**DevOps SnowFlake Pipeline Process**

**Scope:**

This document provides details of DevOps Snowflake pipeline branching strategy for SUPPLY

**Process Flow:**

A screenshot of a computer

Description automatically generated

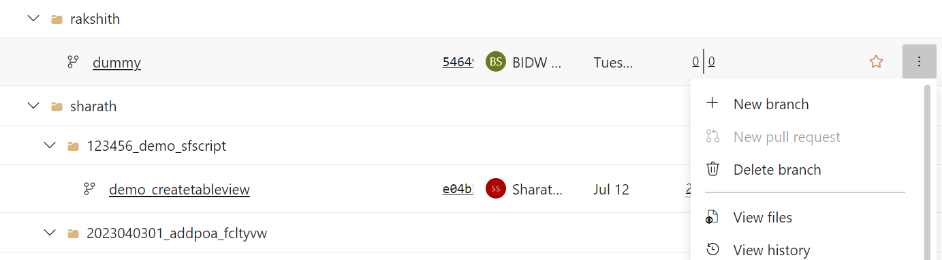
**Step by step process:**

1. Developer to create a folder with his/her name in the repo by creating a dummy branch under users/ folder using Create New branch

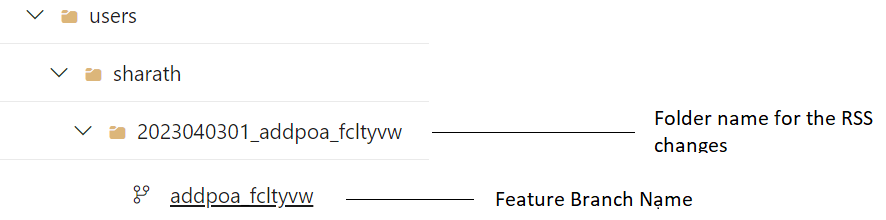
A screenshot of a computer

Description automatically generated

1. Developer to create a feature branch with his/her RSS item parent folder as <TOPAZ ID>\_<rssdesc>/<rssdesc> (ex: users/sharath/2023040301\_addpoa\_fcltyvw/addpoa\_fcltyvw)
2. Feature branch name to be created as short RSS description Ex: addpoa\_fcltyvw
3. Delete the dummy branch created - click on Delete Branch and delete it (post creation of feature branch)

****

1. All characters to be of lower case while creating folder, feature branch



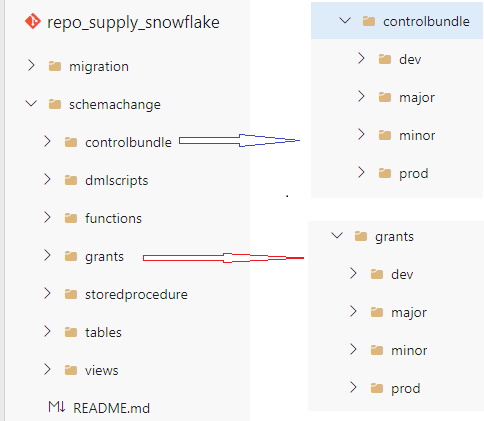
1. The branch contains the below folders:

* migration
* schemachange

A screenshot of a computer

Description automatically generated

1. schemachange will contain the below folders:
   1. controlbundle
      1. dev
      2. minor
      3. major
      4. prod
   2. dmlscripts
   3. functions
   4. grants
      1. dev
      2. minor
      3. major
      4. prod
   5. storedprocedure
   6. tables
      1. alter\_tables
      2. create\_tables
   7. views



1. Developer creates individual script for the changes RSS needs with the format R\_\_<script name>.sql - capital R followed by two underscores is mandatory
2. Developer uploads the scripts into corresponding folder under feature branch and commits

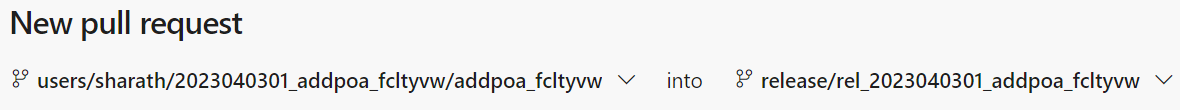
A screenshot of a computer

Description automatically generated

1. Developer creates a new branch based on main branch under release/ folder with same name as created in feature = branch along with Topaz\_id – rel\_<TOPAZ\_ID>\_<rssdesc>



1. Developer creates a new pull request to merge from newly created feature branch into release branch created in step 10



1. D and A Lead (the lead assigned for the RSS/project) first rejects the automated trigger created for release branch creation. Then reviews the Pull Request details then takes action
2. Once the PR is approved, the merge from feature branch to release branch is completed
3. If the lead finds any issues with scripts, the PR is rejected
4. Automated deployment trigger is created once the merge from feature branch to release branch is completed. Technical lead reviews and takes action on the automated deployment request
5. Once the approval is received on the automated deployment request, the DB Scripts are executed in DEV environment
6. The pipeline waits for Minor deployment trigger approval. This is cancelled by the lead to ensure there is no roadblock to approve any other deployments into DEV as pipeline execution is sequential
7. Post completion of Unit testing, the D and A lead re-deploys and approves the deployment request from the same pipeline to deploy the code into Minor environment
8. DBA reviews the deployment request and approves the request to deploy the code into Minor
9. Post successful completion of testing, from the same pipeline, major deployment request is approved by lead, followed by testing team and DBA respectively to deploy code into Major
10. As per prod release timelines, from the same pipeline, production deployment request is approved by lead, support team and DBA respectively to deploy code into Production.
11. Post completion of business validation in production, lead merges the code from release branch into main branch by manually triggering the release to main sync task in the same pipeline
12. For any execution of scripts failure in the pipeline, the lead reviews and identifies fixes of scripts from developer and redeploying the newly created script
13. For any other pipeline failure, the team needs to reach out to the SF PipeLine support team