



HIN GitHub

ETL Offshore Team Training

March 18, 2025

CONFIDENTIAL – Contains proprietary information. Not intended for external distribution.

Agenda

1. **General GitHub Orientation**
2. **GitHub Actions + HIN Workflows**
3. **HIN GitHub standards**
4. **Questions**



General GitHub Orientation



CONFIDENTIAL – Contains proprietary information. Not intended for external distribution.

General GitHub Orientation

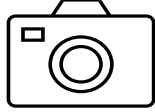
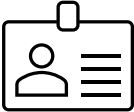
Git & GitHub

- **Git**
 - An open-source **Version Control System**
 - Keep track of changes to one or more files over time
 - Able to return to previous states if needed
 - Tracks **snapshots**, not file changes
 - <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>
- **GitHub**
 - Platform for storing files using Git
 - Web-based UI



General GitHub Orientation

Commits and Branches

- **Commit = Snapshot** 
 - All elements of the repo at that moment
 - Every commit is assigned a SHA
- **Branches = Commits we name** 
 - Automatically points to the most recent commit - **head**
 - Easy to review commit history
- **Important details**
 - GitHub doesn't track file changes, per se
 - Instead, it tracks **commit histories**
- **Change Process**
 - Create a feature branch, update/add files
 - Merge feature branch into long-lived branch

<https://agripongit.vincenttunru.com/>



General GitHub Orientation

Commit History

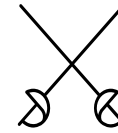
About commits

- **For branches**
 - All history for the commits with a given name
 - ***Especially helpful to confirm feature branches made from correct long-lived branch (prd/prod)***
 - Please don't reuse exact feature branch names
- **For files**
 - All commits associated with a file for a given branch
- **Repo commit (or file) search**
 - Searching commits
 - Finding files
- **PR commit search**
 - GitHub docs



General GitHub Orientation

Merge Conflicts



- **The Cause**
 - Different branches (**different commit histories**) change the same file
- **How to Prevent**
 - Coordinate work within the repository
 - Use the same feature branch for changes to a given file between releases to prd/prod
- **How to Fix**
 - Make sure feature branch was created properly
 - Review commit histories for associated files on relevant branches
 - Move conflicting changes to first branch
 - Revert earlier changes
 - Force reset branches (repo admin must do this)

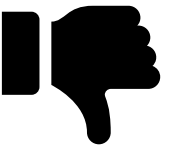
DO NOT RESOLVE CONFLICTS IN THE GUI



General GitHub Orientation

Merge Conflicts: Why not just resolve?

- **Reminders**
 - Git tracks **commits**
 - New change = New commit
- Resolving in the GUI does a special merge
 - The merge is done with additional commands you don't see that tell GitHub which version of the file to keep
 - In ALL cases you will **merge the entire commit history of the base branch into your feature branch**
 - This **breaks our HIN branching/merging pattern, causes problems in all future PRs using that feature branch**



General GitHub Orientation

Merge Conflicts: Tools to Investigate

- **Confirm feature branch made from prd/prod**
- **File commit histories**
- **PR commit search**
- **Move changes to common feature branch?**
- **Revert?**
 - What it is; what it isn't
 - When it helps
 - When it creates its own complications

When it isn't this simple:

- Ask for support
 - Aruna, Sai, Sudhir
 - Devi, Monica- we are happy to help!
- Rebase and/or Reset base branch



GitHub Actions + HIN Workflows



CONFIDENTIAL – Contains proprietary information. Not intended for external distribution.



GitHub Actions + HIN Workflows

Overview of GitHub Actions

GitHub Actions = CI/CD platform

- Components:
 - Trigger event
 - Workflow file
 - Jobs
 - Steps
 - Actions
 - Runner

Goal for HIN Workflows

- Centralized repo containing:
 - Reusable workflows
 - Composite actions
- Caller workflows in “functional” repos



GitHub Actions + HIN Workflows

HIN Workflows for Prod Support

Let's take a look at each of these:

- Bigquery DDL deployment for hotfixes
 - Triggered by merge to prod
 - Please also merge down to other branches
 - We are working on an automation for this
- GCS workflow to make changes within bucket(s)
 - Workflow dispatch (manual) trigger
- GCS workflow to move file from GitHub to bucket
 - Workflow dispatch (manual) trigger



GitHub Actions + HIN Workflows

Monitoring GitHub Action runs

- Actions tab
- Select Workflow of interest on the left
- Find and click into run of interest

The screenshot displays the GitHub Actions interface for the repository 'HCACloudDataEngineering'. The 'Actions' tab is selected in the top navigation bar. On the left sidebar, under 'All workflows', the 'BigQuery DDL Deployment' workflow is highlighted. The main area shows the workflow details for 'BigQuery DDL Deployment' (bigquery.yaml). It indicates there are 223 workflow runs. A light blue banner states: 'This workflow has a workflow_dispatch event trigger.' Below this, a table lists recent runs:

Run	Status	Branch	Actor	Event	Status	Branch	Actor
Merge pull request #286 from HCACloudDataEngineering/feature/lkp_payo...	Success	qa	PGE3843_hca	workflow_dispatch	4 days ago	1m 9s	...
Merge pull request #285 from HCACloudDataEngineering/feature/lkp_payo...	Success	dev	DGF3843_hca	workflow_dispatch	4 days ago	1m 18s	...



GitHub Actions + HIN Workflows

GitHub Action Errors

Click all the way into the workflow run to see detailed error messages



The screenshot displays the GitHub Actions interface for a workflow named "Call BigQuery DDL Reusable Workflow / Setup Build and Deploy DDLs". The workflow is shown as failed. The "Jobs" section lists two jobs: "Call BigQuery DDL Reusable Wor..." and "Setup Build and Deploy DDLs". The "Run details" section shows the "Bigquery Deploy Step" with a detailed error message. The error message indicates a "BigQueryDeployError" with the following details:

```
218 *****Running queries for tables*****
219 *****Running queries for procedures*****
220 *****Running queries for udfs*****
221 Exception
222
223 During handling of the above exception, another exception occurred:
224
225 Traceback (most recent call last):
226   File "/runner/_work/gcp-hin-ops-ddl/gcp-hin-ops-ddl/artifacts/deploy.py", line 150, in <module>
227     main()
228   File "/runner/_work/gcp-hin-ops-ddl/gcp-hin-ops-ddl/artifacts/deploy.py", line 145, in main
229     raise BigQueryDeployError(error_list)
230   __main__.BigQueryDeployError: 'Errors in SQLs'
231 *****Running queries for base views*****
232 Project_id: hca-hin-dev-cur-ops
233 Running Query: /runner/_work/gcp-hin-ops-ddl/gcp-hin-ops-ddl/artifacts/bigquery/auth_base_views/views/lkp_payor_id.sql
234 Job ID: 87c40f58-89e7-43d8-b295-a7f2597fa9fb
235 State: DONE
236 Errors: None
237 Project_id: hca-hin-dev-cur-ops
238 Running Query: /runner/_work/gcp-hin-ops-ddl/gcp-hin-ops-ddl/artifacts/bigquery/auth_base_views/views/lkp_payor_id.sql
239 *****Running queries for views*****
240 ##### SQLS ERRORS #####
241 /runner/_work/gcp-hin-ops-ddl/gcp-hin-ops-ddl/artifacts/bigquery/auth_base_views/views/lkp_payor_id.sql
242 404 Not found: Table hca-hin-dev-cur-ops:edwvf_base_views.lkp_payor_id was not found in location US; reason: notFound, message: Not found: Table hca-hin-dev-cur-ops:edwvf_base_views.lkp_payor_id was not found in location US
243
244 Location: US
245 Job ID: 26d6e506-107d-409c-b8c3-957855a9482e
246
```



HIN GitHub Standards & Processes



CONFIDENTIAL – Contains proprietary information. Not intended for external distribution.

HIN GitHub Standards

Branch Names: 3 Components

1. Branch types

- **feature**: new code addition
- **fix**: correct existing code
- **hotfix**: urgent correction, generally going directly to production
- **refactor**: restructure functional code to optimize
- **test**: experimentation, PoC, etc.

2. Ticket number

3. Brief description

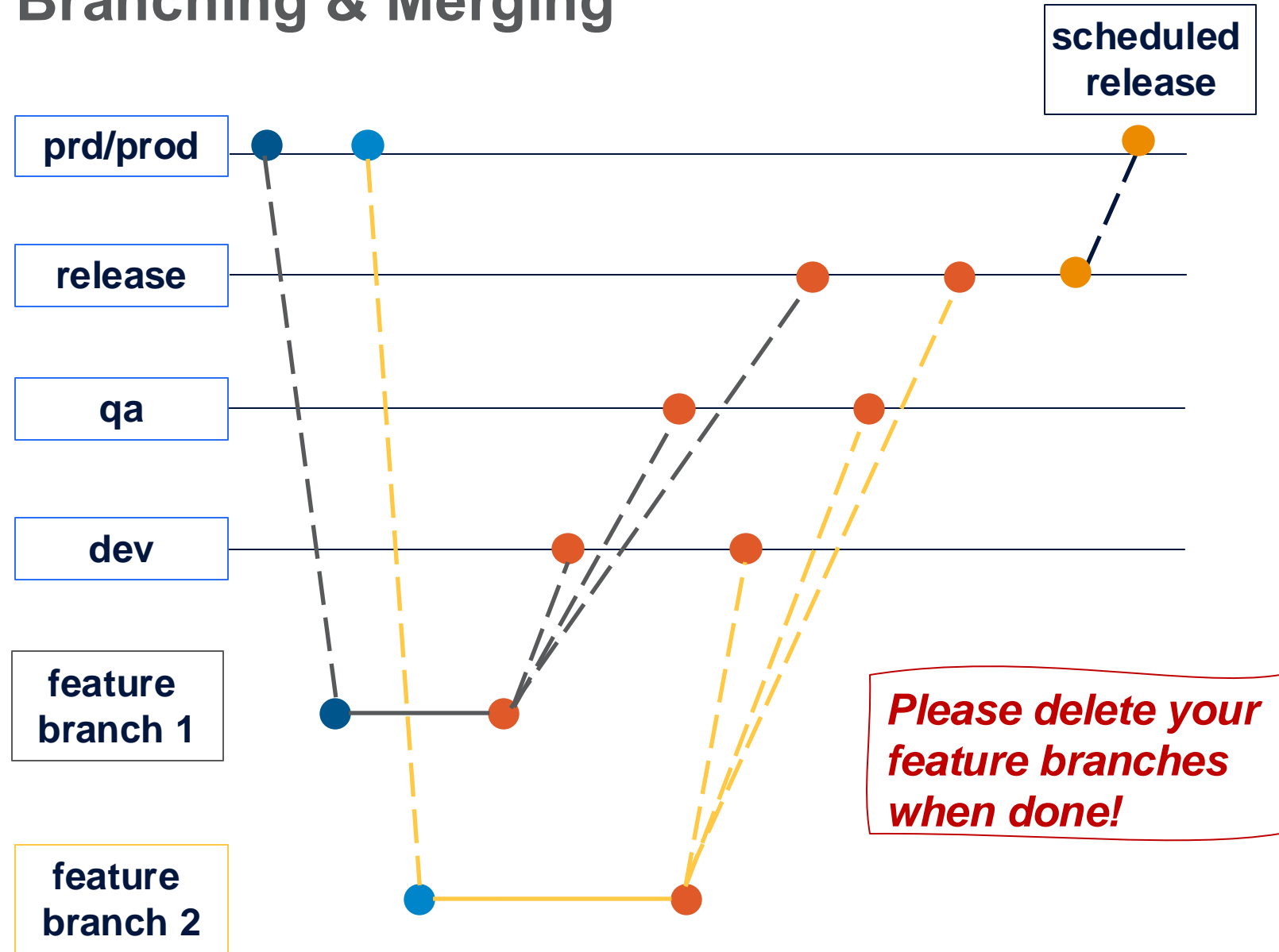
All feature branches MUST be created from
prd/prod

Example branch names:

- feature/TGCP-1234/new_workflow
- fix/TGCP-9876/correct_ingest_dag
- refactor/TGCP-4567/optimize_gsm_script

HIN GitHub Standards

Branching & Merging



HIN GitHub Standards

Branch Protections

Higher environments = More protection

- **dev branch**
 - PR required, no approval
- **qa branch**
 - PR required, peer approval
- **release branch**
 - PR required, code owner approval
- **prd/prod branch**
 - PR required, peer approval, code owner approval
 - Only admins/code owners can merge PR

On call/offshore production support processes currently being explored



HIN GitHub Standards

Pull Request Reviews

- **Developer**
 - Double check files involved
 - Is the **number of files** changed what you expect?
 - Are the files changed **the ones you expect**?
 - Review commit history in the PR
 - Should mostly be **commits made by the developer**
 - Any PR merges should ONLY be into release or prd/prod
 - **Complete relevant section of PR checklist**
 - Monitor any associated action runs
- **Reviewer**
 - Code review
 - Confirm all included files make sense
 - Look through PR commit history
 - SHOULD NOT see any PRs into dev or qa
 - SHOULD NOT see any merges from dev or qa
 - Confirm all previous action runs succeeded


Example of what you DO NOT normally want to see:

<https://github.com/HCACloudDataEngineering/gcp-hin-ops-ddl/pull/48>




HIN GitHub Standards

HIN GitHub SharePoint Site



HIN GitHub Resources & Processes



Welcome!

This is the place to be to learn all the ins and outs of working in HIN GitHub repositories.

Whether you're looking to strengthen your understanding of GitHub in general or just want to make sure you're following the right steps within an HIN GitHub repo, you will find what you need in the pages linked below.





Questions?

CONFIDENTIAL – Contains proprietary information. Not intended for external distribution.

Thank you!

