**Github and Branching approach - User Manual**

# Introduction:

This document features the details about Github and it's branching approaches in order to make changes in existing code or queue scripts for development activities.

**General Info:**

**Master Branch :** Contains the production code and queue scripts.

**Dev Branch :** Contains the out of factory code and queue scripts. All feature branches needs to be created from the Dev Branch.

**Feature Branch :** Whenever code or queues needs to be modified, a new feature branch is created from the Dev Branch in Github repositories. After successfully deploying the changes in prod region the changes must be merged with Master branch and the feature branch should be deleted by the Prod support team.

# Steps for creating a new feature branch:

Make sure the user has required access privilege to create and modify the repository.

**Step 1:** Navigate to the required repository’s DEV branch.

**Step 2:** Click the branch selector menu.

**Step 3:** Enter an unique name for the new feature branch.

**Step 4:** Press Enter. New feature branch will be created in the same repository.

**Example Scenario:**

Consider repository **adapter-iib** where we want to create a new feature branch DEMO-FEATURE1 and DEMO-DEV as its base.

**Step 1:** Navigate to the adapter-iib repository in Github.com and select DEMO-DEV in the branch menu. Ref Fig 1.1.

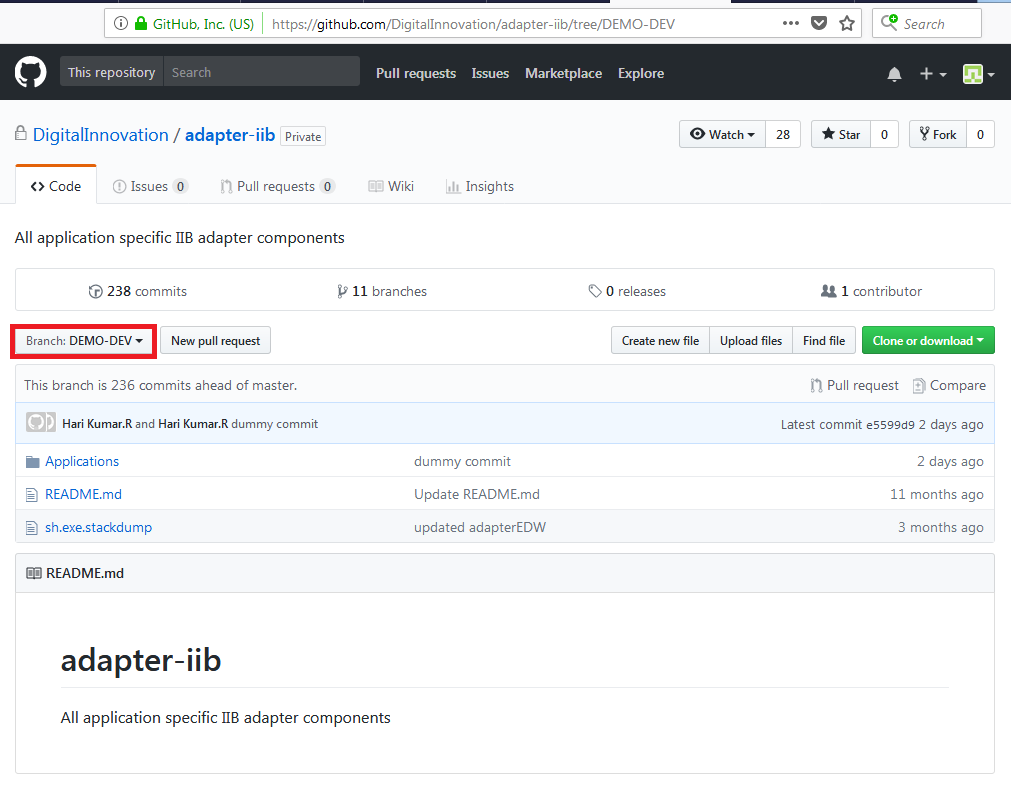


Fig 1.1

**Step 2:** Click on the branch selector menu. Ref Fig 1.2.

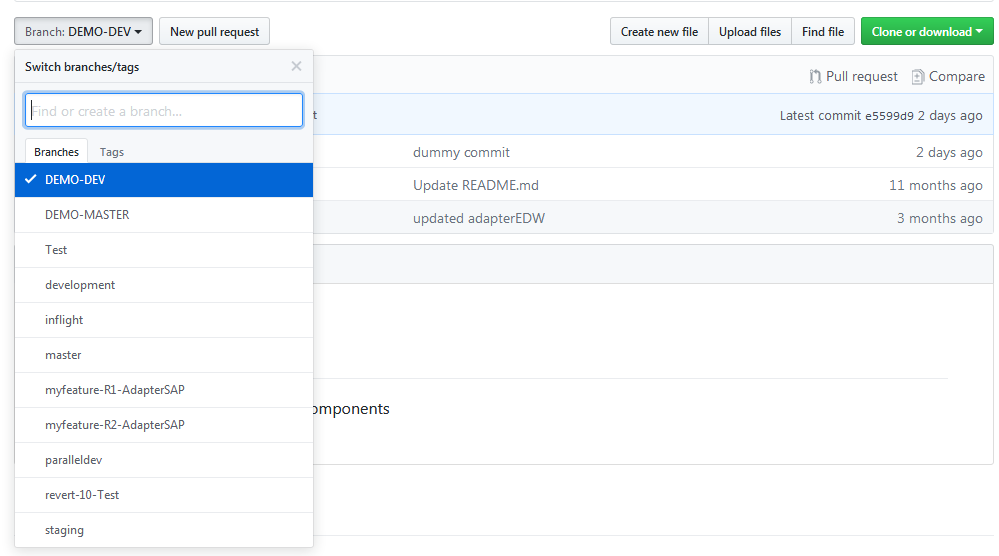


Fig 1.2

**Step 3:**Enter the new Feature branch name in the menu. Ref Fig 1.3.

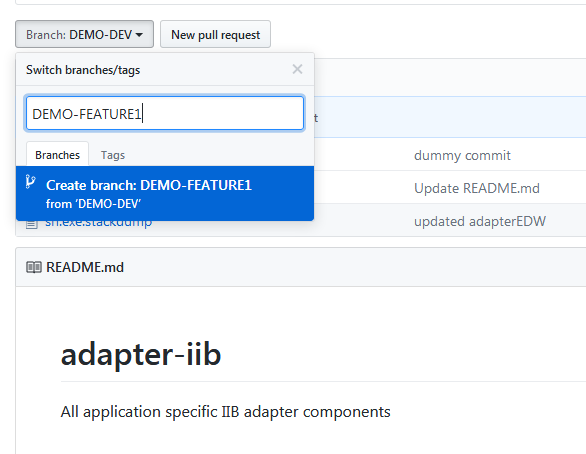


Fig 1.3

**Step 4:** Press Enter. New branch will be created in repository. Ref Fig 1.4.

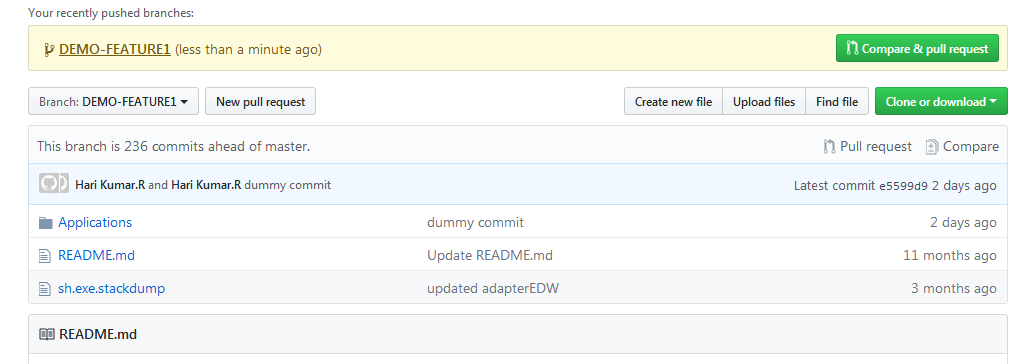


Fig 1.4

Similarly we can create new feature branches in any Github repository.

# Scenario I

# Feature Branches HappyPath:

Happy Path - A feature branch which moves from development region through the non-prod environments and finally into the production region without any blockers.

To perform development activities in any application, the below steps needs to followed.

**Step 1:** Create a new unique feature branch from DEV Branch.

**Step 2:** Do the required changes.

**Step 3:** Commit the changes to the feature branch.

**Step 4:** Check for new changes in DEV branch.

**Step 4(a):** If any changes present, merge those DEV Changes into the feature branch.

**Step 5:** After releasing the application from factory, merge the changes from feature branch into DEV branch.

**Step6:** Once the changes are deployed into prod successfully, the respective feature branch needs to be deleted.

# Example:

Consider a new change is to be made in AdapterEDW\_01 application present in adapter-iib.

**Step 1:** Create a new feature branch **DEMO-01-AdapterEDW\_01** from DEMO-DEV branch.

Ref Fig 2.1

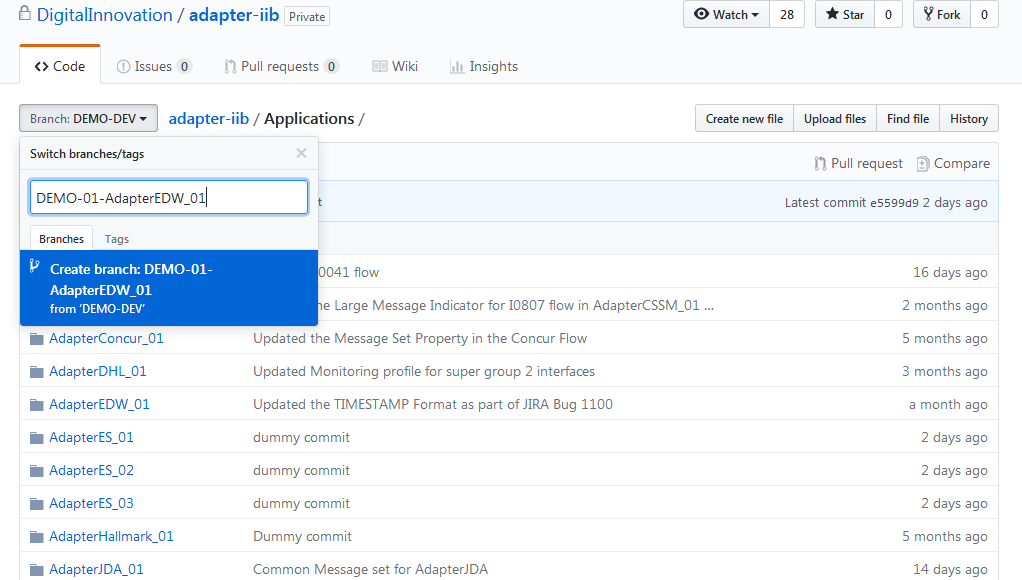


Fig 2.1

**Step 2:** GoodsMovement\_ToEDW\_F0038\_v1\_Database\_Load\_Message.esql code is modified. Ref Fig 2.2



Fig 2.2

**Step 3:** Commit the new changes into the feature branch by selecting commit change button after adding a comment. Ref Fig 2.3

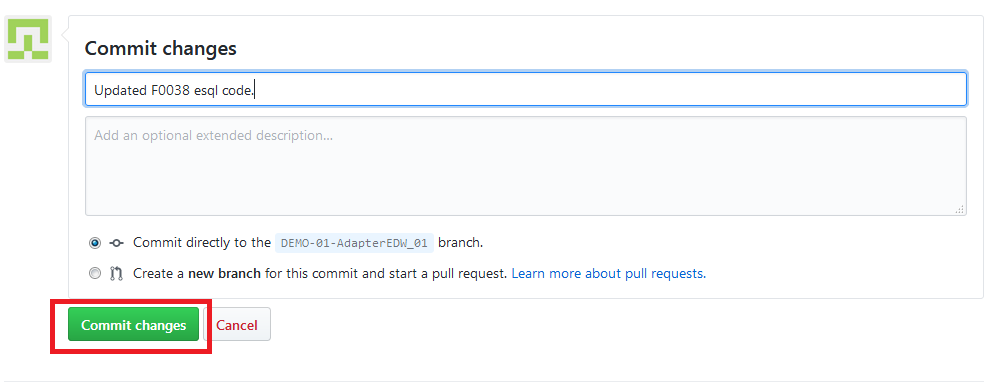


Fig 2.3

**Step 4:** After releasing the application into non-prod environments, merge the changes into DEV branch. Check for any new changes in DEV Branch.

1. Select new pull request option in DEMO-01-AdapterEDW\_01. Ref Fig 2.4(a)

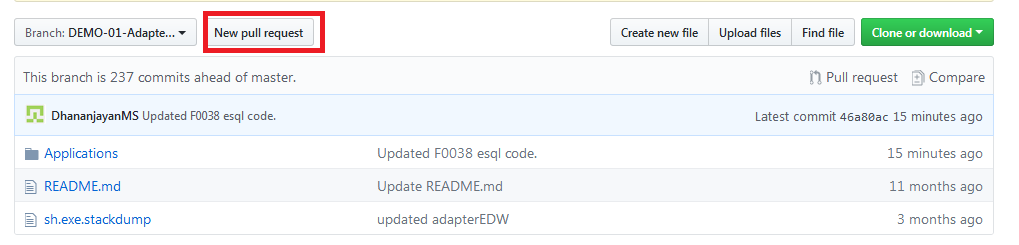


Fig 2.4(a)

1. Select DEMO-01-AdapterEDW\_01 branch as base and DEMO-DEV as compare to check any new changes made in DEMO\_DEV branch. Ref Fig 2.4(b)

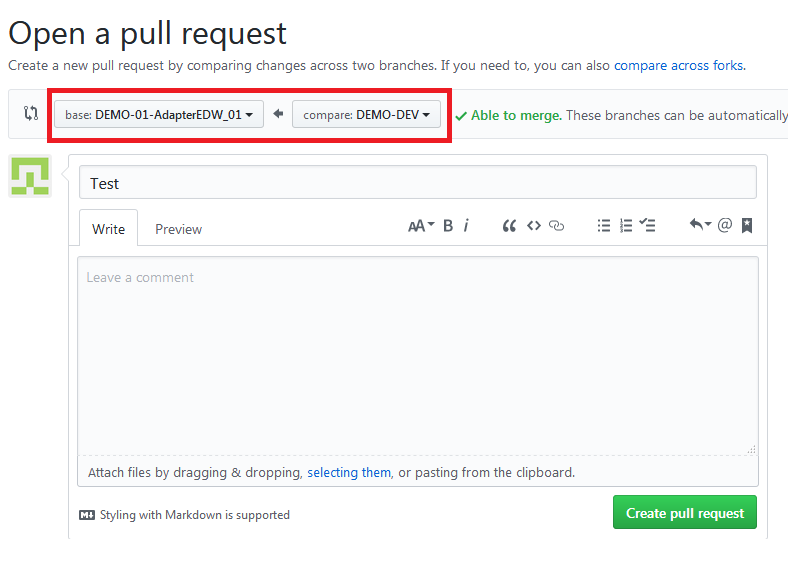


Fig 2.4(b)

1. If no new changes are made we can proceed without any merge from DEMO-DEV. Ref Fig 2.4(c)

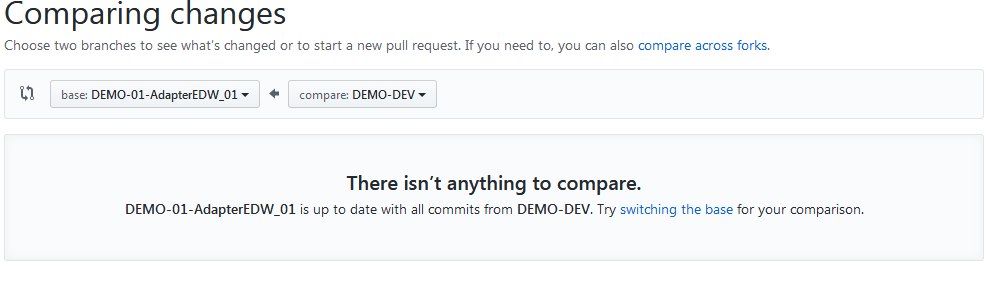


Fig 2.4(c)

**Step 5:** Merge the changes from feature branch into DEV.

1)Select new pull request option in DEMO-01-AdapterEDW\_01. Ref Fig 2.5(a)

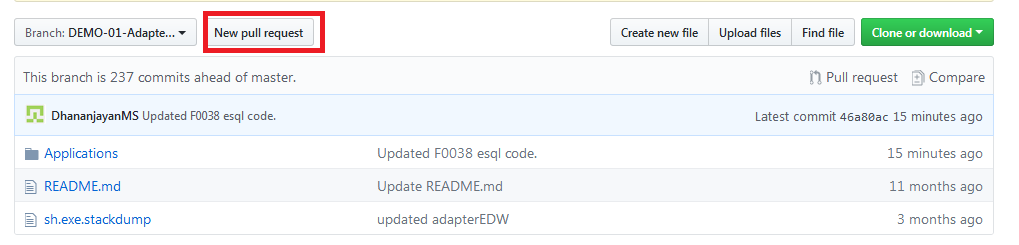


Fig 2.5(a)

2) Select DEMO\_DEV branch as base and DEMO-01-AdapterEDW\_01 as compare. Select Create pull request.Ref Fig 2.5(b)

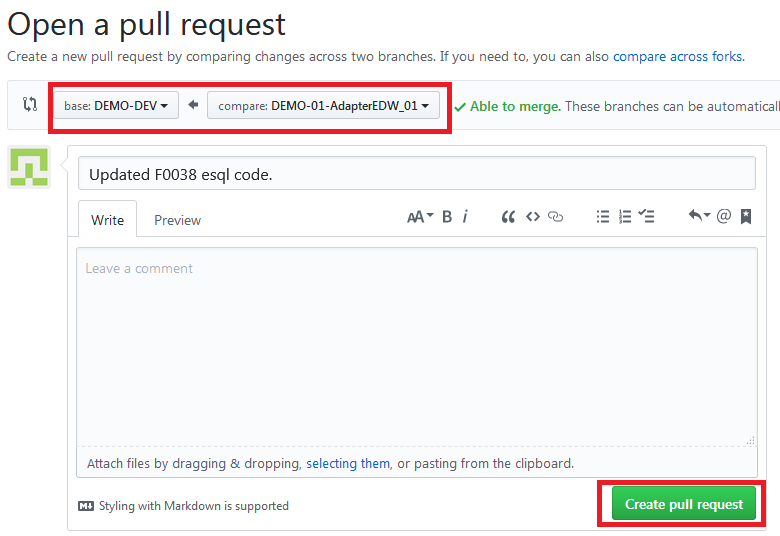


Fig 2.5(b)

1. Select Merge pull request option. Ref Fig 2.5(c)

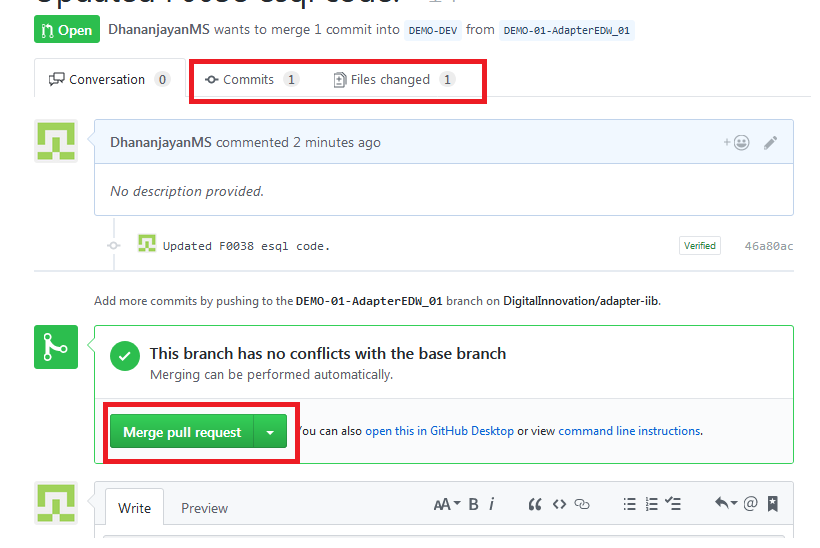


Fig 2.5(c)

NOTE: **Commits -** displays all the commits made in feature branch. **Files changed** - lists all the modified files present in the branch. These options can be used for verification purposes.

1. Add comment about the merge and select confirm merge. Ref Fig 2.5(d)

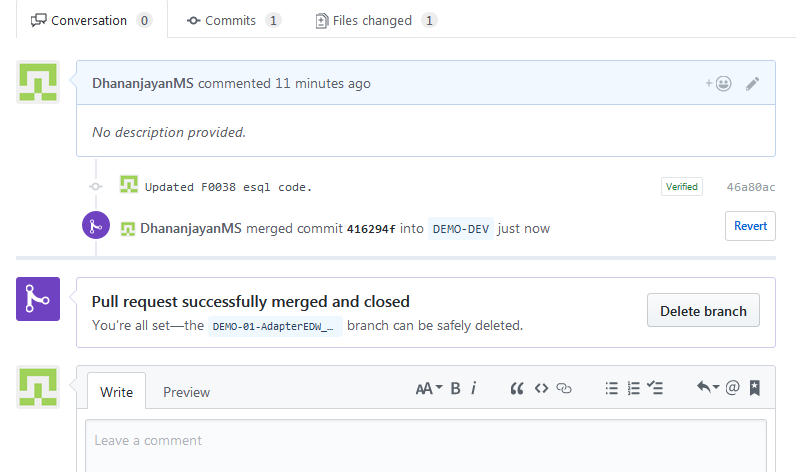


Fig 2.5(d)

The Changes are successfully updated in DEV branch. The above needs to be followed for all development activities.

**Step 6:** To delete a branch.

1) Navigate to the main page of the repository. Ref Fig 2.6(a)

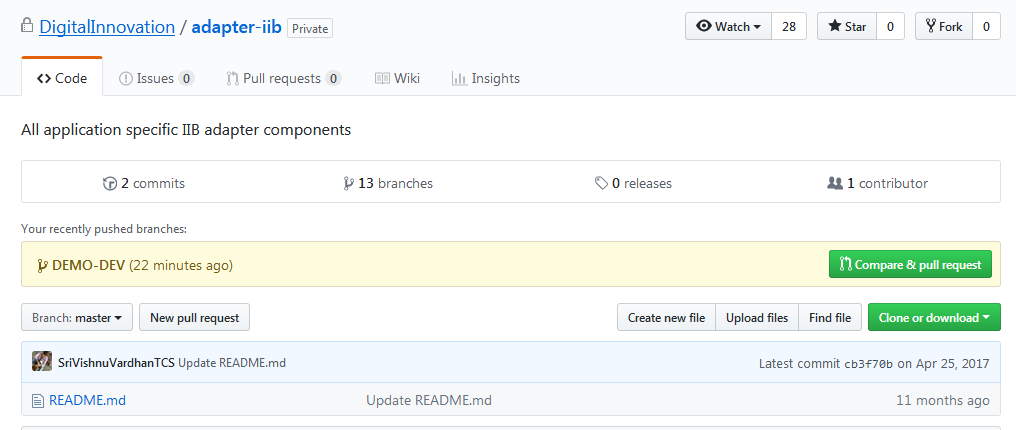


Fig 2.6(a)

1. Select NUMBER branches option.Ref Fig 2.6(b)

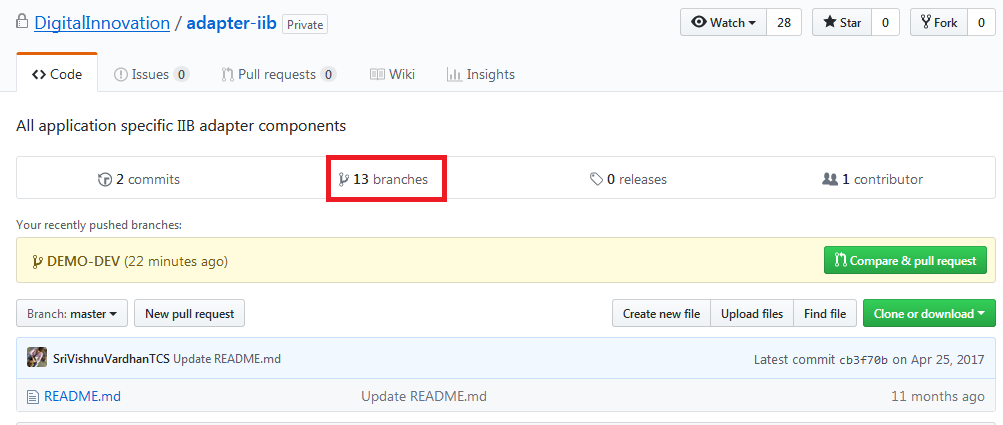


Fig 2.6(b)

1. Scroll to the branch you want to delete, then click delete icon to delete the branch. Ref Fig 2.6(c)

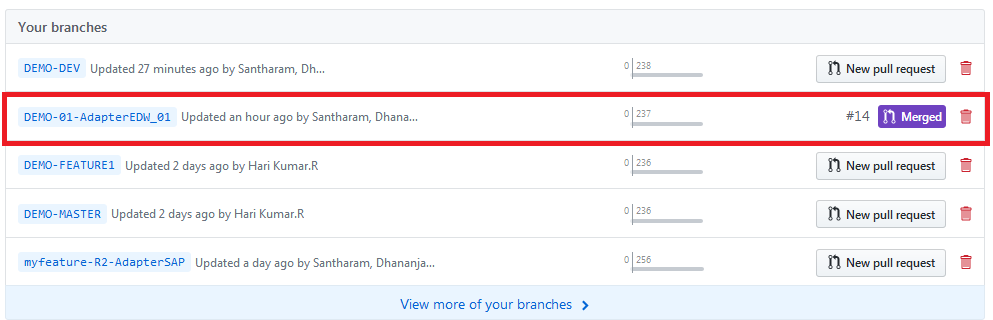


Fig 2.6(c)

# Scenario 2:

# Feature Branches HappyPath with Bug-fix:

Happy Path with Bug Fix - A feature branch while moving from development through the non-prod environments contains a bug, Bug-fix is applied in non-prod , and finally deployed into production.

When a non-prod bug fix is applied to the applications following steps needs to followed in Dev region

**Step 1:** Apply the bug fix changes in application in the corresponding feature branch.

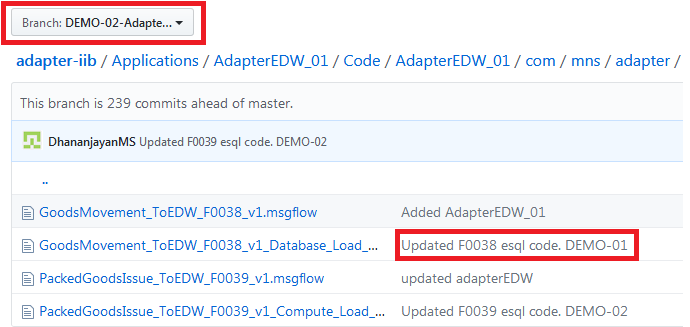
**Step 2:** Merge the feature branch’s changes into DEV branch.

**Step 3:** The bug fix needs to be updated in all the variants of the same application containing the defective version.

**Step 4:** Release all the bug fixed versions to other regions.

# Example:

Consider changes made in AdapterEDW\_01 using feature branch DEMO-01-AdapterEDW\_01. Further changes are made in same application by DEMO-02-AdapterEDW\_01 over the changes of DEMO-01.



When a non-prod bug fix is provided in DEMO-01 , the same needs to be updated in DEMO-02 version too.

**Step 1:** Update the non-prod code fix and commit the changes in DEMO-01-AdapterEDW\_01 branch. Ref Fig 3.1

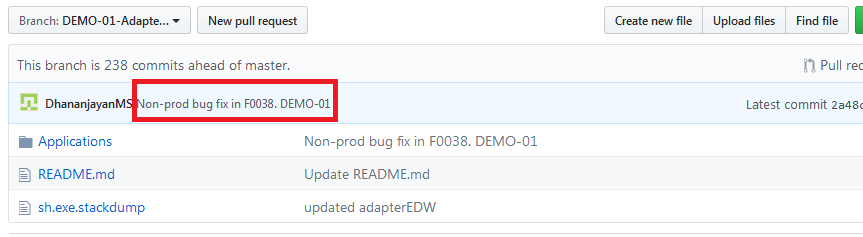


Fig 3.1

**Step 2:** Merge the changes with DEMO-DEV branch. Ref Fig 3.2

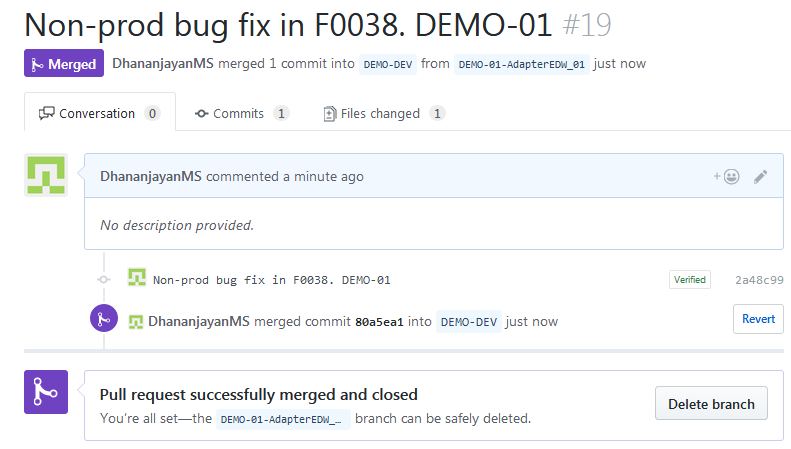


Fig 3.2

**Step 3:** To update the DEMO-01 non-prod fix in the DEMO-02 feature branch below steps needs to followed.

1. Select New pull request option in DEMO-02-AdapterEDW\_01. Ref Fig 3.3(a)

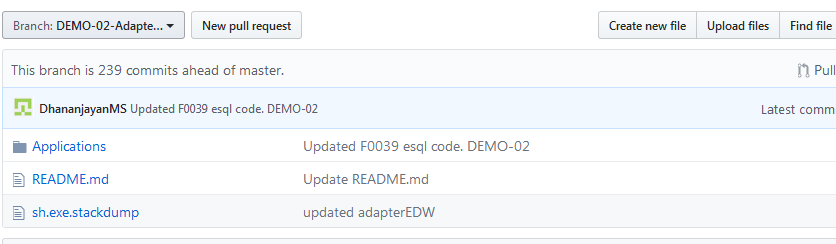


Fig 3.3(a)

1. Select DEMO-02-AdapterEDW\_01 as base and DEMO-DEV as compare. Ref Fig 3.3(b)

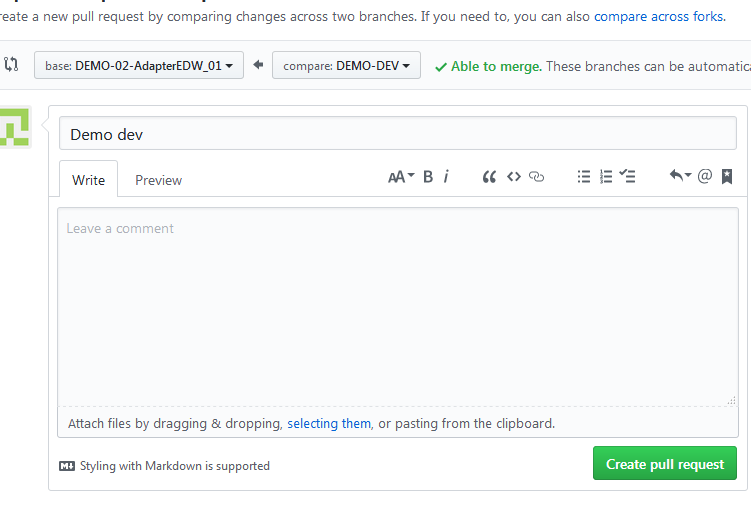


Fig 3.3(b)

Note : **base** - refers the destination branch where new update is made. **compare -** refers the source branch where the change is already present.

1. Merge the required changes from DEV branch. Ref Fig 3.3 (c )

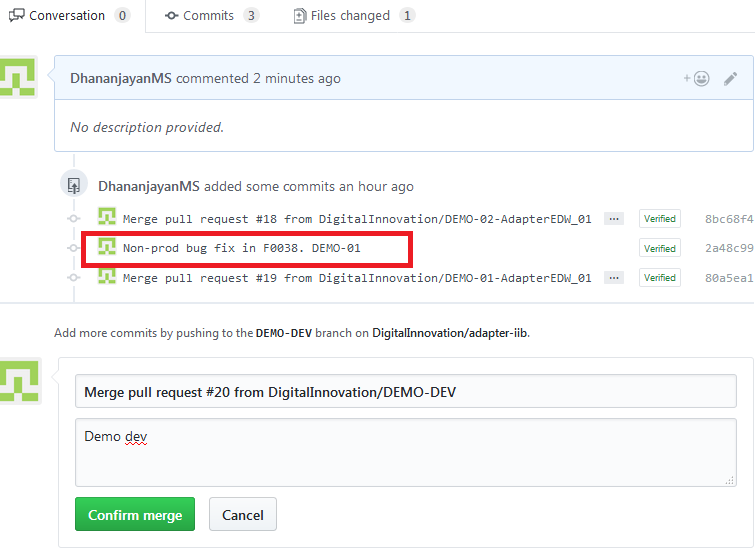


Fig 3.3 (c)

1. Required changes are successfully made in DEMO-02 branch. Ref Fig 3.4 (d)

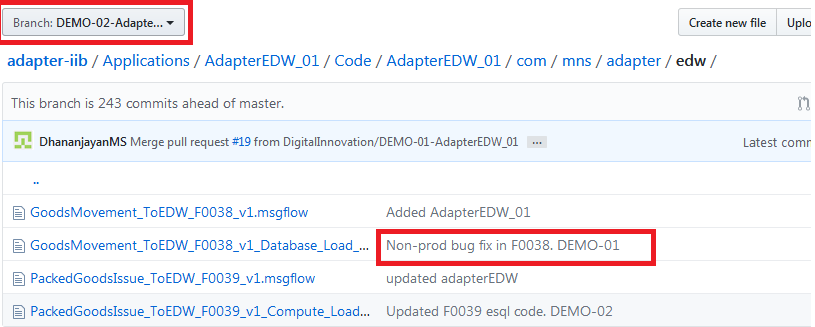


Fig 3.4 (d)

After Updating the non-prod fix in DEMO-02-AdapterEDW\_01 ,new stack-id has to be shared with the non-prod regions.

# Reverting A Merge:

To revert a merge following steps needs to be followed.

**Step 1 :** Navigate to the repository. Select Pull requests tab in the web page. Ref Fig 4.1.

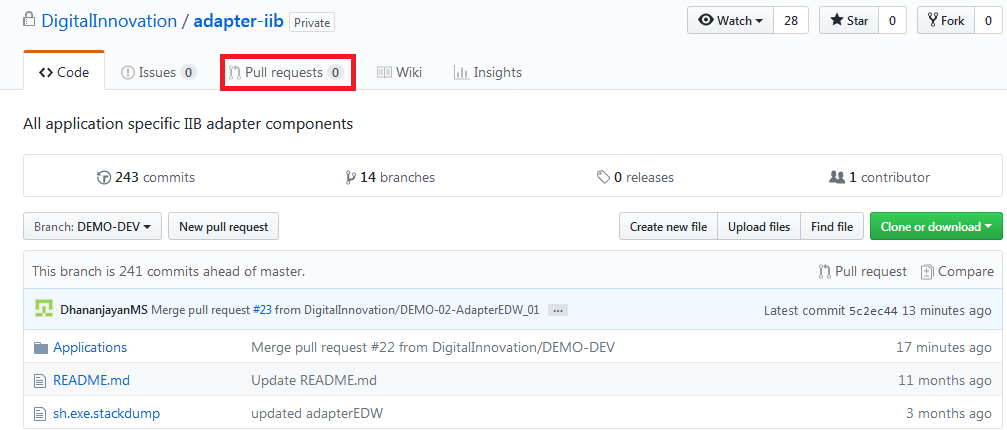


Fig 4.1

**Step 2:** Select closed Pull request. Ref Fig 4.2.

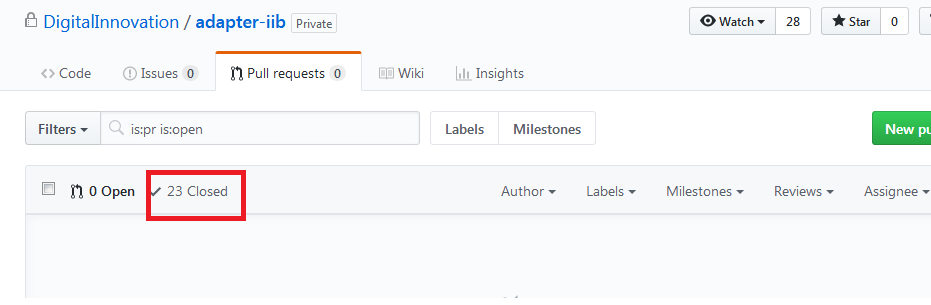


Fig 4.2

**Step 3:** In the "Pull Requests" list, click the pull request you'd like to revert. Ref Fig 4.3

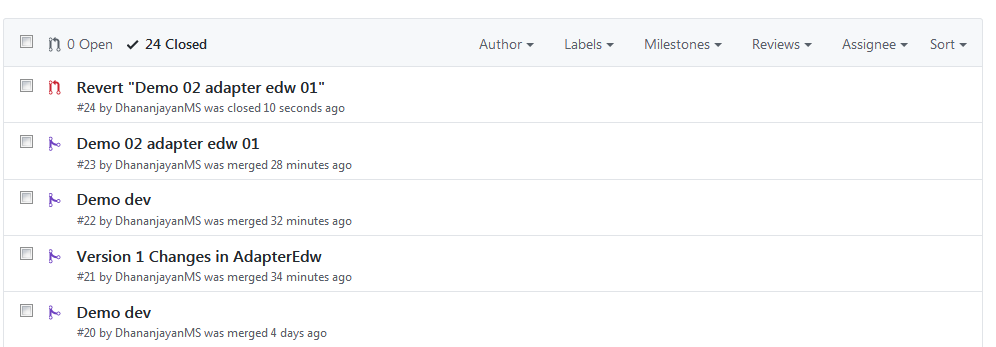


Fig 4.3

**Step 4:** Click Revert at the bottom of the pull request. Ref Fig 4.4

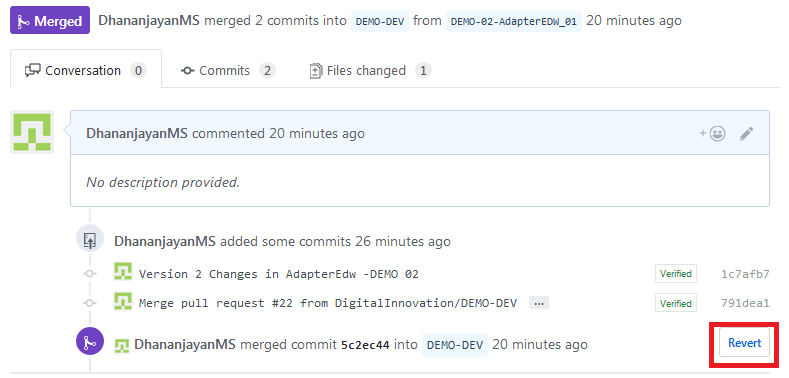


Fig 4.4

**Step 5:** Select Create pull request option. Ref Fig 4.5

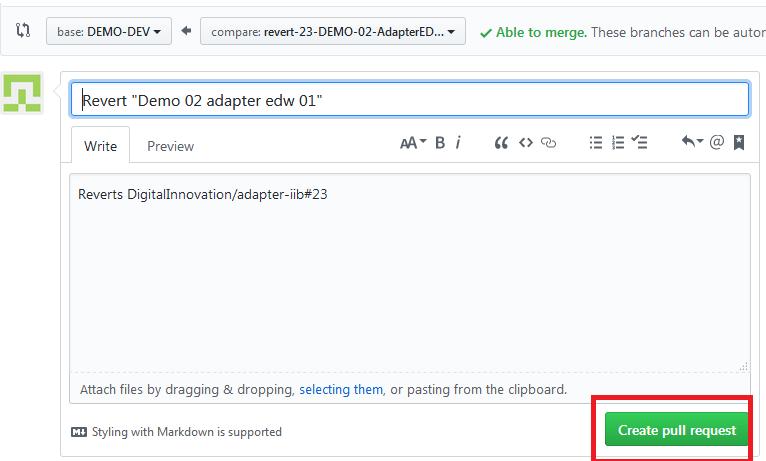


Fig 4.5

**Step** **6:** Select Merge Pull Request and Confirm merge to complete the revert. Ref Fig 4.6

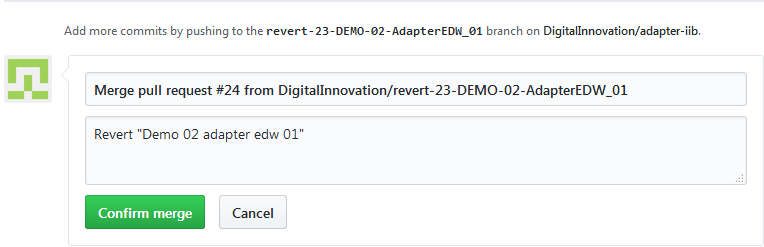


Fig 4.6

This will revert all the changes made by the selected merge from the branch.

# Scenario 3:

# Feature Branches Unhappy Path Leapfrogging:

Unhappy path features the scenario when a most recent release needs to leapfrog a older release .

# Example:

Consider two existing release components of AdapterEDW\_01 Version 1 and Version 2 present in non-prod regions. Version 1 changes made by DEMO-01 feature branch and Version 2 by DEMO-02 feature branch. Version 2 contains the changes made by Version 1.

When Version 2 needs to leapfrog the Version 1 below steps needs to followed.

**Step 1:** Remove the Version 1 changes from DEMO 02 feature branch, by reverting the merge with DEMO-DEV. Ref Fig 5.1 (a). Reversion will remove all the changes made by the merge. Ref Fig 5.1 (b)

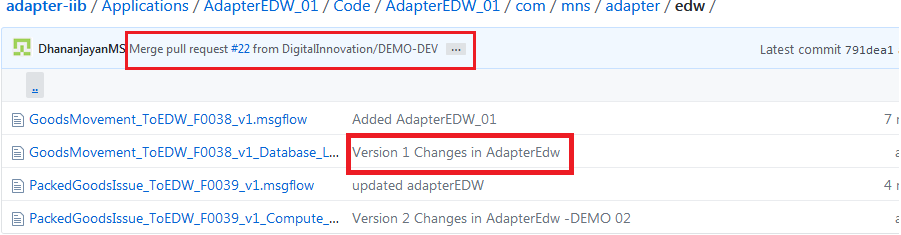


Fig 5.1 (a)

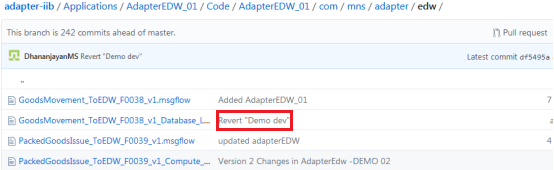


Fig 5.1 (b)

**Step 2:** Remove the Version 1 changes from DEMO-DEV branch, by reverting the merge with DEMO-01. Ref Fig 5.2 (a). Reversion will remove all the changes made by the merge. Ref Fig 5.2 (b)

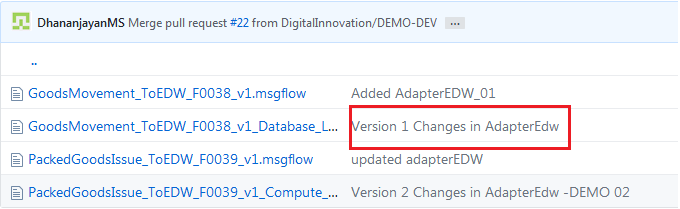


Fig 5.2 (a)

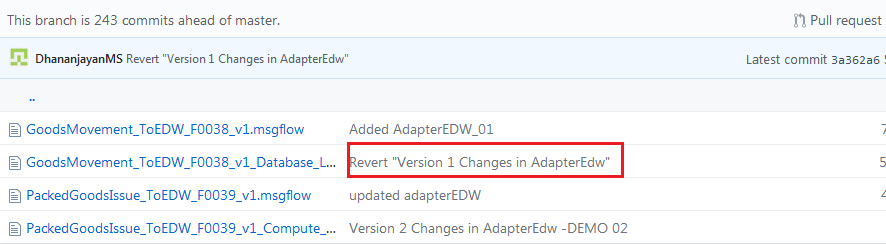


Fig 5.2 (b)

**Step 3:** After reverting the version 1 changes from both DEMO 02 and DEMO-DEV, share the updated Stack id to higher environments.

**Step 4:** Since DEMO-02 changes are going before DEMO 01 changes, DEMO-01 needs to be updated with DEMO-02 changes by merging the DEMO-01 with DEMO-DEV branch.

**Step 5:** After updating DEMO-01 share the latest stack id.

# Scenario 4:

# Hot fix Branch HappyPath:

When an incident occurs in the production region, a fix will be applied via hot fix feature branch and gets tested in pre-prod region. After testing ,the fix is deployed in prod. The changes are merged in MASTER branch directly and the hot fix feature branch will be deleted by production support team.

This fix change has to be updated in all the variants of the application in non-prod regions. It must be performed from oldest release to recently release order.

# Example:

The below steps needs to be followed when a hot fix is performed in the master branch.

Consider a hot fix is performed in AdapterEDW\_01 in master branch. The same has to be updated in all the variants of this application. DEMO-01 is oldest variant and DEMO-02 is the recent one.

**Step 1:** Update the hot fix changes in oldest release of this Application(DEMO-01). Ref Fig 6.1.

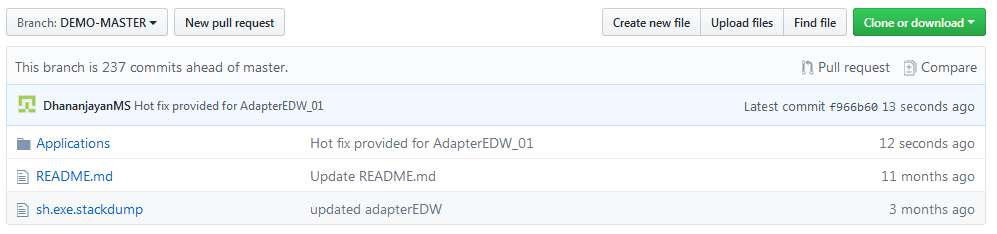


Fig 6.1

**Step 2:** Merge the DEMO-01 with DEMO-MASTER branch. Ref Fig 6.2.

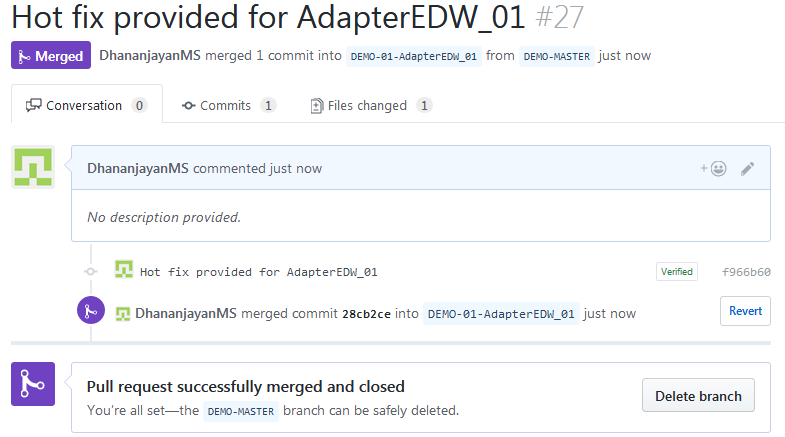


Fig 6.2

**Step 3:** Update the DEMO-01 changes to DEMO-DEV and release the updated version. Ref Fig 6.3.

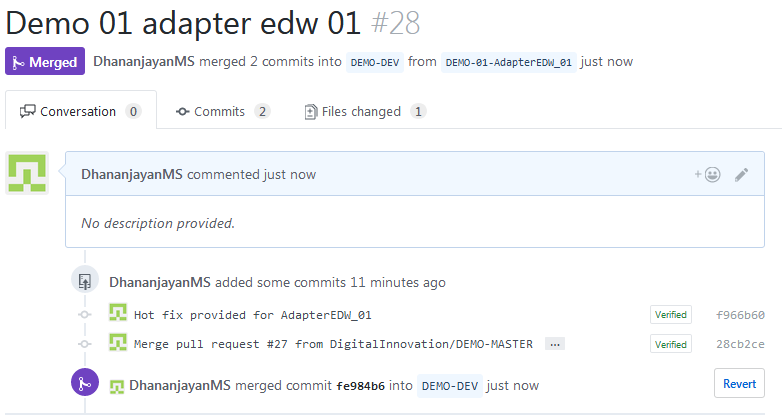


Fig 6.3

**Step 4:** Update the other variants from DEV branch and release the updated version to non-prod. Ref Fig 6.4.

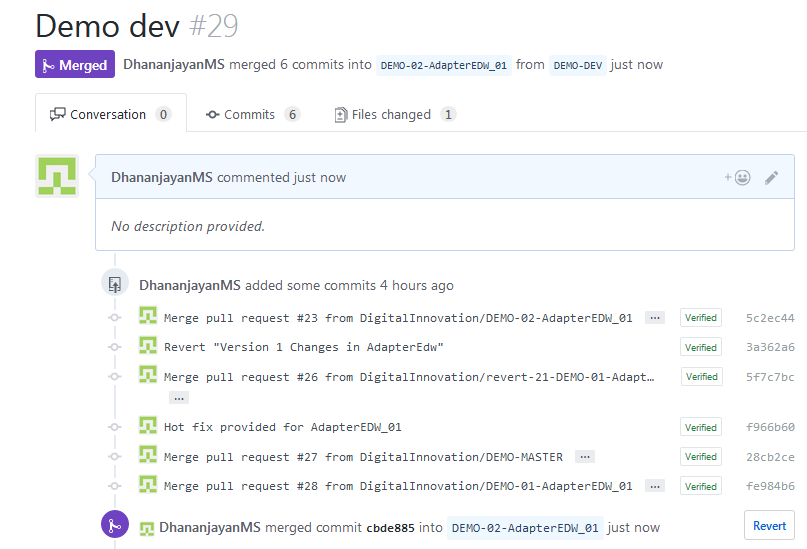


Fig 6.4

Similarly all other variants has to updated from oldest to recent release.

# Reference:

1. <https://confluence.platform.mnscorp.net/display/SYSINT/FeatureBranches_HappyPath>
2. <https://confluence.platform.mnscorp.net/display/SYSINT/FeatureBranches_HappyPath_with_BugFix>
3. <https://confluence.platform.mnscorp.net/display/SYSINT/FeatureBranches_UNHappyPath_Leapfrogging>
4. <https://confluence.platform.mnscorp.net/display/SYSINT/HotfixBranch_HappyPath>

## ****Thanks for Reading****