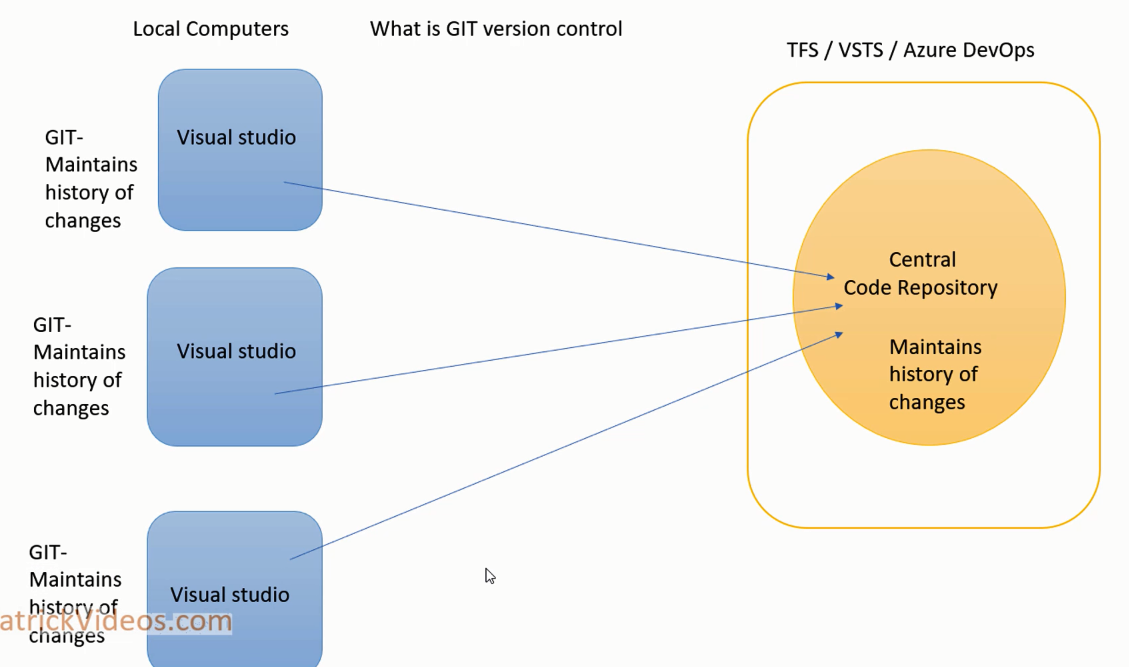


Every software project need above all. Azure DevOps is three in one tools that provide above all.

**TFS:** Team foundation server is tool provided by Microsoft, which do all above three things, but it requires on-premises installation. Means you need to install the software in your company.

**VSTS:** Cloud version ofTeam foundation server, which has renamed and now known as Azure DevOps.

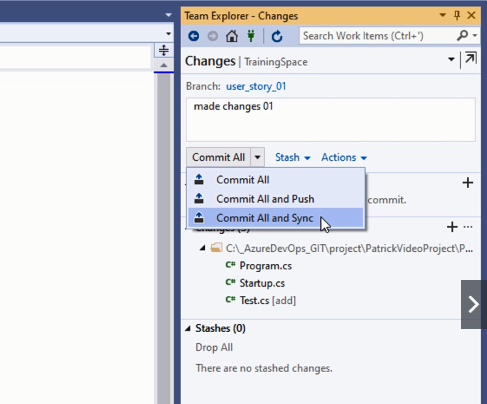
****

1. **So now where the picture of Git Comes:** Previously developers’ works and check-in the code direct to central repository, which maintain the version of code.

But later developers found the need of local version control, this local version control call the git. So GIT maintain the history of changes locally.

And central repository (TFS/VSTS/Azure DevOps) maintains the history of who made what changes.

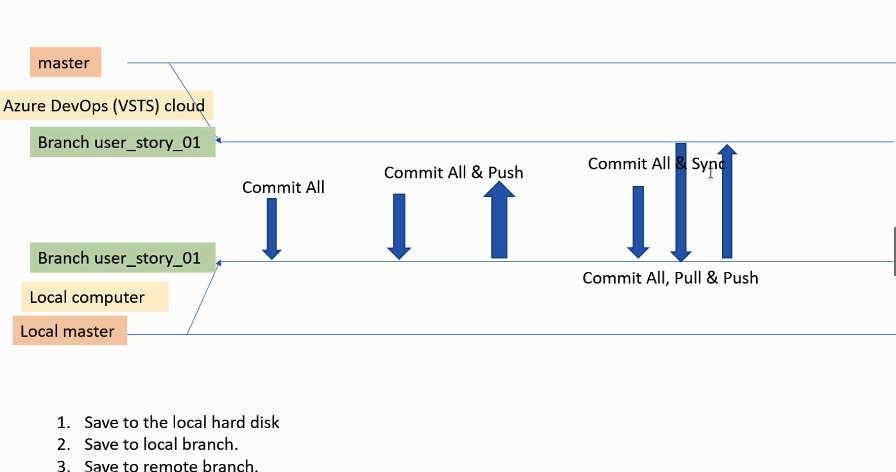
1. **Now Create a Project in Visual Studio 2017:** Created .Net Core MVC application
2. **Save work and commit:** When you did some changes - > and finally trying to move them further, We have three option as shown below



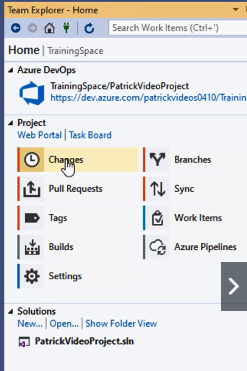
Commit All 🡪 Save your changes to local branch

Commit All and Push 🡪 It will save the changes to local branch and push it to remote branch. (It may cause the problem of merge conflict)

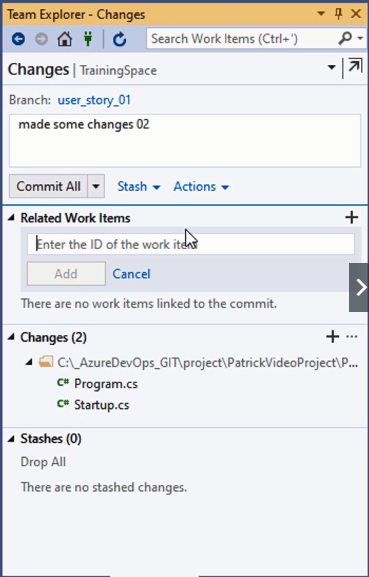
Commit All and Sync 🡪 It will save the changes to local branch, then pull the changes from the remote branch to local branch, and finally push the changes from the local branch to remote branch. (Recommended when many developers working on project)



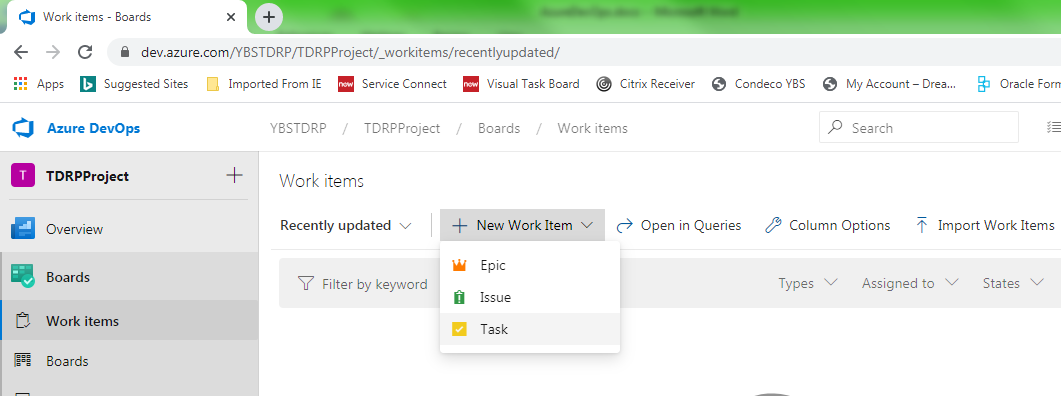
So when you done the changes 🡪 Go to Team Explorer 🡪 Home Icon on top 🡪 Click on changes



Give Comment what changes you have made, you can add the work Item as well.



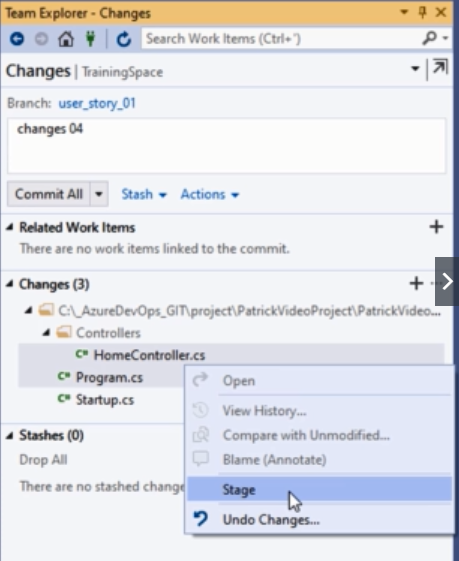
1. **Link Commits to Work Items:** This feature uses to associate the developed changes or files with particular requirement or Task, So that we aware of for what this commit has done.

****

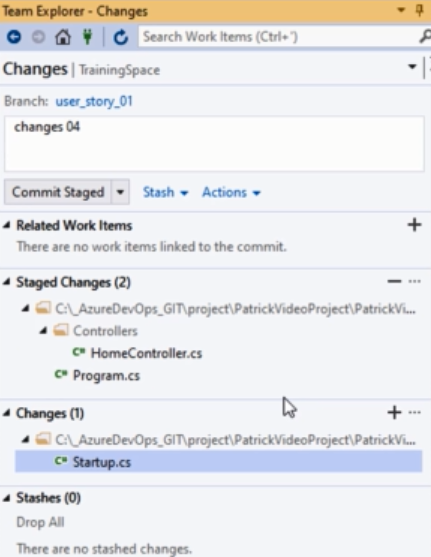
**For example:** If you have any user story or create new task or user story 🡪 and developer worked on that 🡪 than you can define the commit with associated work Item

1. **Stage:** When you working on project and you made some changes in the files 🡪 All these files known as changed files, and when you commit 🡪 all these files checked-in

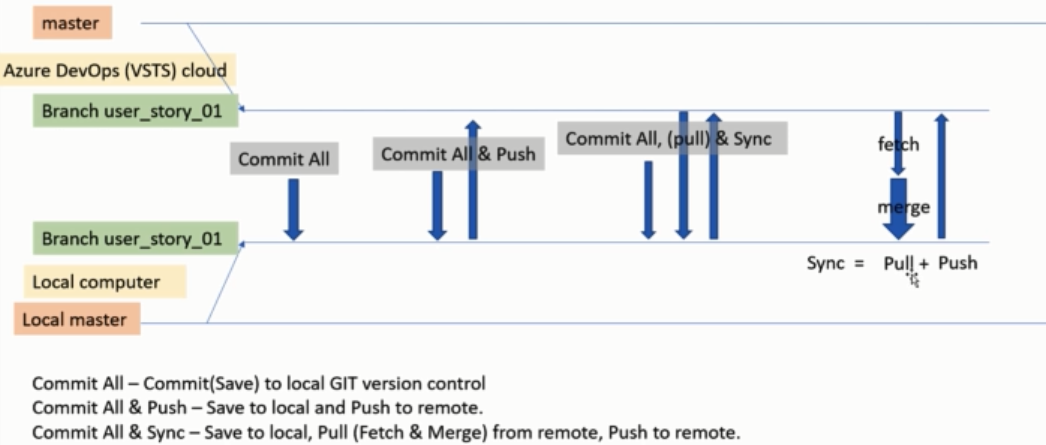
But what if you do not want to check-in some specific file? Here comes the stage

****

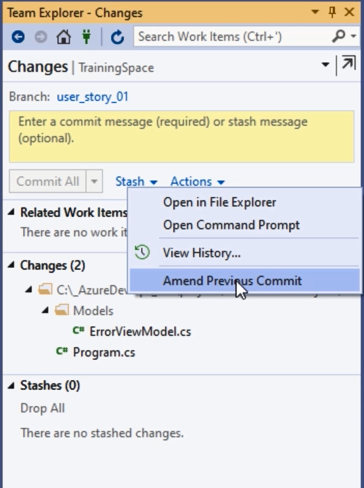
So Select the file 🡪 right click and click stage 🡪 so when you stage, only the stage file go in.



Pull 🡪 Fetch and merge the changes from the remote



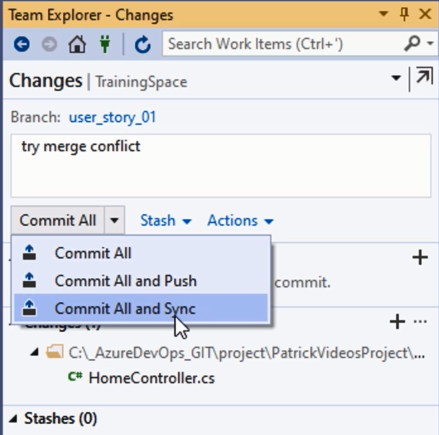
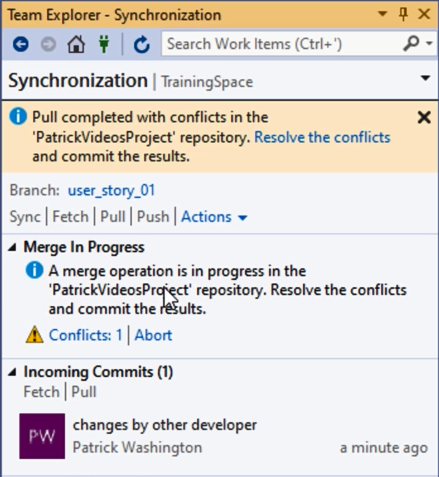
1. **Amend Previous Commit:** Suppose if you have commit some files for the user story, but forget to commit some for same task 🡪 You don’t want to add new commit and prefer to use previous one 🡪 then prefer use Amend Commit



1. **Merge Conflict:** Suppose two developers working on same file.

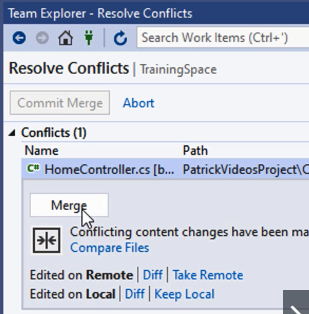
Example: I made change in Controller from visual studio, and made change in same controller from remote branch.

When you go for commit in Team Explorer 🡪 Commit All and Sync 🡪 It will show the conflict

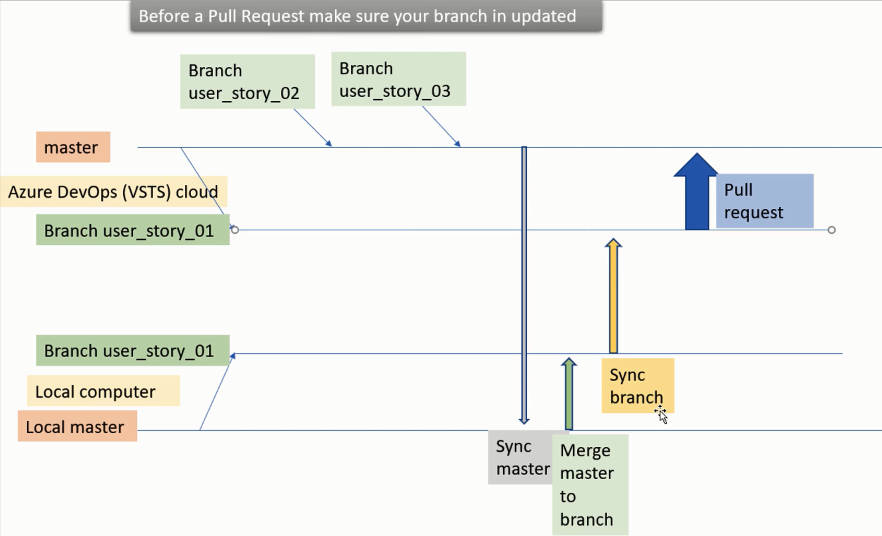
 🡪 

So Click on Conflicts 🡪 You can compare the files 🡪 If you happy with change 🡪 Merge it

Else you can Take Remote file or Keep local

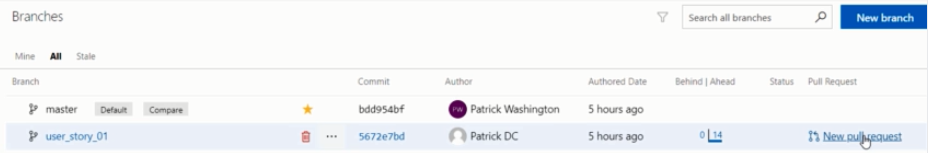


1. **Update Branch From Master:**



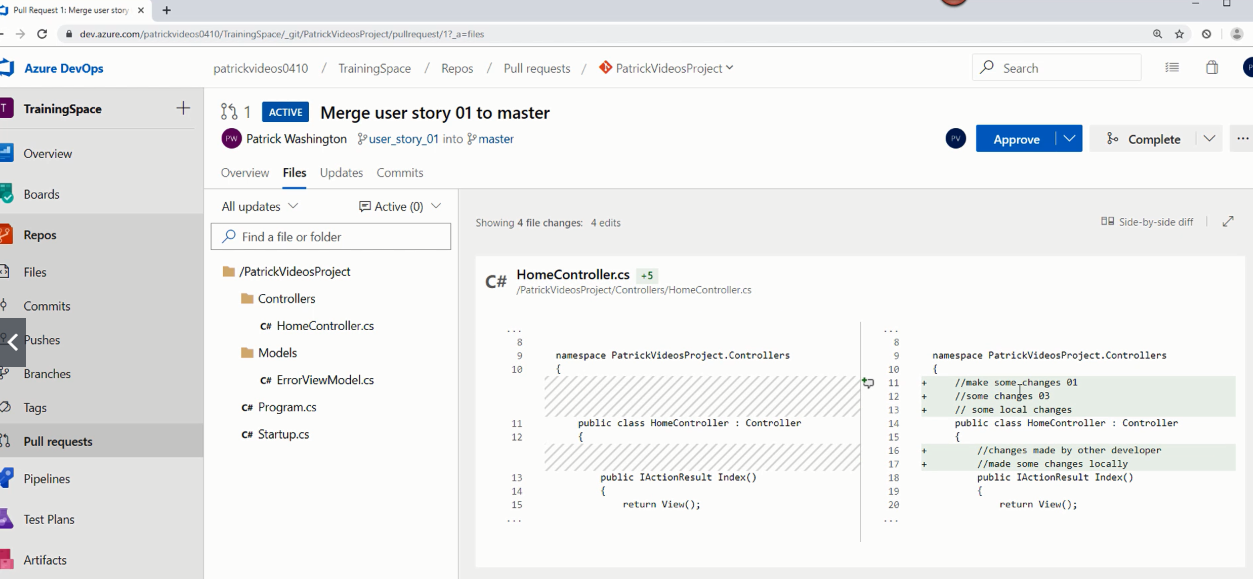
1. **Pull Request and Code Review:** Use the pull request for the code review.

Plan is to merge the remote branch with Master 🡪 the way we can do this by pull request



Click on New Pull Request 🡪 Add reviewers name for code review

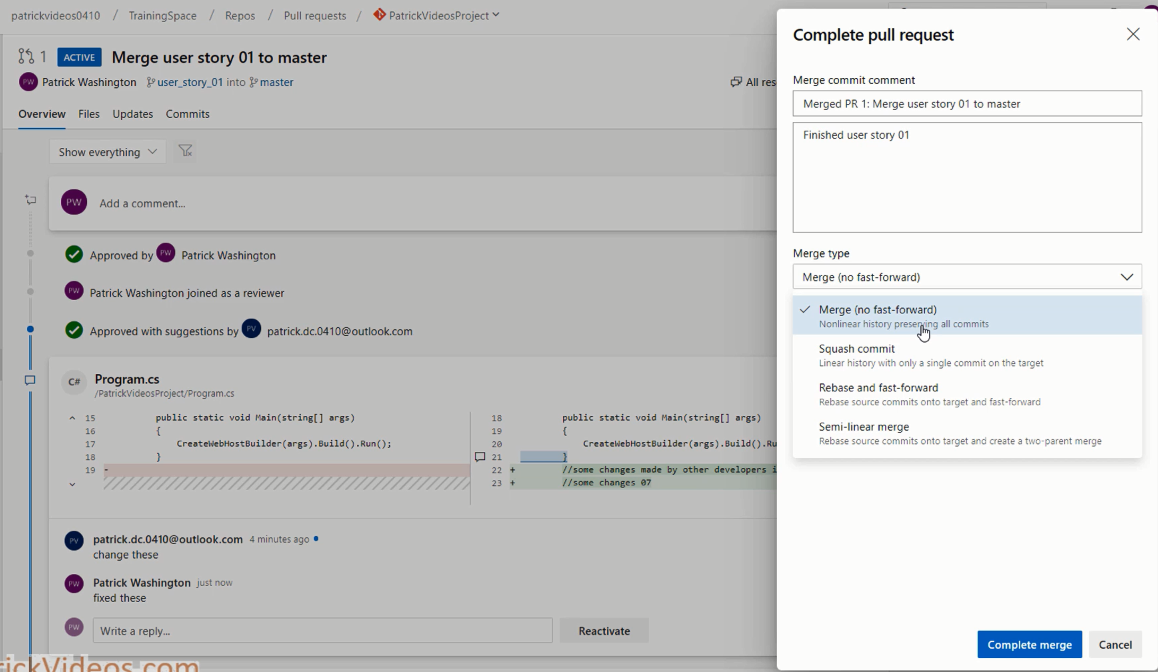
Reviewer will login 🡪 go to pull requests 🡪



Now after reviewing 🡪 He can approve the change

When approver done with Approval 🡪 Developer go pull requests 🡪 you can see the person given approval and comments made by reviewer, you can resolve the comment

Finally 🡪 Complete the commit 🡪 Always prefer to use Squash Commit



Merge Commit 🡪 The entire commits take place in base branch will move the master, means if you did 100 commit then all of them reach to master

Squash Commit 🡪 All the changes you made will go with one commit, so master will look very clean.

1. **Branch Policy:** Branch policy help you to protect your branch from the people merging the code without approval

Merging the code without specifying for which user story this change is.

1. **Tags:** Tags help you to labelling current version of code, so that in future if you want to see the old version of code you can use it.

Or

You in production you tag previous version properly and want to rollback to previous version because of current version not working, you can restore the previous Tag.

Click on Tag 🡪 You can download the code or you can clone it.

