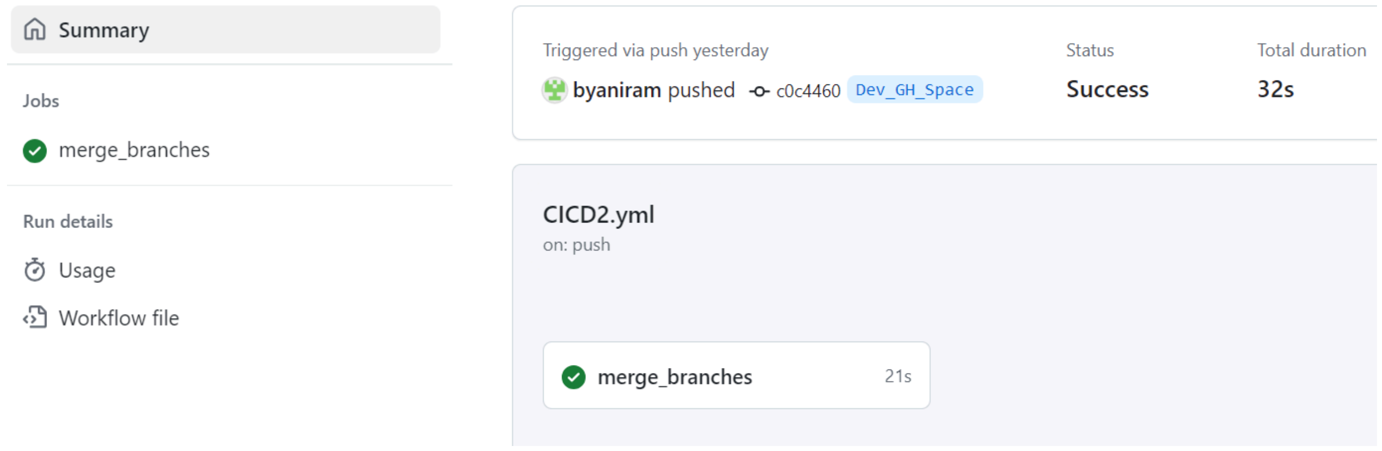


RB_Snaprepo / .github / workflows /

byaniram Update SnapLogic_CICD.yml

Name	Last commit message
..	
SnapLogic_CICD.yml	Update SnapLogic_CICD.yml

The workflow run status will be visible under the *Actions* tab.



Triggered via push yesterday
byaniram pushed c0c4460 Dev_GH_Space
Status: Success
Total duration: 32s

CICD2.yml
on: push

merge_branches 21s

Note:

If you wish to manually execute the pull-merge post code review, then you can uncomment the two lines in the script to enable *workflow_dispatch, and execute the Actions* workflow manually from the *Actions* tab on GitHub.

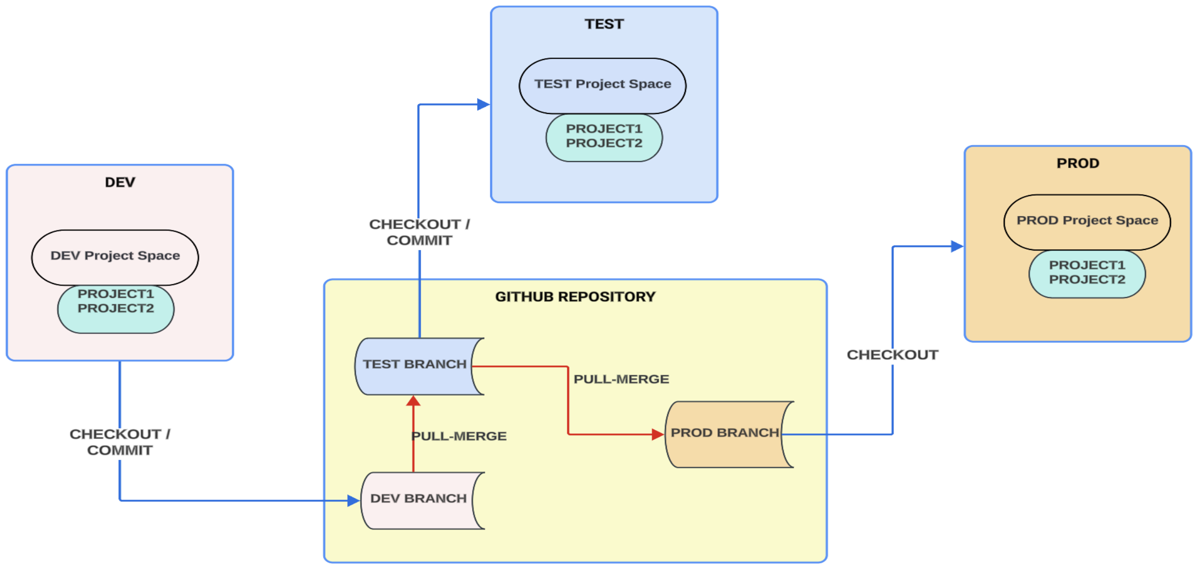
```
# Uncomment the below line if you need to execute the workflow manually.

# workflow_dispatch:
```

You can edit and modify the YAML file as per your requirements. Subsequent commits and deployments from Dev->Prod can be automated similarly.

Comments	Action
SnapLogic Dev environment Manager Interface **Asset -> Add to repository. Ensure status shows *Tracked*** **Project Space -> Commit to Git**	**Developer commits new code or updates assets in the Dev environment to the GitHub repository**
Create a new Pull Request on GitHub, and merge the newly committed assets by choosing the *Prod* branch as the base, and the *Dev* branch as the compare branch.	**Create and merge Pull Request**
SnapLogic Prod environment Manager Interface **Project Space -> *Git Pull***	**Pull the updated assets into the Prod environment**

Table 2.0 - Steps for subsequent / future asset deployment

Deployment flow (Dev->Test->Prod)



Deployment flow: Dev->Test->Prod

Note: *Future versions of this document will cover additional deployment scenarios. Please post your comments on the article.*