Git workflow

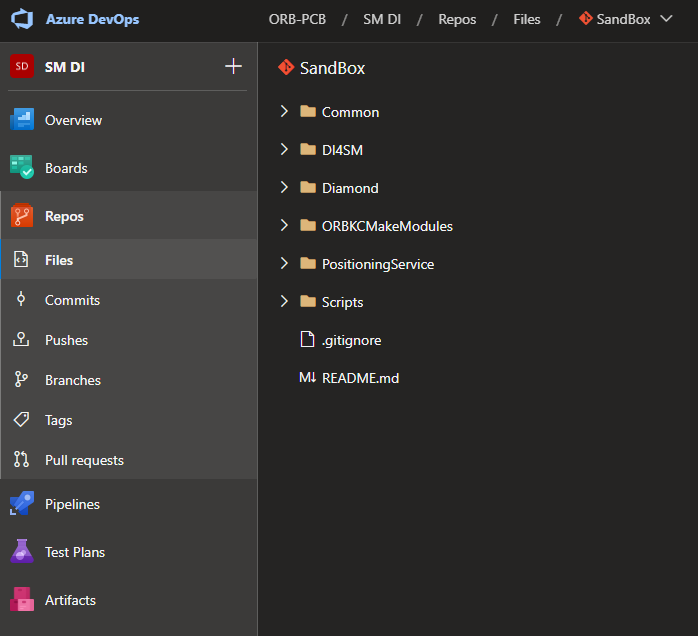
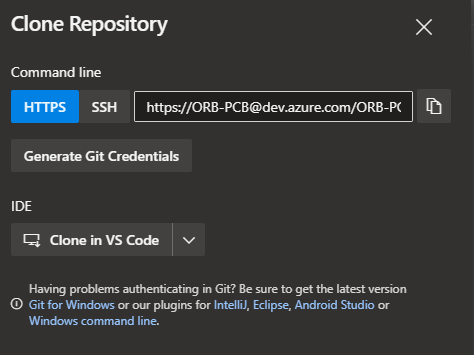
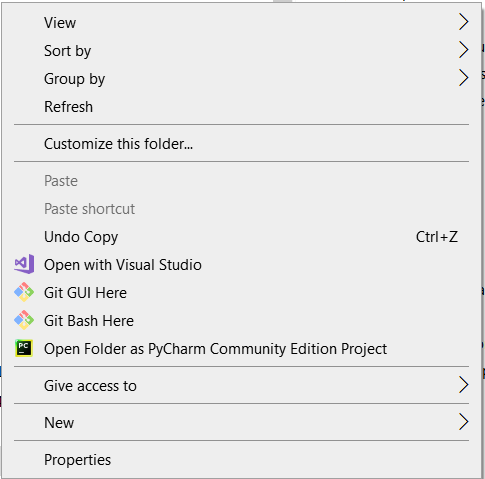
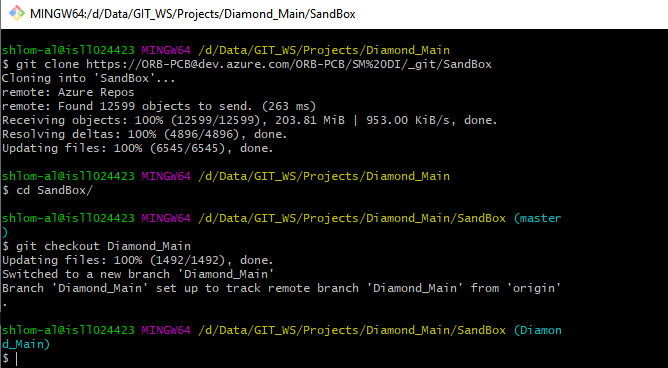
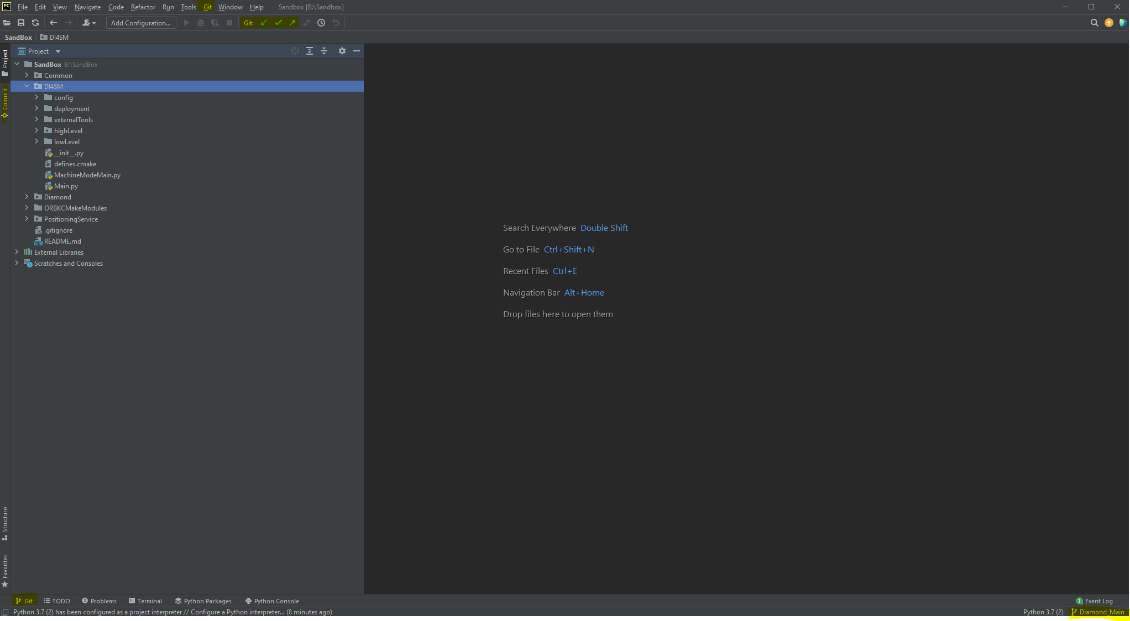
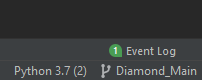
Date: 21.07.22

# Overview

## Clone repository

|  |  |
| --- | --- |
|  | Usage: git clone [url]  This command is used to obtain a repository from an existing URL. |

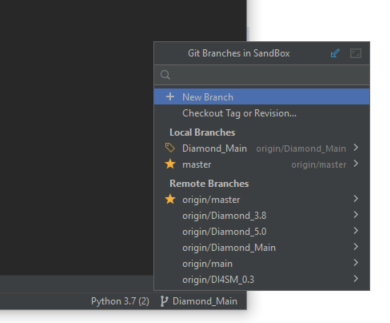
1. Go to [Azure Repo](https://dev.azure.com/ORB-PCB/SM%20DI/_git) on google chrome
2. Select the specific repository (SandBox)
3. Select Diamond\_Main branch
4. Click on clone button (right side on top)
5. Under sw folder create folder for the specific repository and main branch for example: Diamond\_Main, DI4SM\_0.3, Diamond\_0.5
6. Open Git bash from the new Diamond Main folder:
   1. Right-click from into the folder
   2. Choose Git bash here
7. Into Git bash run:
   1. git clone <https://ORB-PCB@dev.azure.com/ORB-PCB/SM%20DI/_git/SandBox>
   2. cd SandBox/
   3. git checkout Diamond\_Main
8. Create script runner for new env - D:\EnvScripts\Run\DI4SM\run\_pycharm\_release\_local.bat \\Isll024423\git\_ws\Projects\Diamond\_Main

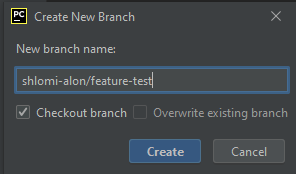


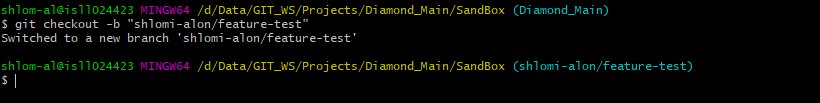
## Created a branch and checked it out in local repository and remote repository

|  |  |
| --- | --- |
|  | git checkout -b "shlomi-alon/feature-test”  This command creates a branch but and switch you to that branch. |

Create a branch where you can add feature development.  
When you are ready to make these features known to all, you can merge the changes into your azure DevOps repository and then delete the no-longer-needed branch.

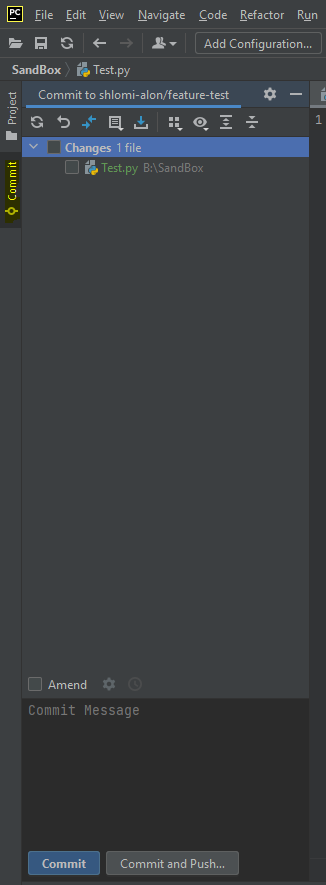








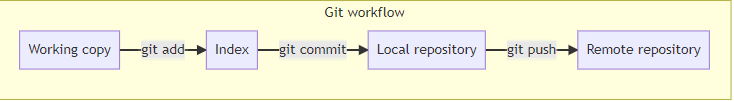
## Made a change in the new branch

Now you can development your feature by creating new files or change files   
For example create test.py file, save and close it.

## Committed the change to the new feature branch

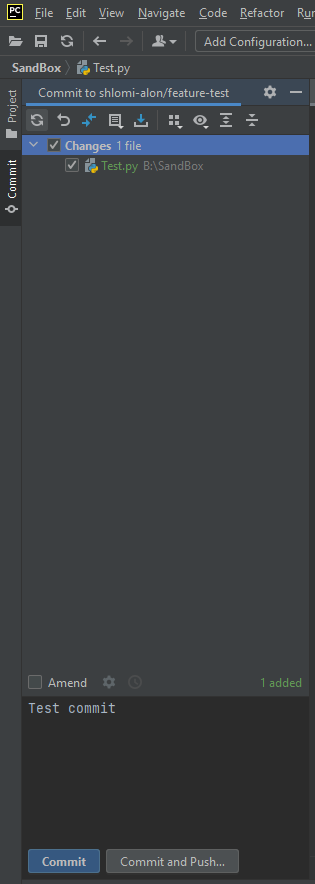
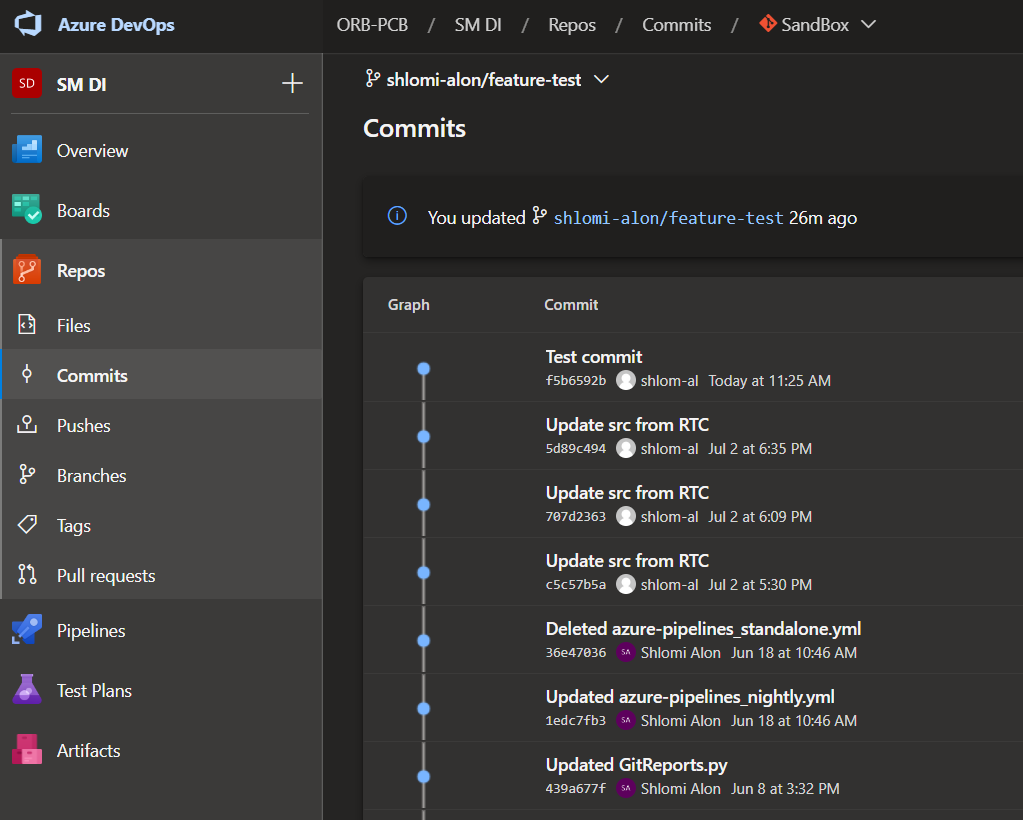
|  |  |
| --- | --- |
|  | Usage: git add [file]  this command will take any changed and untracked files in the repo and add them to the repo and update the repo's working tree.  Usage: git commit -m “[ Type in the commit message]”  This command records or snapshots the file permanently in the version history.  Usage: git push origin [shlomi-alon/feature-test]  This command sends the branch commits to your remote repository (origin shlomi-alon/feature-test) |

Save changes



* **Commit** Button – save only in local history
* **Commit and Push..** Button - save local and remote history

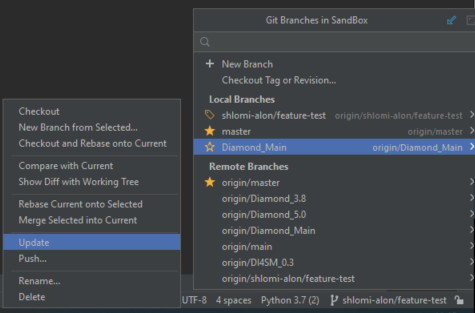
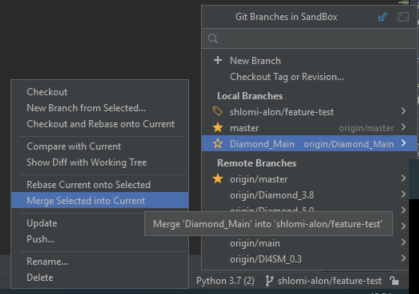
Click on “Commit and Push..” -> “Push”  
If you navigate to azure repo into the repository and into your branch you can see the commits on the server.

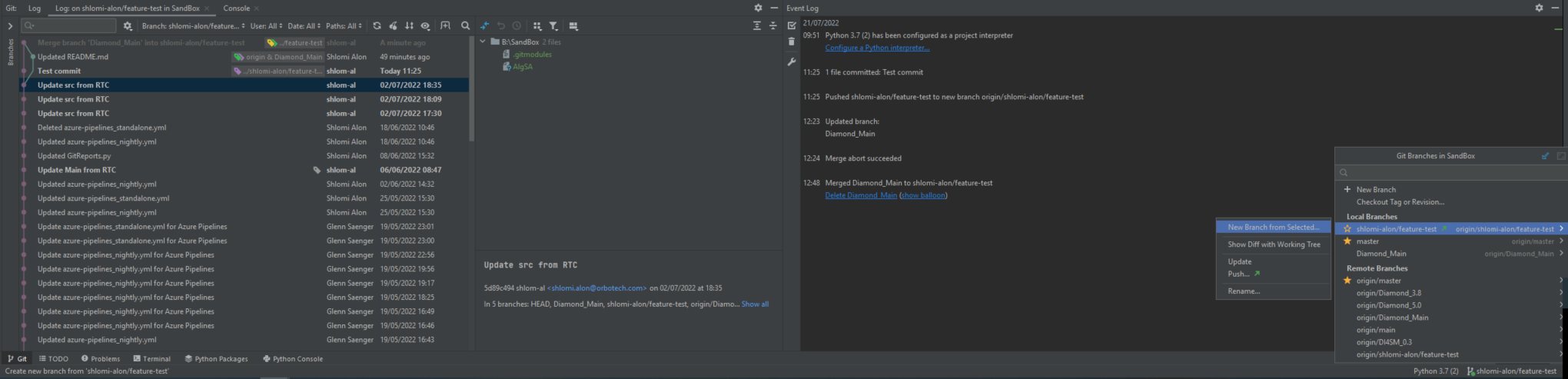


## Sync local feature branch from remote Diamond\_Main

Now that "shlomi-alon/feature-test" is ready for delivery, you can merge "shlomi-alon/feature-test" branch into the "Diamond\_Main" branch that you checked out from there.

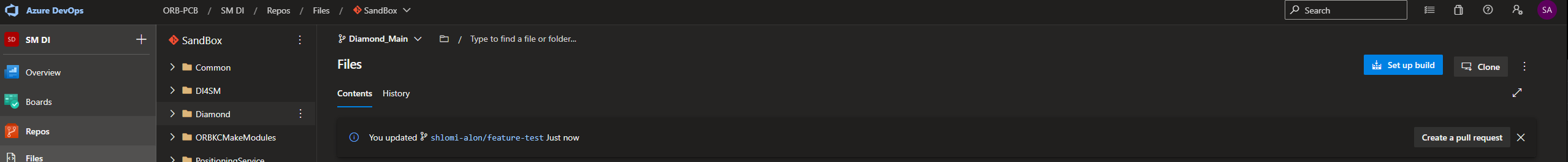
* Update Diamond\_Main branch from the remote server.
* From current feature branch (shlomi-alon/feature-test) merge updating main branch (Diamond\_Main).
* If you have conflicts – resolve them and go back to step 4.
* Compile and test your system and fix code.
* Commit and push if needed.

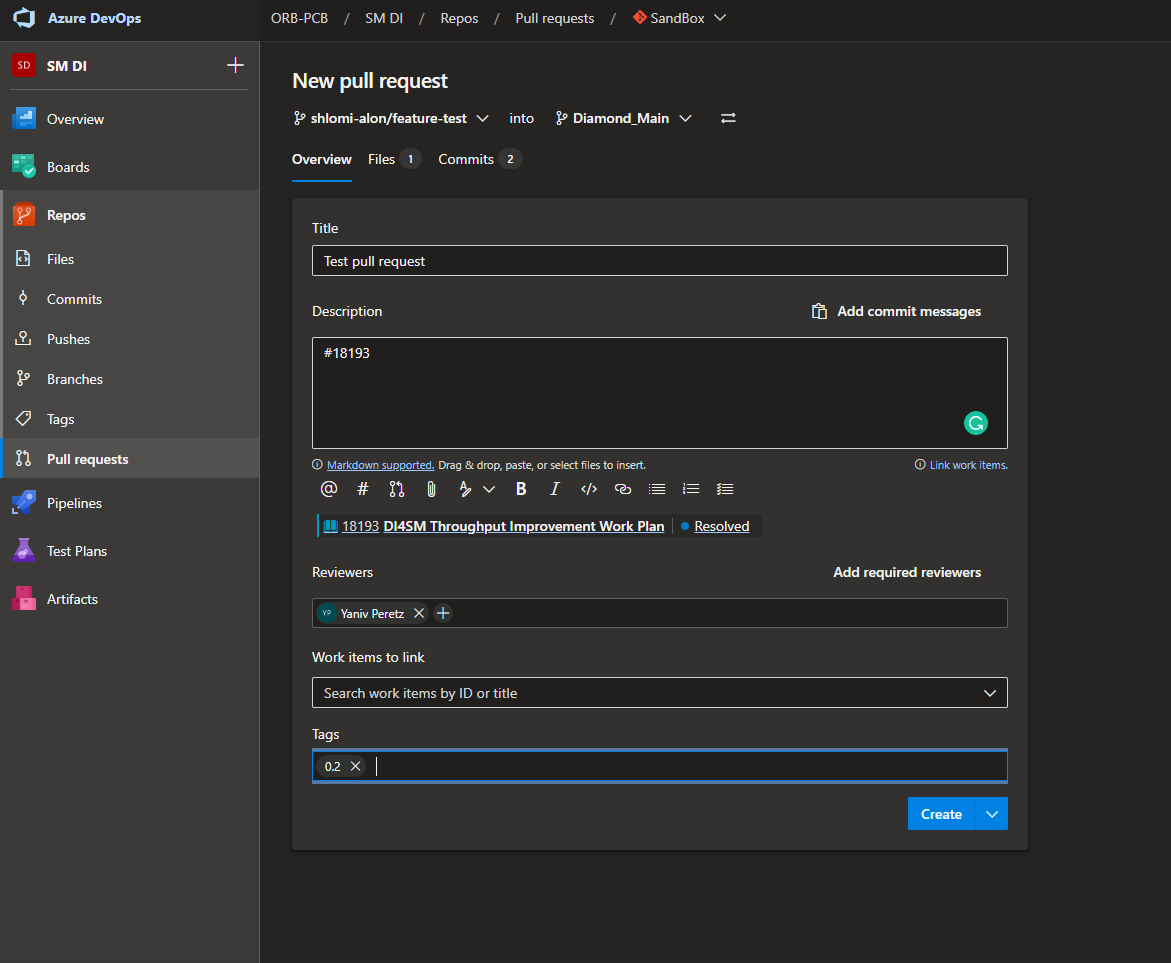
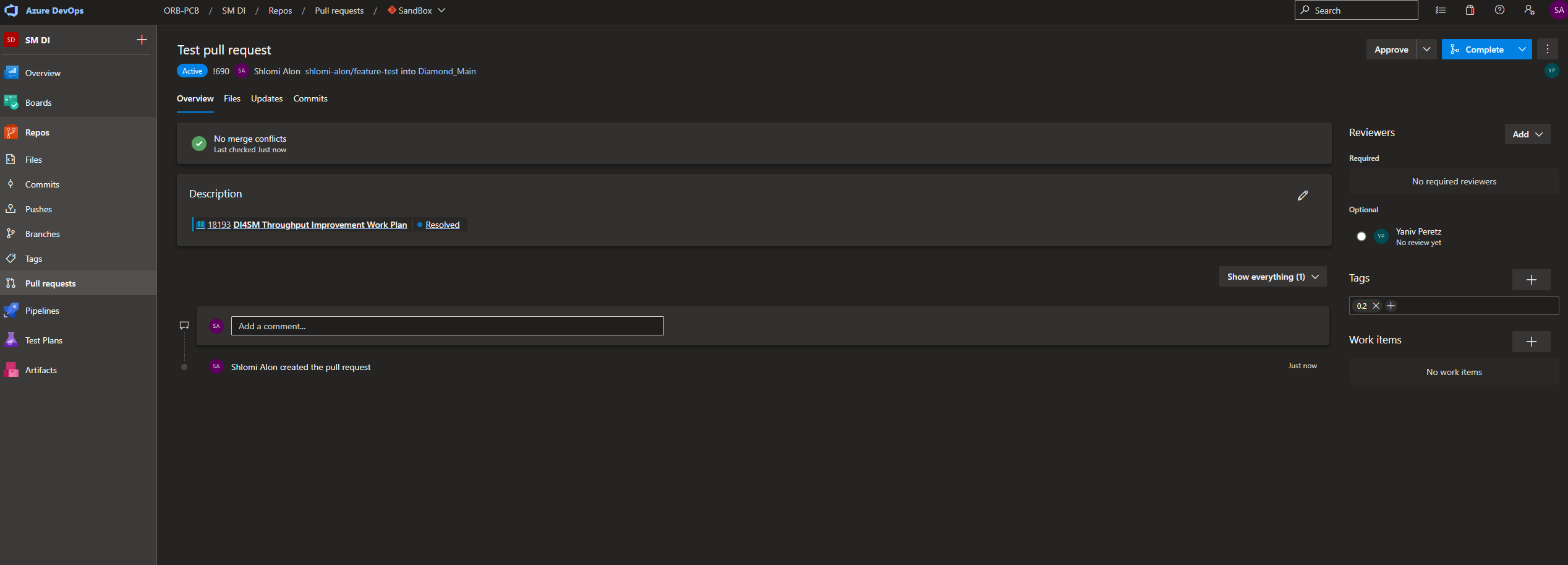


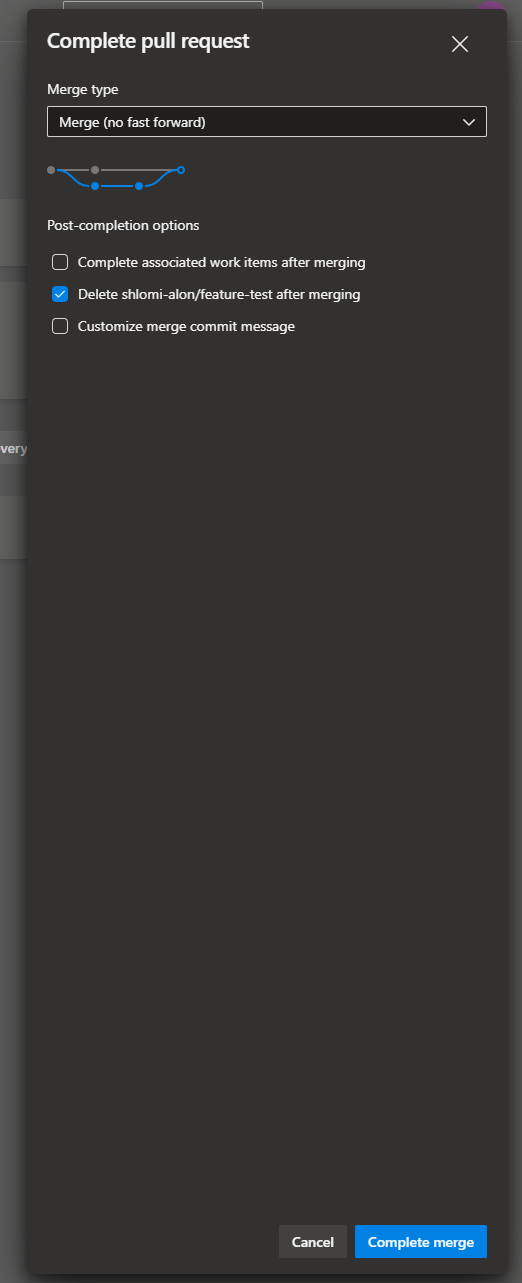


## Create pull request

1. Create pull request from <https://dev.azure.com/ORB-PCB/SM%20DI/_git/SandBox/pullrequests?_a=mine>
2. Fill the relevant fields.
3. Click on create.
4. Click on complete – for merge into history of Diamond\_Main.

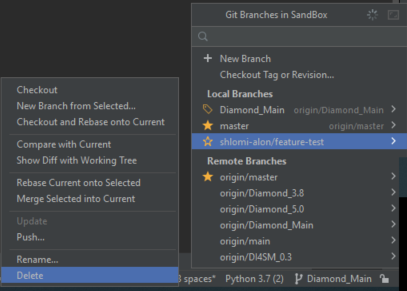
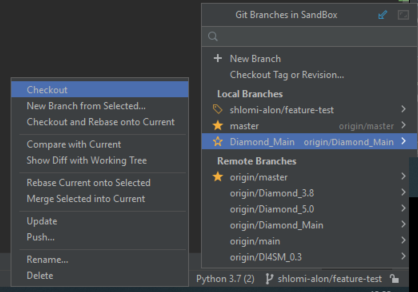
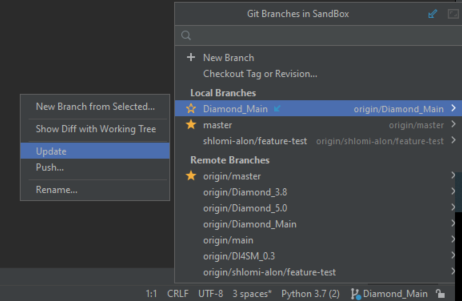






## Prepare locally for the next feature

* Checkout to Diamond\_Main branch.
* Update from remote (we merged pull request so we need to update locally).
* Delete local shlomi-alon/feature-test branch.
* Go back to step 2 for new feature branch.

## Important notes

* This guide refers to the main branch as Diamond\_Main, but it's relevant for all main branches (DI4SM\_0.3, Diamond\_3.8, Diamond\_5.0)**.**
* The merging of branches should be continuous as we do it today in RTC (See diagram below)

