## **AZPSPRS-STARS Source Control & Deployment Workflow**

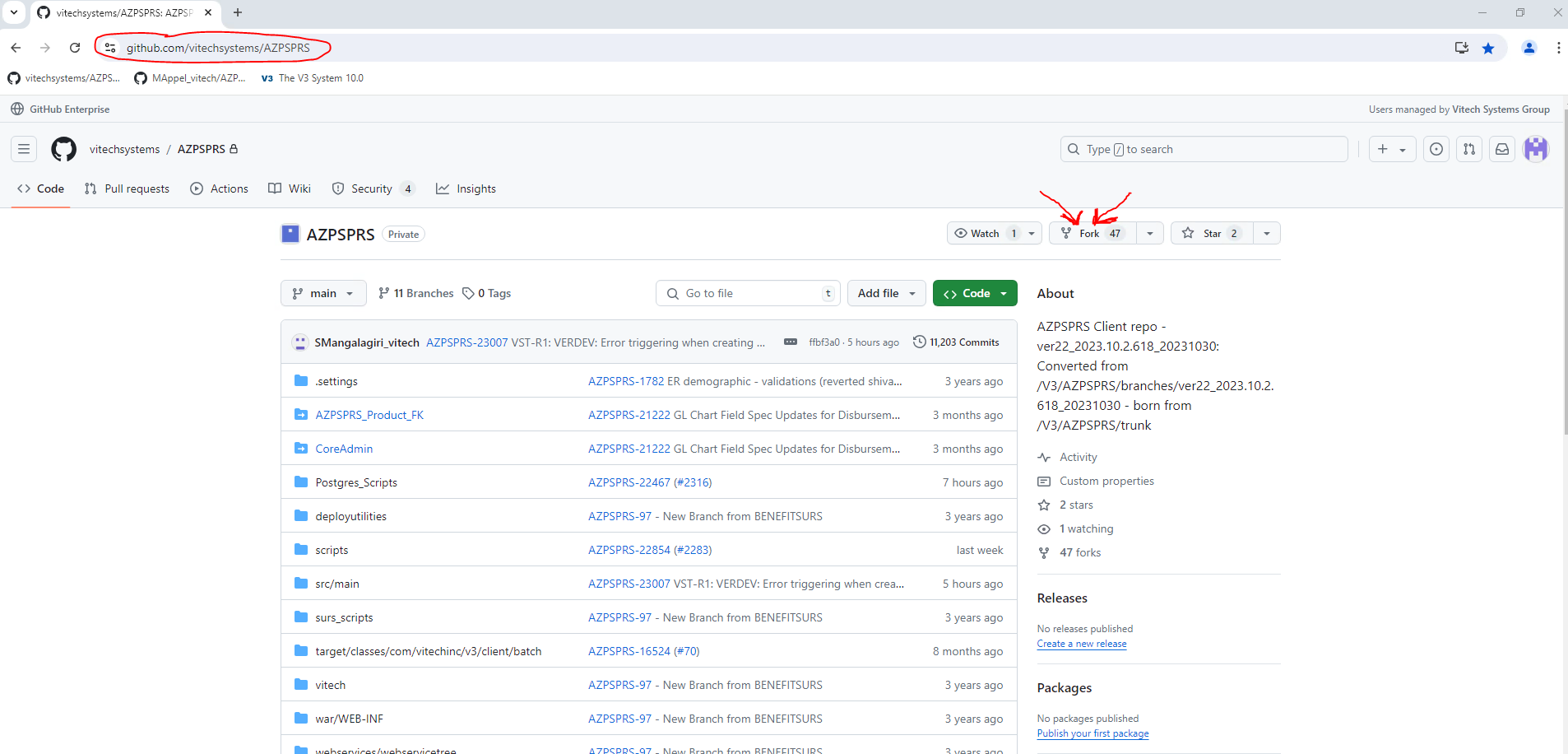
## **Section I. Source Control Management with Git**

1. **GitHub Access**

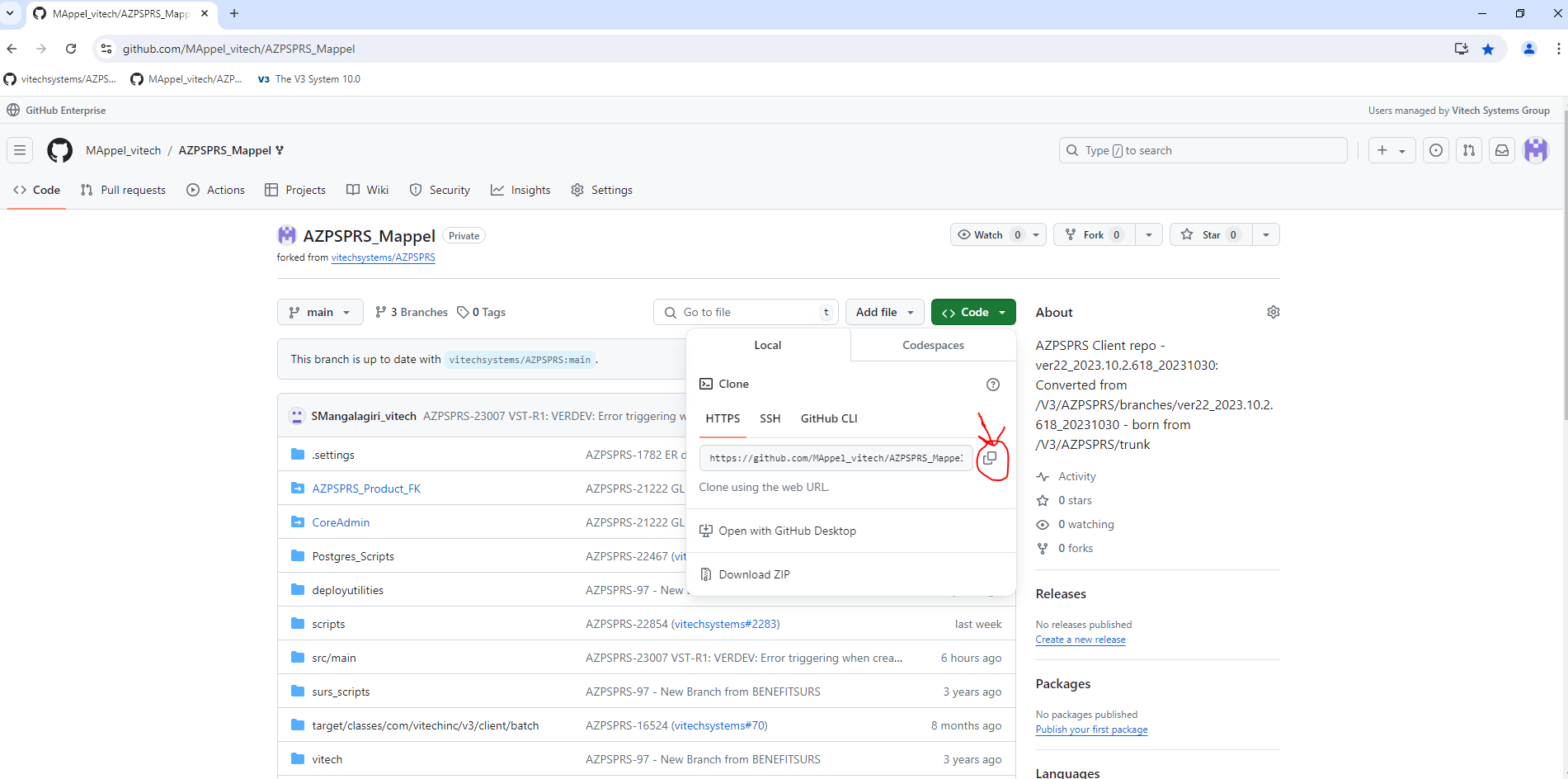
* Open the (<https://github.com/enterprises/vitech>) in Chrome.
* If it doesn't recognize your username, use <username>\_vitech and your Vitech B2C password.
* Ensure Chrome is your default browser as GitHub authentication will use browser login from the command prompt.

1. **Forking the Repository**

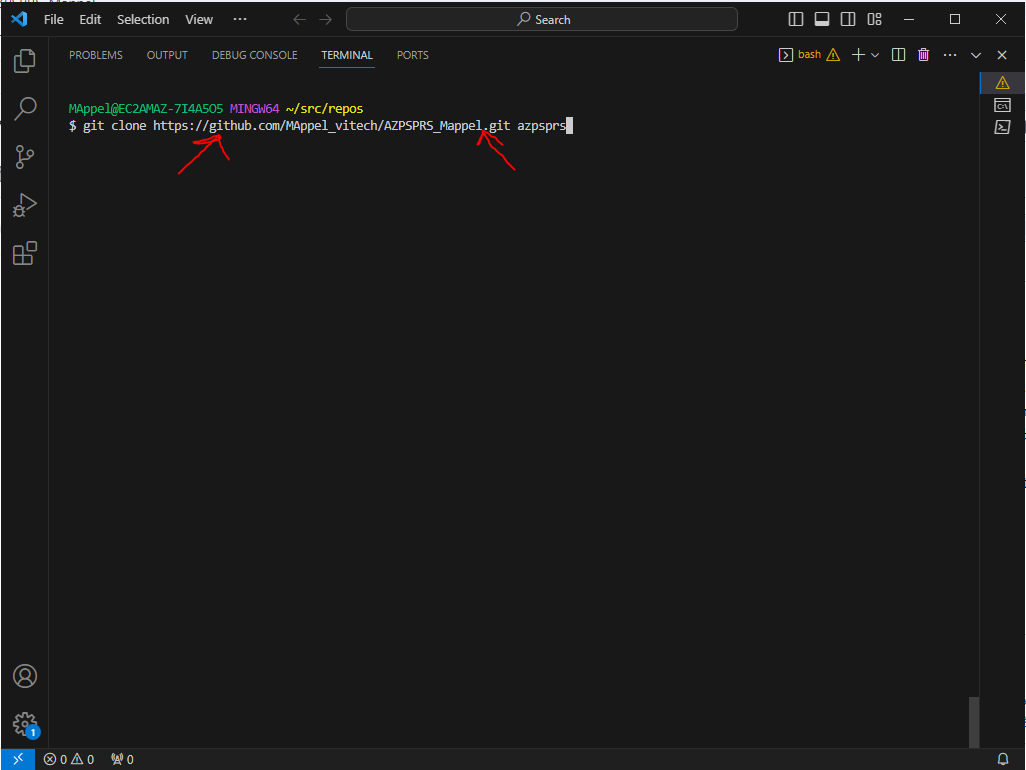
* Navigate to the main GitHub repository page: vitechs/azpsprs
* Fork the repository by clicking the **Fork** button.



* Clone your fork to the local file system using the following steps:
  + Click green code button to expand and then click icon to copy clone url to clipboard

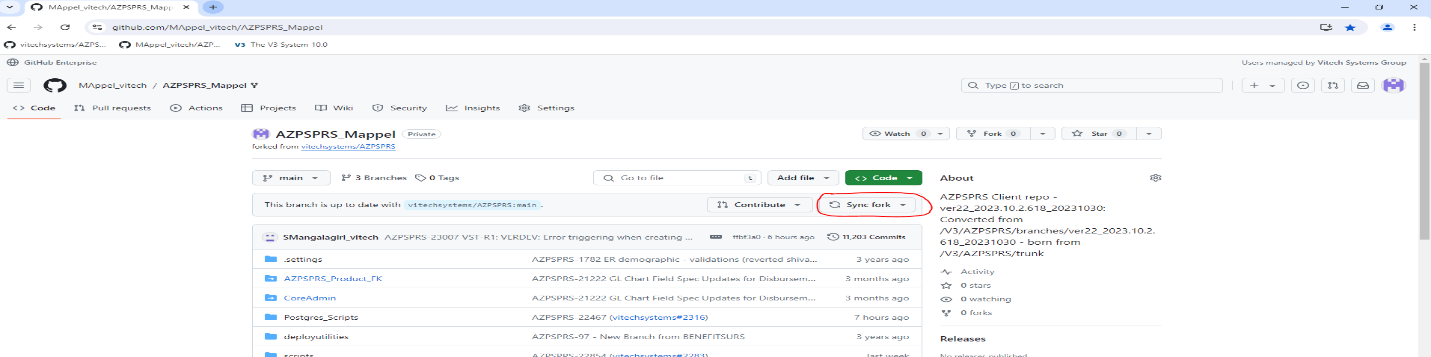


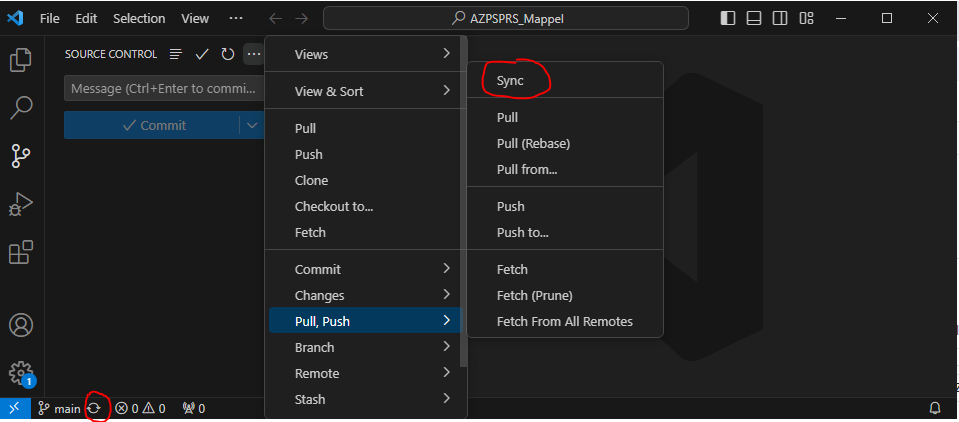
* + Go to your local file system and clone your forked repository



* Add the .gitignore file to the root of the repo. This file instructs Git which type of files not be considered for Git Tracking. (.gitignore file can be found [here](#_References))
* A screenshot of a computer

  Description automatically generated



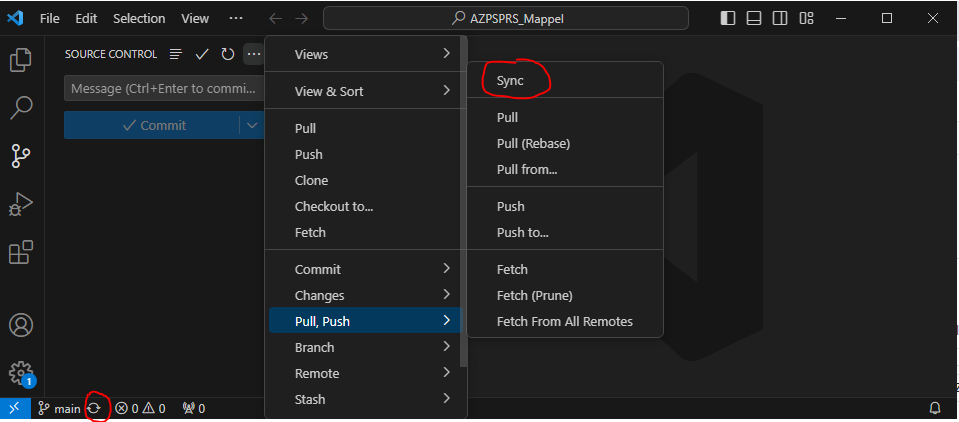


1. **Keep forked repo and local up to date**

* Ensure that your fork and local are synced with the Vitech Main UpStream

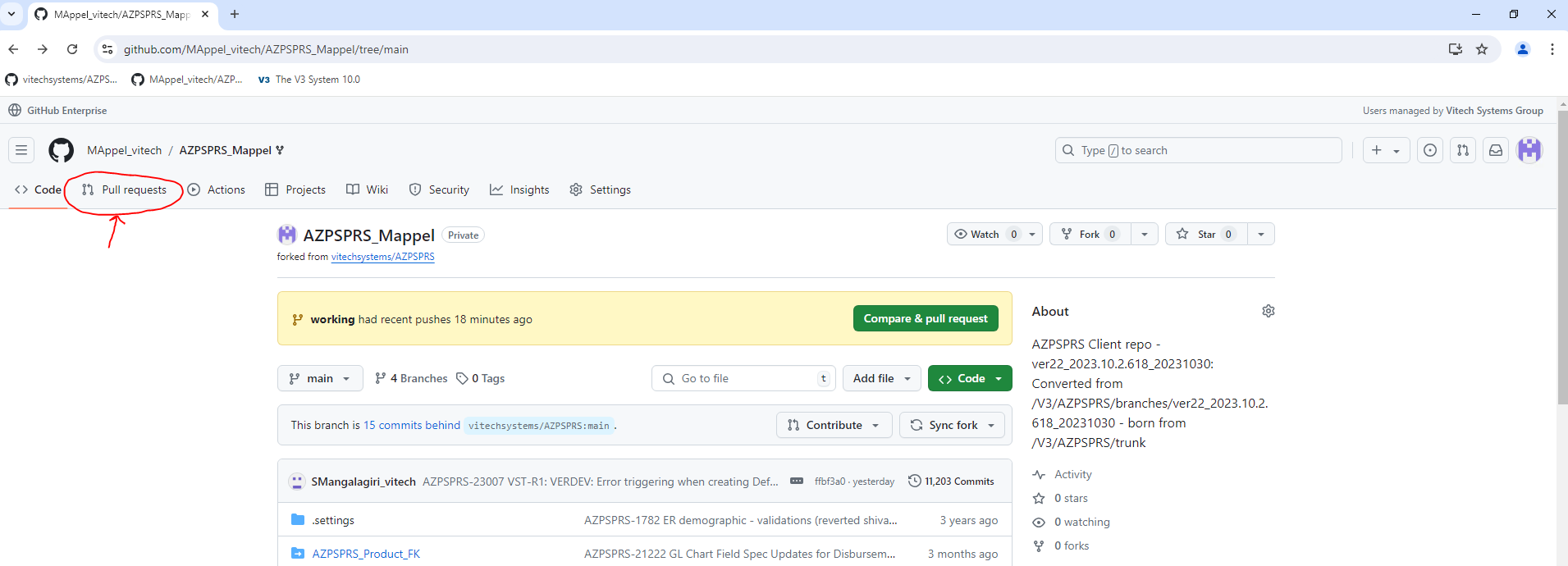
1. **Code Changes**

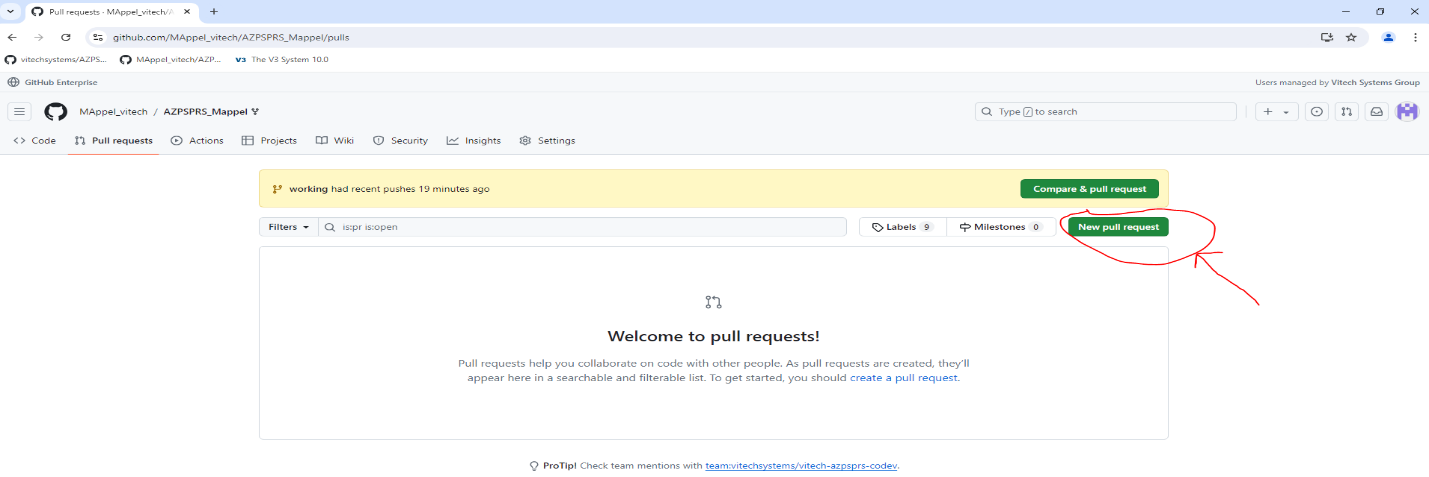
* Use” mvn compile” command after any code changes.
* Using Git GUI/GitHub Desktop or vscode editor, stage and commit changes to local with the correct ViTech Jira ticket number. Then push(Sync) your selected changes to Forked Repo.



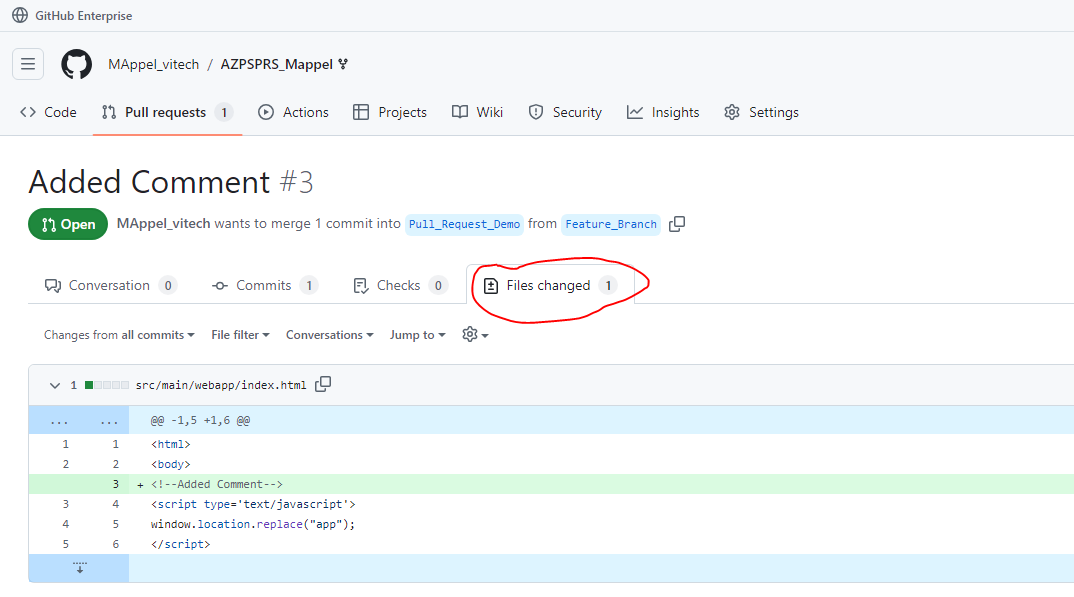
1. **Pull Request (PR) Creation**

* Create a PR to merge changes back into the main Vitech AZPSPRS repository.
* In the PR comment, reference the individual work unit Jira ticket using the format:  
  “<AZPSPRS-Ticket No.>: <Ticket Summary> example: [AZPSPRS-12345](https://jira.vitechinc.com/jira/browse/AZPSPRS-12345): Issue while generating DL013 Document”



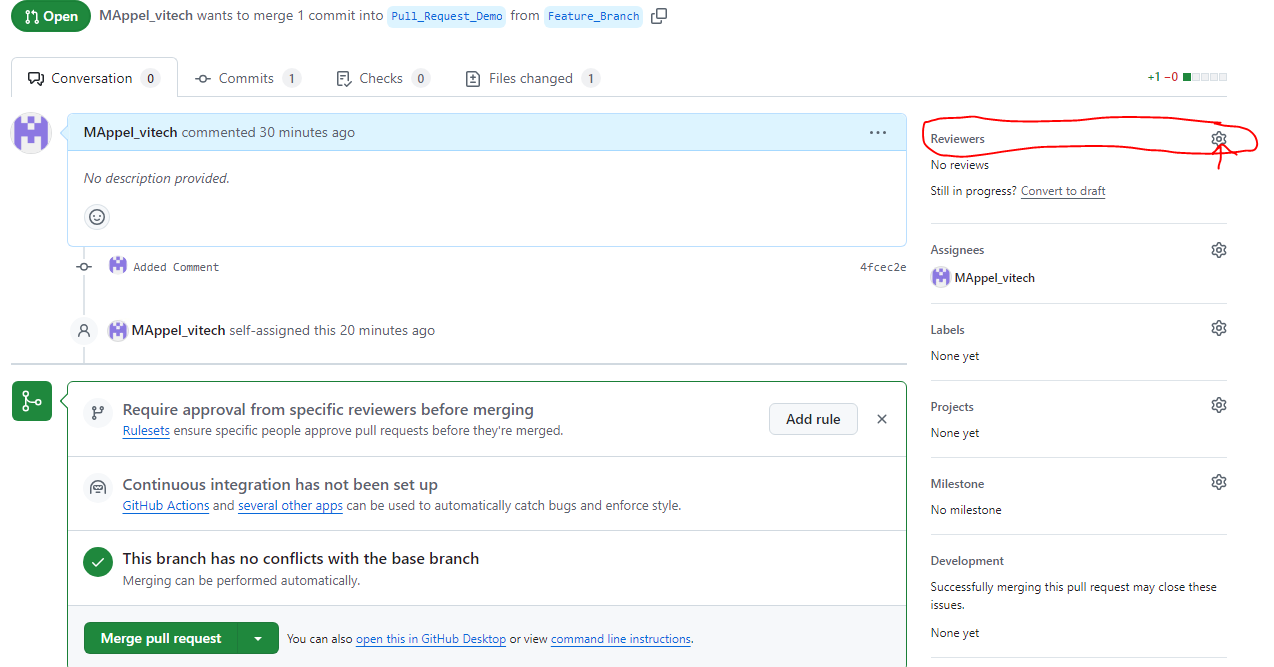


* Ensure only your relevant changes files are included in the "Files changed" section before adding Reviewers. If not close the PR and re-create with your changes.



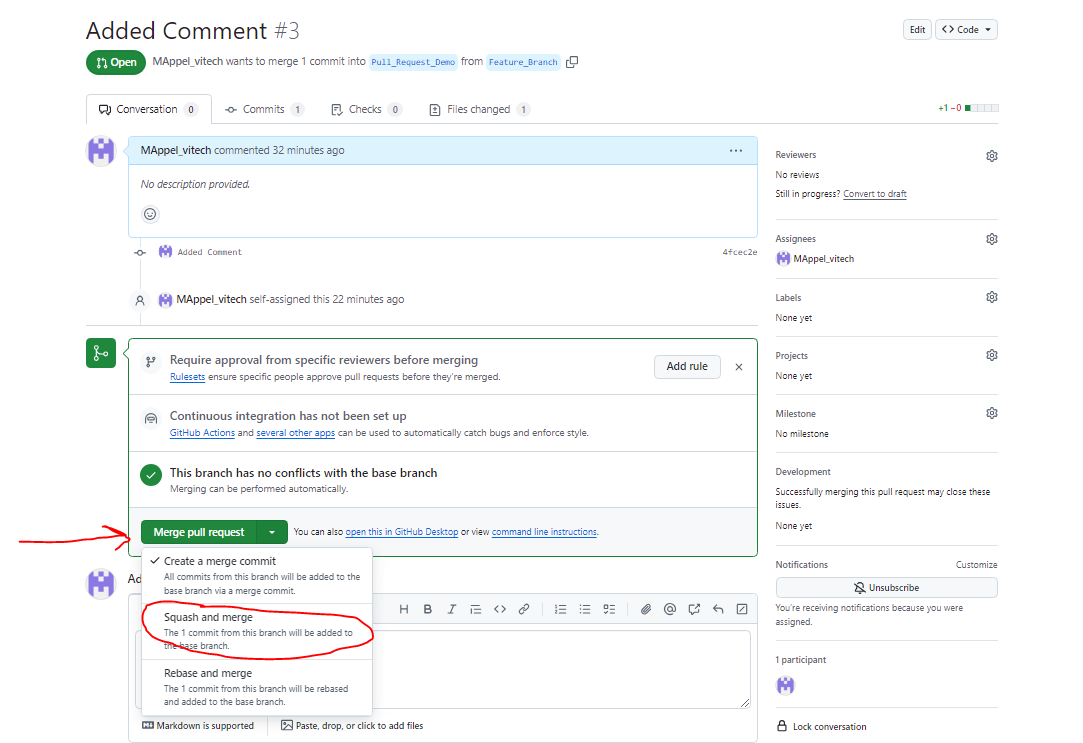
* Assign appropriate reviewers from the following list:

|  |  |  |
| --- | --- | --- |
| **Sr** | **Artifact Type** | **Reviewers** |
| 1 | Documents | Eswara Vayugundla  Subramanyam Nistala  Sridhar Kammary Sidharth Kamat |
| 2 | Query | Anil Atyam  Eswara Vayugundla  Venkatesh Racharla  Sridhar Kammary Sidharth Kamat |
| 3 | Reports | Ramesh Ausali  Nishanth Kona  Jagadeesh Kummara Sridhar Kammary Sidharth Kamat |
| 4 | Business Rule | Jagadeesh Kummara Suresh Mangalagiri Sridhar Kammary Sidharth Kamat |



1. **Finalizing the PR**

* After review and approval, squash and merge the changes into the main branch upstream (vitechsystems/AZPSPRS)
* Make sure to follow up with any required test documentation.



## **Section II – Deployment Workflow**

* The information below from Vitech is an example of the overall deployment workflow, including any required documentation and testing procedures. Use the steps Section I of this document as a reference to implement the workflow.

Most fixes will consist of either Java Files and SQL scripts or both.

* All java and sql files are to be committed to github (as per provided training)
* All sql files must be placed in the script flush queue detailed below.

Once the files have been placed in the queue and github, the next scheduled build will do the following *(currently builds are scheduled to run daily @ 10:30 AM EST, followed by deployments to all test environments)*:

1. Build a war file with all committed java changes
2. Deployments to all environments in the promotion path. A Deployment consists of
   1. Executing/ Flushing of all sql scripts
   2. Deploying of the war file generated in #1.

**Process to place files in the queue:**

You must drop your changed/written SQL scripts in the below location always. If their SQL script-1 is dependent on SQL Script-2, then place the script-2 first and then script-1 or else if you drag and drop all files, system will allow you to sort files in the right order in a pop-up window. Once the script files are dropped, V3 automatically prepends timestamp to each script file.

The bamboo CI/CD pipeline will use these timestamps to determine the order to flush scripts.

AZPSPRS TRUNK Q site: <https://scripts.vitechinc.com/s3list.php?dir=AZPSPRS/PG_TRUNK_QUEUE/_(OPEN-CLIENT-SCRIPTS-ONLY)/>

**Example:**

For example, if a report is being developed. For this you must prepare a script to register an entry in report\_definition, Business\_rule, business\_rule\_expr, parameter\_group and parameter\_group\_detail table for example ticket AZPSPRS-100 / EmployerReport.

Once the script is ready. Let’s azpsprs-100-employer\_report-1.sql

1. Run the script azpsprs-100-employer\_report-1 in AZPSPRSDEV Instance

2. Develop the Callback java file (For Rendering Models, defaulting values, validating field values), Let’s say Java file Name: EmployerReportTest.java and bean registered in employer\_report\_config.xml

3. Design the report design and the final file is: employer-report.jrxml

4. Integrate all and test, and if all looks good. Please follow below steps:

1. Create Pull Request (PR) for all above files:
2. azpsprs-100-employer\_report-1.sql
3. EmployerReportTest.java
4. Employer\_report\_config.xml
5. employer-report.jrxml
6. Attach the Unit test document to the PR
7. Add the right reviewers for type of issues/ fix as Reviewers and create Pull request.
8. Change the Ticket status from “In Development” to “Code Review” and add the PR (Pull Request URL) detail ticket step details and send it to Sidharth Kamat.
9. Once file changes look good, we will either Fail the PR or approve the PR changes. If all looks good, we will add comments in PR and in step notes to commit changes. Then Go to Step#vii)
10. If failed, close the existing PR and make changes at required places, again process starts from STEP# i)
11. Once we approve, you can ‘Merge’ changes to AZPSPRS Main Repo.
12. Place the SCRIPT-1.sql file in AZPSPRS Queue Flush folder (If you place the script in Queue flush folder, then only configuration(script) changes will be propagated to sub-sequent environments)

A screenshot of a computer

Description automatically generated

## **Refere****nces**

1. V3locity AZPSPRS CoDev Instructions



1. Vitech Code Check-in workflow([AZPSPRS-23465](https://jira.vitechinc.com/jira/browse/AZPSPRS-23465))



1. Add the .gitignore file to root of repo. This file instructs Git which type of files not be considered for Git Tracking



1. Jira Process (This is still a draft version)

