PRA Health Sciences

Getting started with Jira and GitLab

Table of Contents

1. JIRA

JIRA offers advanced Bug and issue tracking tool that's crucial for the Agile Methodology that we incorporate for our Project Development. In WorkBench project, we are using JIRA for creating Epics, Stories, Bugs ,Tasks, Sub-Tasks and track all the Sprint activities.

1.1 Access

Request access to Jira Software for the project through Project Manager.

- Once access has been granted, we will get an email invite to join the Atlassian community : Access the link and sign up with the PRA email id and password for single sign-on access.
- The following link that displays the list of Projects should be accessible : Projects - Jira (atlassian.net)
- Click the WorkBench from the projects list.

2. GitLab

GitLab is a complete DevOps platform, delivered as a single application, fundamentally changing the way Development, Security, and Ops teams collaborate and build software. From idea to production, GitLab helps teams improve cycle time from weeks to minutes, reduce development costs and time to market while increasing developer productivity. For WorkBench we are using GitLab for all the development activities and CI/CD processes.

2.1 Access

- Request access to GitLab through Project Manager : User account based on PRA email id will be created for GitLab group PRAHS.
- Also ensure we get appropriate access to the project.
  - For WorkBench developers, Developer access will be granted to PRAHS / PRAHS Development / Workbench · GitLab
  -

2.2 Getting started

- We need to set up the SSH key in-order to start the development activities.
  Please follow the instructions:
  - Install Gitbash from \\NA2SASFILE1\CHEDev\Dev\CodeCheckout\WorkBench\GitBash
  - Open Gitbash (Start->All Programs->Git->Git Bash)
  - Execute command "**pwd**" in gitbash to check if you are in correct directory which should be as follows : /c/Users/"*username*"
  - Execute this command to generate ssh key: **ssh-keygen -t rsa**
  - When prompted for location to store key, do not enter any location- it will be in default (\.ssh\..)
  - Enter pass-phrase (Nothing but password. It will be used to perform some operations eg: Push,pull,clone,etc.,). Re-confirm the pass-phrase – **KEY** will be generated.
  - Open file - C:\Users\UserName\.ssh\id_rsa.pub with a text editor Be sure that you don't copy any whitespace while copying public key's content (id_rsa.pub)

- Open gitlab.com and navigate to “edit profile” in right-top corner
- Select ‘SSH Key’ from left side panel
- Paste entire key copied from id_rsa.pub to Keys textbox and click "Add key" button.

- SSH key is added to the Git account. We can now clone the repository to a local folder. First time cloning should be done from GitBash as the known_hosts file should have appropriate entry.
  Please follow the steps to Clone a repository in your workspace :
  - Create a workspace folder – This could be any path like “D:\Workbench\UserName\WorkBench .
  - In GitBash, change directory to the workspace folder :

    cd /d/Workbench/UserName/WorkBench
  - Clone WorkBench repository to the workspace folder using following command

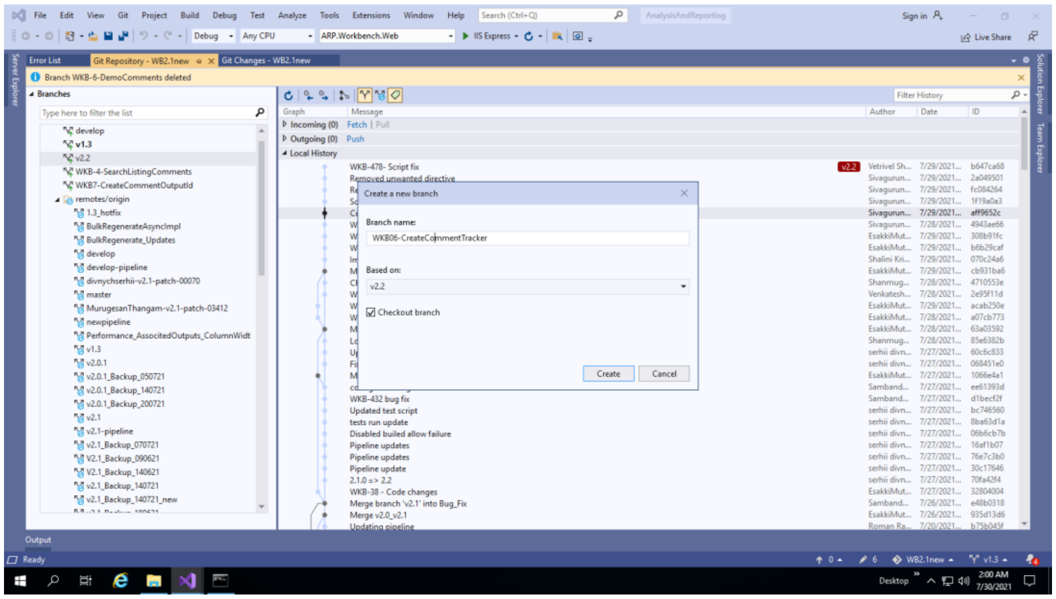
    git clone git@gitlab.com:prahs/prahs-development/workbench.git
  - It will prompt for PassPhrase and cloning will start once you have given correct pass phrase

2.3 Manage GIT Repositories

2.3.1 Create and develop code in Local Workspace

- Once the cloning is done, Open the solution file in VS2019, Git->Manage Branches
  develop : contains the local branch (It is subject to change, please verify Dev lead for current development branch name,
  remote/origin : contains the remote branch

- Create a copy of the remote repository branch in your local. For ex, if we are working in v2.2 repository, Right click on that branch from remotes/origin and Checkout. This will create a copy of the remote repository under local. From this copy of remote repository create the feature branch which we will use for code development. **Right click-> New local branch from** and give appropriate name, in our case WKBXXX_YYYY
  (XXX-> Jira ID of the work item, YYYY -> Short issue description)



Branch WKB-6-DemoComments deleted

Branches

- develop
- v1.3
- v2.2
- WKB-4-SearchListingComments
- WKB7-CreateCommentOutputId
- remotes/origin
  - 1.3_hotfix
  - BulkRegenerateAsyncImpl
  - BulkRegenerate_Updates
  - develop
  - develop-pipeline
  - divnychserhii-v2.1-patch-00070
  - master
  - MurugesanThangam-v2.1-patch-03412
  - newpipeline
  - Performance_AssocitedOutputs_ColumnWidt
  - v1.3
  - v2.0.1
  - v2.0.1_Backup_050721
  - v2.0.1_Backup_140721
  - v2.0.1_Backup_200721
  - v2.1
  - v2.1-pipeline
  - v2.1_Backup_070721
  - V2.1_Backup_090621
  - V2.1_Backup_140621
  - v2.1_Backup_140721
  - v2.1_Backup_140721_new

Incoming (0) Fetch | Pull
Outgoing (0) Push
Local History

Message	Author	Date	ID
WKB-478- Script fix	Vetrivel Sh...	7/29/2021...	b647ca68
Removed unwanted directive	Sivagurun...	7/29/2021...	2a049501
Re	Sivagurun...	7/29/2021...	fc084264
Sc	Sivagurun...	7/29/2021...	1f19a0a3
Co	Sivagurun...	7/29/2021...	aff9652c
W	Sivagurun...	7/28/2021...	4943ae66
W	EsakkiMut...	7/29/2021...	308b91fc
W	EsakkiMut...	7/29/2021...	b6b29caf
Im	Shalini Kri...	7/29/2021...	070c24a6
M	EsakkiMut...	7/29/2021...	cb931ba6
Cl	Shanmug...	7/28/2021...	4710553e
W	Venkatesh...	7/28/2021...	2e95f11d
W	EsakkiMut...	7/29/2021...	acab250e
W	EsakkiMut...	7/28/2021...	a07cb773
M	EsakkiMut...	7/28/2021...	63a03592
Lo	Shanmug...	7/28/2021...	85e6382b
Up	serhii divn...	7/27/2021...	60c6c833
Fi	serhii divn...	7/27/2021...	068451e0
M	EsakkiMut...	7/27/2021...	1066e4a1
co	Samband...	7/27/2021...	ee61393d
WKB-432 bug fix	Samband...	7/27/2021...	d1becf2f
Updated test script	serhii divn...	7/27/2021...	bc746560
tests run update	serhii divn...	7/27/2021...	8ba63d1a
Disabled builed allow failure	serhii divn...	7/27/2021...	06b6cb7b
Pipeline updates	serhii divn...	7/27/2021...	16af1b07
Pipeline updates	serhii divn...	7/27/2021...	76e7c3b0
Pipeline update	serhii divn...	7/27/2021...	30c17646
2.1.0 => 2.2	serhii divn...	7/27/2021...	70fa42f4
WKB-38 - Code changes	EsakkiMut...	7/27/2021...	32804004
Merge branch 'v2.1' into Bug_Fix	Samband...	7/26/2021...	e48b0318
Merge v2.0_v2.1	EsakkiMut...	7/26/2021...	935d13d6
Updating pipeline	Roman Ra...	7/20/2021...	b75b045f

Create a new branch

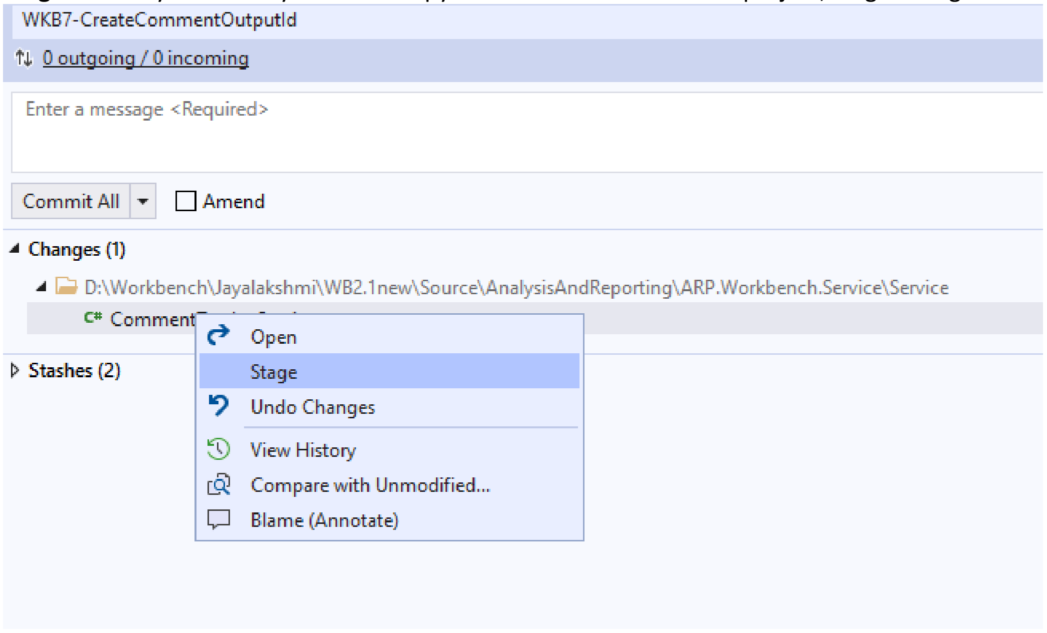
Branch name: WKB06-CreateCommentTracker

Based on: v2.2

[✓] Checkout branch

Create Cancel

- Check-out the newly created feature branch and do the appropriate changes.
- **Stage** : When you're ready to save a copy of the current state of the project, stage changes.



WKB7-CreateCommentOutputId

0 outgoing / 0 incoming

Enter a message <Required>

Commit All | ☐ Amend

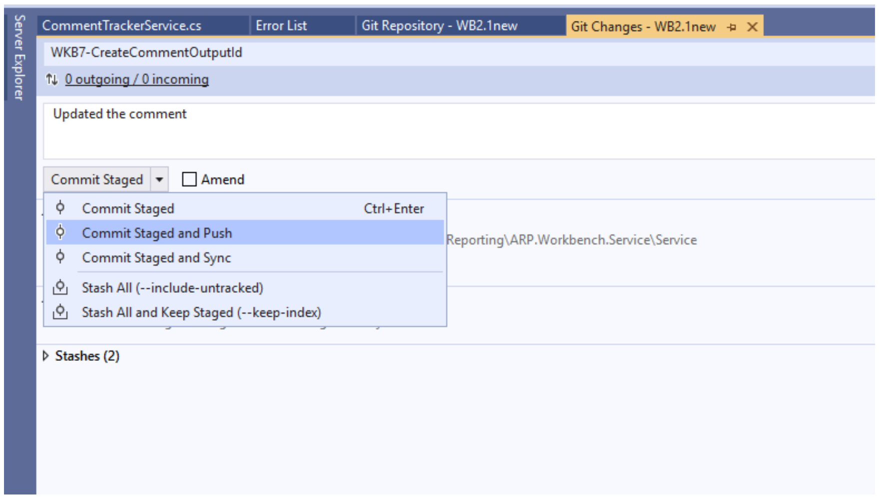
Changes (1)

D:\Workbench\Jayalakshmi\WB2.1new\Source\AnalysisAndReporting\ARP.Workbench.Service\Service

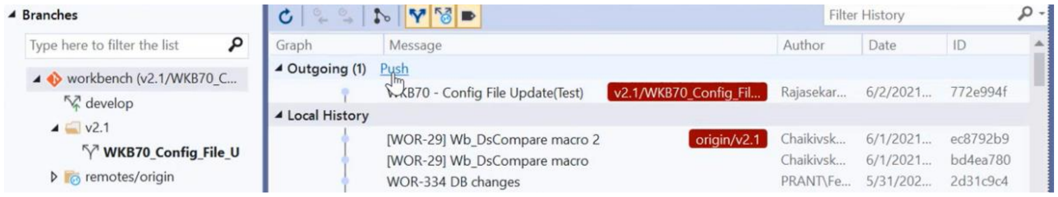
Comment

Stashes (2)

- **Commit** : The changes are still in local. We need to push the changes to remote repository. Use the following option:



Commit Staged : This only commits the changes, We have to Push the changes, which can be done later.



Graph	Message	Author	Date	ID
	KB70 - Config File Update(Test) v2.1/WKB70_Config_Fil...	Rajasekar...	6/2/2021...	772e994f
	[WOR-29] Wb_DsCompare macro 2 origin/v2.1	Chaikivsk...	6/1/2021...	ec8792b9
	[WOR-29] Wb_DsCompare macro	Chaikivsk...	6/1/2021...	bd4ea780
	WOR-334 DB changes	PRANT\Fe...	5/31/202...	2d31c9c4

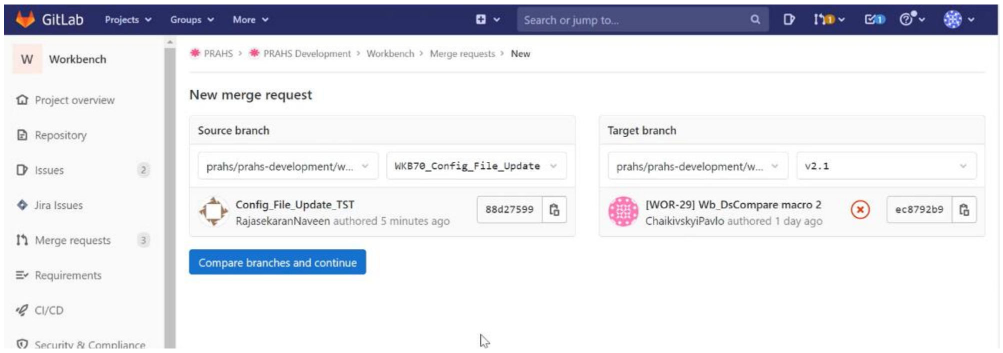
Commit and Push : Commits and Creates a remote branch. The changes are now available in Server.

Commit and Sync : Pulls the latest code and then pushes the code to server .

- If we are using a feature branch for long time, It is a good practice to ensure the local code is up-to-date. In-order to do that:
  Check-out the local copy of the main remote repository branch and pull latest to ensure the code is up to date.

  Then check-out the feature branch where we did the code changes (WKBXXX_YYYY). Right click and chose Merge v2.2 to WKBXXX_YYYY option. (If source branch: v.2.2 and Target branch : WKBXXX_YYY)
- Now Push the committed changes.
- After pushing, changes are available in Remote repository. This needs to be merged with the main remote repository branch for ex. V2.2.

- **Merge**: Select the appropriate repository in Gitlab and Create Merge Request**.** Select appropriate Source and Target branch.



New merge request

Source branch	Target branch
prahs/prahs-development/w... · WKB70_Config_File_Update	prahs/prahs-development/w... · v2.1
Config_File_Update_TST — RajasekaranNaveen authored 5 minutes ago — 88d27599	[WOR-29] Wb_DsCompare macro 2 — ChaikivskyiPavlo authored 1 day ago — ec8792b9

- Fill in the fields :

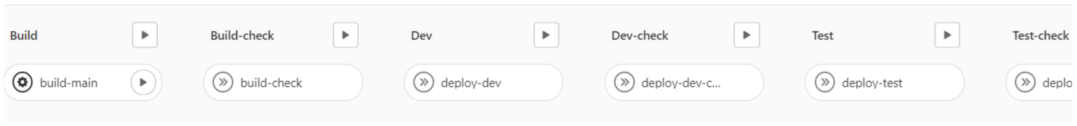
  Description : Describe the change, Assignee : Developer, Reviewers : add Reviewers, Approvals required : 1, Delete source branch when merge request is accepted : Yes
  With these “Create Merge Request”. The reviewers will be notified and once a reviewer approves, “Merge” will be enabled. Once done changes will be merged to develop(release branch:subject to change) branch.

  Note : Merge will not be enabled If the branch to be merged is not in sync with the latest updates in target branch(v2.2 in our case). Try to use Rebase option available in the GitLab. This will update sync the code as per the latest version in repository. Sometimes, say in case of merge conflicts, it may prompt to manually merge the changes in local and commit. Code changes are merged to the main repository now.

2.4 Build/Deployment

- In the GitLab site, There will be an option called Pipelines under CI/CD. Open the appropriate Pipeline which needs to be built and deployed. (Note: Pipeline will be created for every commit in Gitlab, we need to chose the pipeline corresponding to the Merge operation that we did in previous step).

  When we open the Pipeline, we can see the following screen using which we will have to manually trigger the steps to build and deploy :



- Build-Main – Builds the code. Displays build errors if any. This may include failed unit-tests, code-coverage not being met. Build-check is triggered automatically following the Build
- Deploy-Dev - Trigger this button for Dev Deployment. This will deploy latest code base to development environment. Deploy-Check is triggered automatically.
- Similarly Deployment options are available for higher environments like Test, Beta,DryRun, UAT and Prod.
  Note : The ability to build and deploy are made user specific. For ex, developers will have the Build and Deploy -Dev buttons enabled, while Testers alone will have the option to Deploy-Test.