

Azure Repos

Azure Repos is a set of version control tools provided by **Microsoft Azure** that helps teams manage their source code. It offers two types of version control systems:

1. **Git Repositories** – A **distributed version control system** that allows multiple developers to work on the same project with branching, merging, and pull request features.
2. **Team Foundation Version Control (TFVC)** – A **centralized version control system**, where all versions of the code are stored on a central server, and developers check out files to make changes.

Key Features of Azure Repos:

- **Unlimited Git repositories:** Host private or public repositories for free.
- **Branching & Merging:** Supports Git branching strategies like feature branches, GitFlow, and trunk-based development.
- **Pull Requests & Code Reviews:** Developers can review code before merging changes.
- **CI/CD Integration:** Works with **Azure Pipelines** for automated builds and deployments.
- **Permissions & Security:** Provides role-based access control (RBAC).
- **Integration with DevOps Tools:** Works with Azure DevOps, GitHub, Jenkins, and more.

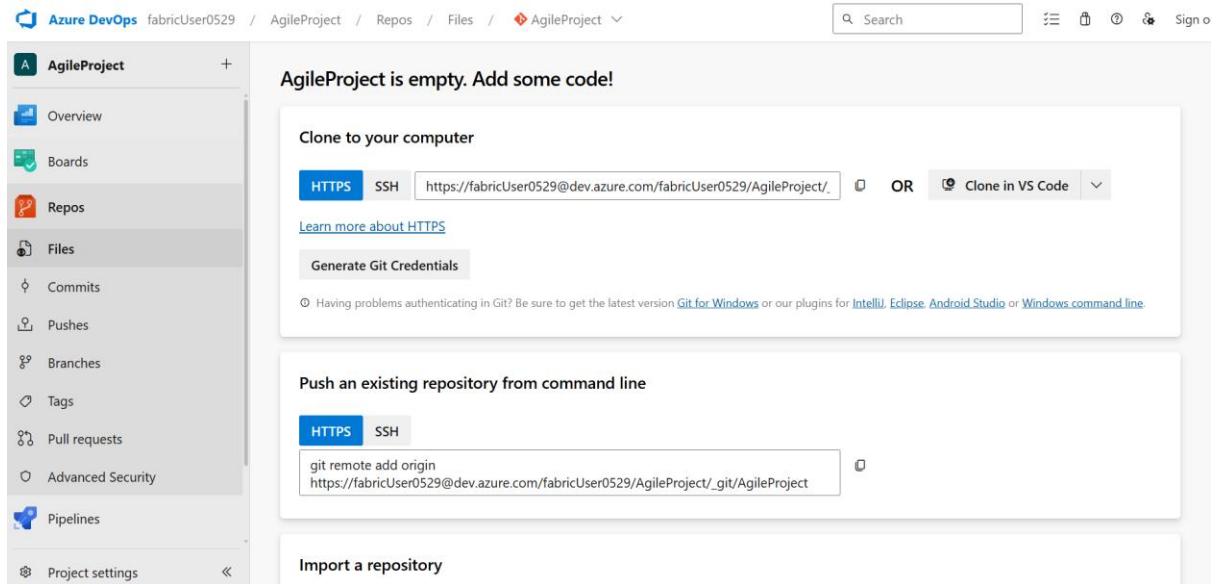
The process involves using **Azure Repos** in **Azure DevOps** for version control. First, initialize a **Git repository** in a local folder using VS Code, then link it to Azure Repos by adding a **remote origin** and pushing the code. Changes can be committed and pushed, with visibility in Azure Repos. A new **branch** is created, changes are made, and a **pull request** is raised to merge them into the main branch. Additionally, existing **GitHub repositories** can be imported into Azure Repos.

The end goal is to manage code efficiently, collaborate seamlessly, and integrate with DevOps pipelines.

To begin with the Lab

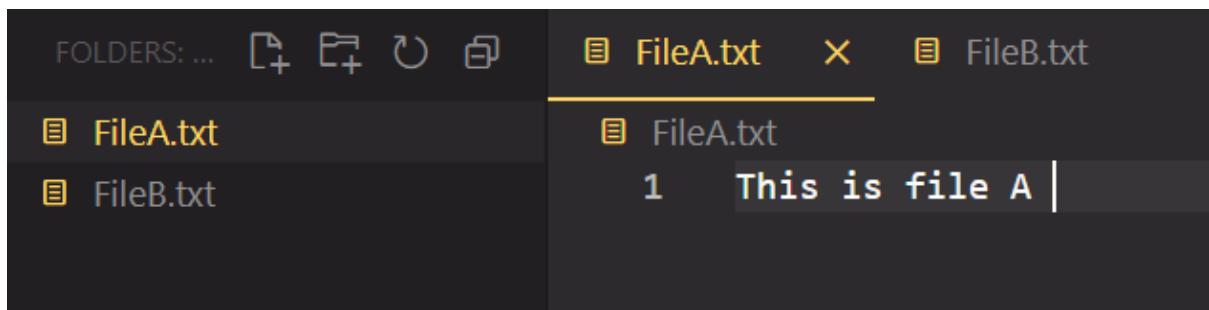
1. In this lab, we are going to look at the Azure Repos feature that is available in Azure DevOps.
2. So, open your Azure DevOps and go to Azure Repos Feature in the Agile Project.
3. When you go ahead and do this, you will see that there is a default Git repository with the same name as a project that gets created. You can now upload code, like you do with GitHub onto Azure Repos. So again, this is like a remote repository for you on the

internet. So, you could either use GitHub for hosting your code on the internet or you can make use of Azure Repos.

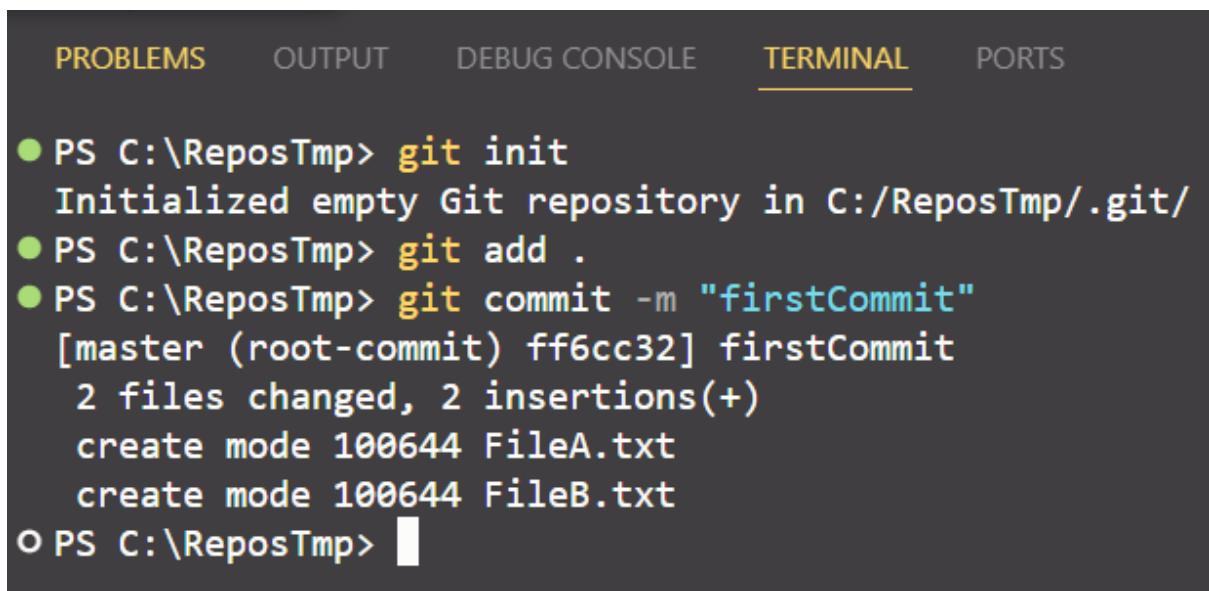


The screenshot shows the Azure DevOps interface for the 'AgileProject' repository. On the left, a sidebar lists project navigation options: Overview, Boards, Repos (selected), Files, Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, and Pipelines. The main content area displays a message: 'AgileProject is empty. Add some code!'. It includes sections for 'Clone to your computer' (with HTTPS and SSH links) and 'Push an existing repository from command line' (with similar links). A note at the bottom suggests using the latest version of Git for Windows or its plugins for IntelliJ, Eclipse, Android Studio, or Windows command line. Below these sections is an 'Import a repository' button.

4. So, now what I have done is I have created a folder and, in that folder, I have two files as you can see below.
5. Now open the terminal in VS Code and initialize an empty git repository in this folder.



6. Here you can see that in the terminal we have initialized our folder as a git repository.



The screenshot shows the VS Code interface with a dark theme. The top navigation bar has tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined, indicating it is active), and PORTS. The terminal window displays the following command-line session:
PS C:\ReposTmp> git init
Initialized empty Git repository in C:/ReposTmp/.git/
PS C:\ReposTmp> git add .
PS C:\ReposTmp> git commit -m "firstCommit"
[master (root-commit) ff6cc32] firstCommit
2 files changed, 2 insertions(+)
create mode 100644 FileA.txt
create mode 100644 FileB.txt
PS C:\ReposTmp>

- Then we are going to copy the command to add a remote origin from our Azure Repos. Once it is added then we will push our code to the Azure Repos.

Push an existing repository from command line

HTTPS **SSH**

```
git remote add origin
https://fabricUser0529@dev.azure.com/fabricUser0529/AgileProject/_git/AgileProject
```

- When you run the push command it will ask you to sign in with the Microsoft account.

```
PS C:\ReposTmp> git remote add origin https://fabricUser0529@dev.azure.com/fabricUser0529/AgileProject/_git/AgileProject
PS C:\ReposTmp> git push -u origin --all
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 281 bytes | 281.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Analyzing objects... (4/4) (3 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (52 ms)
remote: Storing index... done (34 ms)
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
To https://dev.azure.com/fabricUser0529/AgileProject/_git/AgileProject
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

- If your commit is successful then your files will be inside the Azure Repos.

Name	Last change	Commits
FileA.txt	4m ago	ff6cc32f firstCommit Pulk...
FileB.txt	4m ago	ff6cc32f firstCommit Pulk...

- We can make changes to the file commit them and push them onto the repository.

```

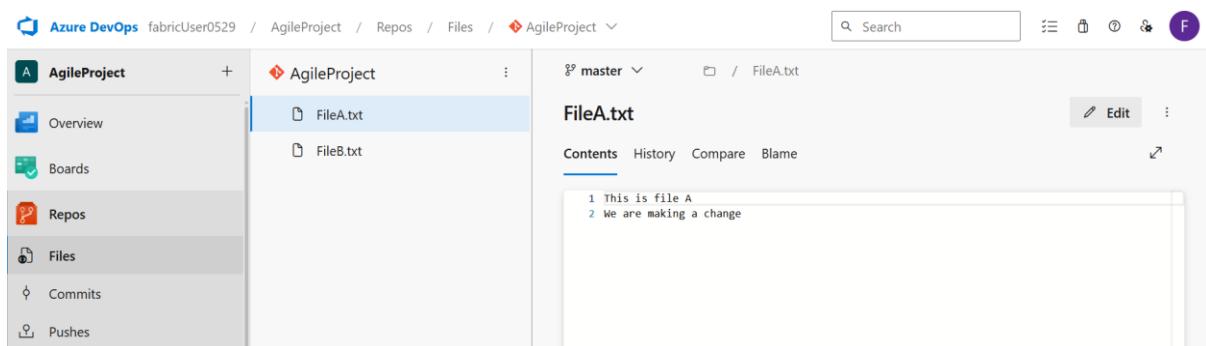
FOLDERS:... 📂 🗃 ⏪ ⏴ 🗑️
FileA.txt X FileB.txt
FileA.txt
1 This is file A
2 We are making a change

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\ReposTmp> git add FileA.txt
● PS C:\ReposTmp> git commit -m "Changed FileA"
[master addaa423] Changed FileA
  1 file changed, 2 insertions(+), 1 deletion(-)
● PS C:\ReposTmp> git push -u origin --all
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 318.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Analyzing objects... (3/3) (5 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (33 ms)
remote: Storing index... done (33 ms)
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
To https://dev.azure.com/fabricUser0529/AgileProject/_git/AgileProject
  ff6cc32..addaa423 master -> master
branch 'master' set up to track 'origin/master'.
○ PS C:\ReposTmp>

```

11. We can always see the changes in Azure Repos.



😊 Branches

12. Now we want to go ahead and create a branch in our local repository and then push all of those changes again onto Azure Repos. These sorts of operations again, are those, that you are doing when pushing your changes onto GitHub, you can also do it in Azure Repos.

13. Using the git checkout command we have created a new branch.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\ReposTmp> git checkout -b newBranchA
Switched to a new branch 'newBranchA'
○ PS C:\ReposTmp>

```

14. Now we have made a change in the File B and committed that change as well.

The screenshot shows the VS Code interface. On the left, a sidebar titled 'FOLDERS' lists 'FileA.txt' and 'FileB.txt'. The 'FileB.txt' entry is highlighted. In the center, a code editor window displays 'FileB.txt' with the following content:

```
1 This is File B
2 We are making a change in file B
```

Below the code editor is a tab bar with 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is underlined), and 'PORTS'. The 'TERMINAL' tab is active, showing a command-line history:

```
PS C:\ReposTmp> git checkout -b newBranchA
Switched to a new branch 'newBranchA'
● PS C:\ReposTmp> git add FileB.txt
● PS C:\ReposTmp> git commit -m "Changed FileB"
[newBranchA a4ce0c3] Changed FileB
 1 file changed, 1 insertion(+)
○ PS C:\ReposTmp>
```

15. In the end we pushed all the changed onto the repository.

```
● PS C:\ReposTmp> git push -u origin --all
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Analyzing objects... (3/3) (5 ms)
remote: Validating commits... (1/1) done (1 ms)
remote: Storing packfile... done (44 ms)
remote: Storing index... done (31 ms)
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
To https://dev.azure.com/fabricUser0529/AgileProject/_git/AgileProject
 * [new branch]      newBranchA -> newBranchA
● branch 'master' set up to track 'origin/master'.
branch 'newBranchA' set up to track 'origin/newBranchA'.
○ PS C:\ReposTmp>
```

16. On the browser, first refresh the page then you can see the new branch. Switch to it.

The screenshot shows a Git interface with a search bar at the top containing 'FileA.txt'. Below the search bar, there are tabs for 'Branches' and 'Tags', with 'Branches' being the active tab. Under the 'Branches' tab, there is a list of branches. The 'master' branch is listed as 'Default'. Below it, the 'newBranchA' branch is selected, indicated by a checkmark and a blue background. At the bottom of the list, there is a button labeled '+ New branch'.

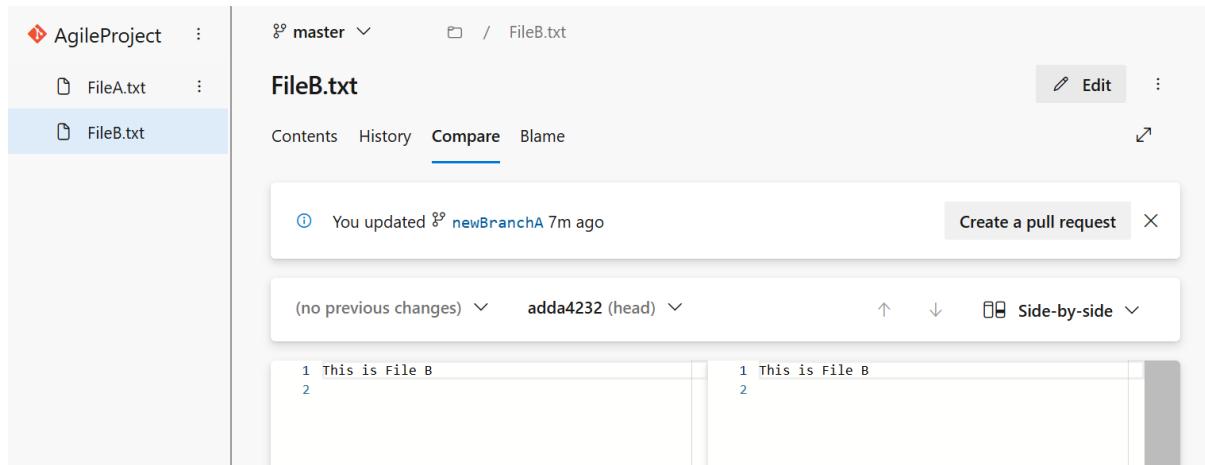
17. Here you can see the file B and its changes.

The screenshot shows a Git interface with a sidebar on the left listing files: 'AgileProject', 'FileA.txt', and 'FileB.txt'. 'FileB.txt' is currently selected and highlighted with a blue background. The main area displays the contents of 'FileB.txt' under the heading 'FileB.txt'. It includes a message: 'You updated newBranchA 5m ago' and a 'Create a pull request' button. Below this, the file content is shown with two lines of text: '1 This is File B' and '2 We are making a change in file B'. At the bottom, there is a comparison view showing the previous commit 'ff6cc32f (previous)' and the current head 'a4ce0c37 (head)'. The diff view highlights changes with red and green colors. A 'Side-by-side' comparison option is available. The bottom right corner features a large orange 'Pull Request' button.

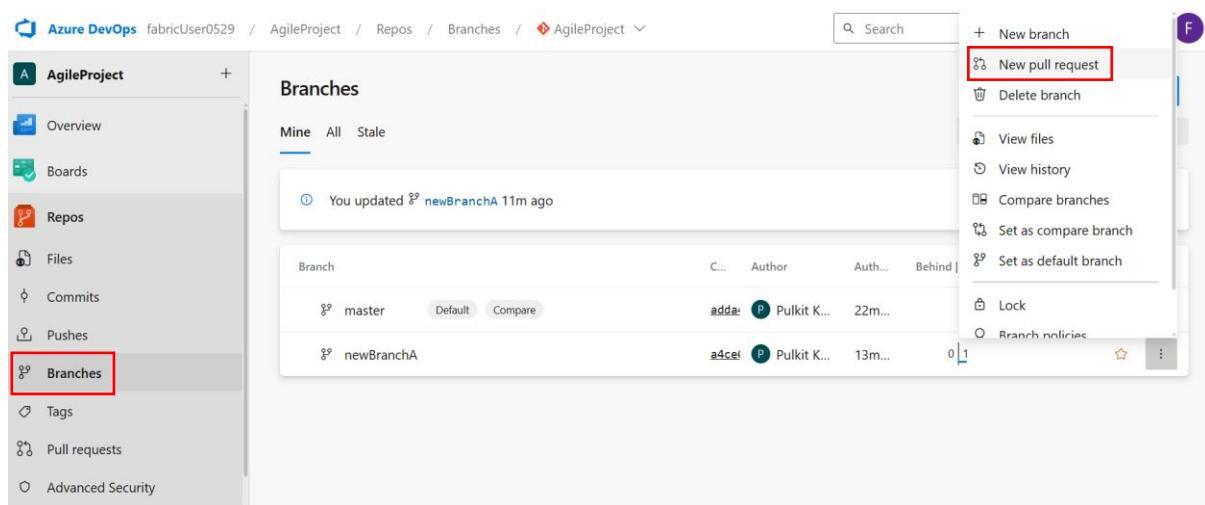
Pull Request

18. Now if you switch to your master branch which is your main branch as well, you will see that the changes made to file B are only reflected in the new branch because this is how it works.

19. So, to merge the changes together we are going to raise a pull request inside the Azure Repos.



20. In the Azure Repos go to Branches, click on three dots on the new branch and then select new pull request.



21. Then give it a title and description, click on Create.

22. It will then tell you that there are no conflicts. Click on Complete.

Changed FileB

Active !1 F fabricUser proposes to merge into [master](#) from [newBranchA](#) ⚙ ⭐

Overview Files Updates Commits

No merge conflicts
Last checked Just now

Description

Changed FileB

Show everything (1) ▾

Add a comment...

f fabricUser created the pull request

Approve ⚙ Complete ⚙ ⋮

Reviewers

Required

No required reviewers

Optional

No optional reviewers

Tags

No tags

Work items

No work items

23. Then for the merge type choose any option of your choice. We have chosen rebase because uses the previous commit messages. You can use fast merge also. Click on complete merge.

Complete pull request

X

Merge type

Rebase and fast-forward



Post-completion options

- Complete associated work items after merging
- Delete newBranchA after merging

i Rebase uses existing messages from commits

Cancel

Complete merge

24. Below you can see that the merger has been successful.

Changed FileB

Completed !1 F fabricUser proposes to merge into [master](#) from [newBranchA](#)

Overview Files Updates Commits

fabricUser completed this pull request Just now

[Cherry-pick](#) [Revert](#)

Changed FileB
a4ce0c37 P Pulkit Kumar Today at 3:16 PM

Show details

No merge conflicts
Last checked 16m ago

Description
Changed FileB

Reviewers Add ▾
Required No required reviewers

Optional No optional reviewers

Tags +
No tags

Work items +

25. Now if we go to the branches, we can see there is only our master branch because our new branch was deleted at the time of the merger. Open it and look for the changes you will find them executed.

Branches

New branch

Mine All Stale

Search branch name

Branch	C...	Author	Auth...	Behind Ahead	S...	P...
master	a4ce0c37	Pulkit K...	18m...			

AgileProject

master / FileB.txt

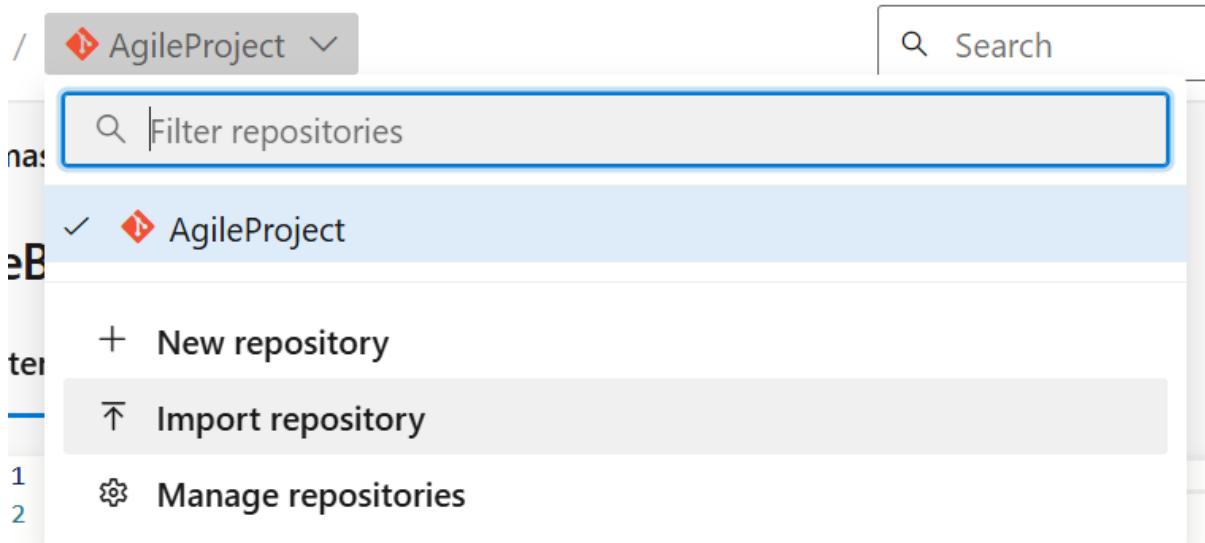
FileB.txt

Contents History Compare Blame

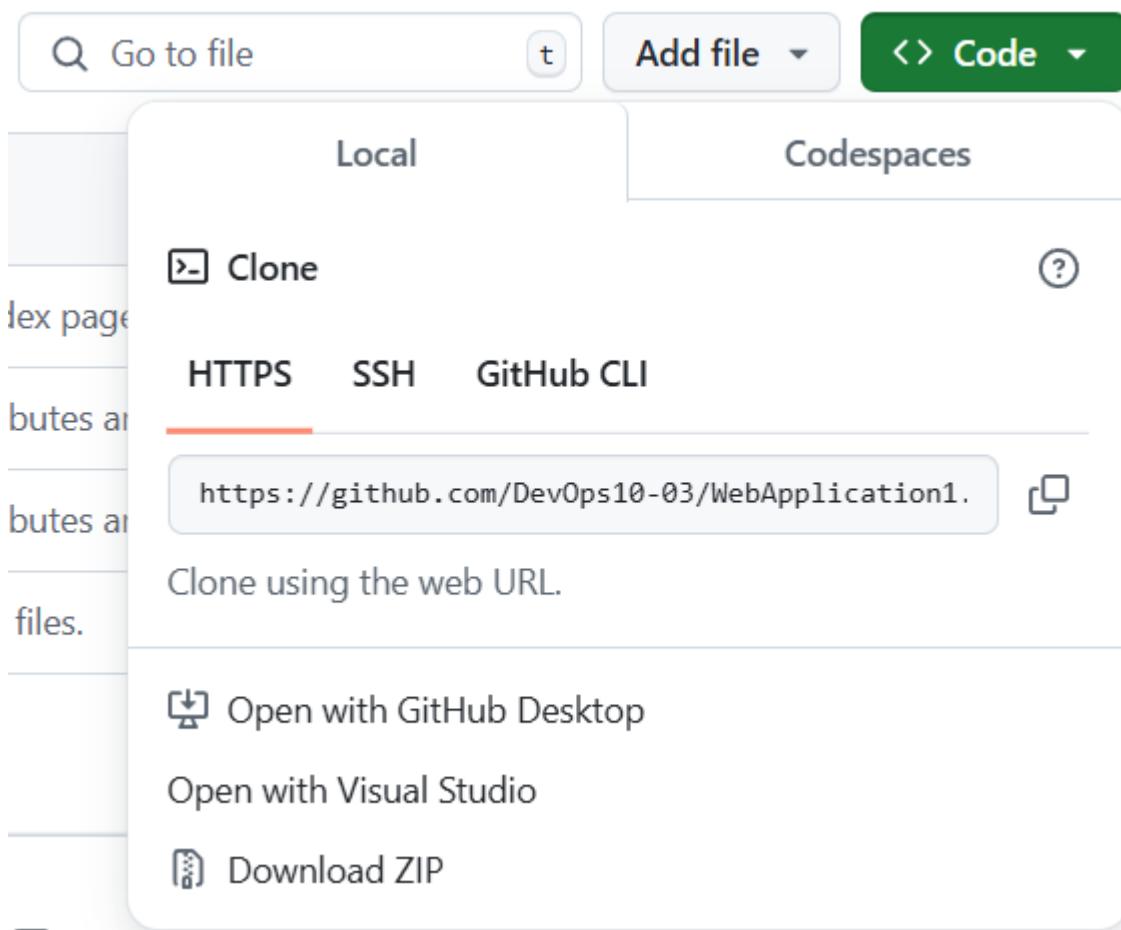
1 This is File B
2 We are making a change in file B

.importing GitHub Repos

26. Now we are going to import the GitHub repositories onto the Azure Repos.
27. For that if you look at the top of the screen there is a drop-down menu click on it and choose to import repository.



28. But before that in the previous labs we have created a Web app and pushed it onto the organization in the GitHub. So, open that repository or any repository if you have and then copy the HTTPS URL to clone it.



29. Then you need to paste the URL in the Clone URL section. After that, you need to give the username and password for your GitHub account for the authentication. You can also use PAT (Personal Access Token).

Import a Git repository

X

Repository type



Git



Clone URL *

<https://github.com/DevOps10-03/WebApplication1.git>

Requires Authentication

Username

Password / PAT *

Name *

WebApplication1.git

Cancel

Import

30. After providing the authentication click on Import and it will start importing your repository.

On its way!

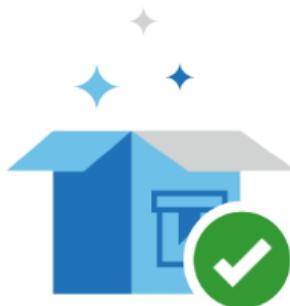


Processing request

Importing <https://github.com/DevOps10-03/WebApplication1.git>

We'll send you a notification when it's ready. For now, you can work on some other project or just take a moment to sit back, relax and enjoy your day.

Import Successful!



Congratulations! Your <https://github.com/DevOps10-03/WebApplication1.git> repository has been successfully imported.

If you are not automatically redirected to your repository page [Click here to navigate to code view.](#)

31. Once the import is completed you can see your repository.

The screenshot shows a GitHub repository interface. On the left, there's a sidebar with a tree view of the repository structure, including a folder named 'WebApplication1'. The main area is titled 'Files' and shows a list of files with their last change times and commit history. The commits are listed from newest to oldest.

Name	Last change	Commits
WebApplication1	Just now	5ed9d4bd changed Index pa...
.gitattributes	4h ago	5f0b3877 Add .gitattribut...
.gitignore	4h ago	5f0b3877 Add .gitattribut...
WebApplication1.sln	4h ago	1af1aa30 Add project files. ...

32. You can also see the history of your commits.

The screenshot shows the same GitHub repository interface, but the 'History' tab is selected. This view provides a visual timeline of the repository's commits. Each commit is represented by a blue dot on a vertical line, with the commit message, author, and date to its right. The commits listed here correspond to the ones shown in the 'Files' tab.

Graph	Commit	Pull Request	Status
	changed Index page 5ed9d4bd Pulkit-Kumar-0 Just now		
	made changes to index page ce092a8d Pulkit-Kumar-0 Today at 3:34 PM		
	Changed the Index Page 6742cb41 P Pulkit Kumar Today at 12:18 PM		
	Add project files. 1af1aa30 P Pulkit Kumar Today at 11:57 AM		
	Add .gitattributes and .gitignore. 5f0b3877 P Pulkit Kumar Today at 11:57 AM		