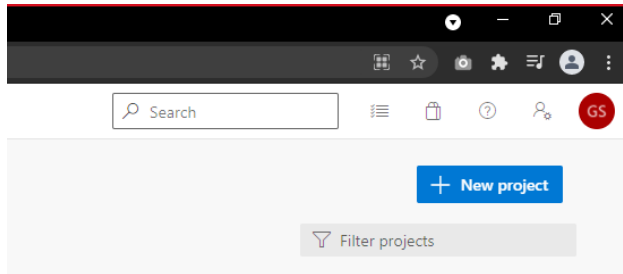


Create Azure Project

Go to the organization page: For example → <https://dev.azure.com/ApteanSandbox/>

Click on **New project**



Enter values and select options as shown in the image below:

A screenshot of the 'Create new project' dialog box in Azure DevOps. The dialog has a title bar with a close button. It contains several fields and options: 'Project name' with the value 'AUFLiteDemo' and a green checkmark; 'Description' with an empty text area; 'Visibility' with two radio buttons, 'Public' (disabled) and 'Private' (selected and highlighted with a blue border); a note about public projects being disabled; an 'Advanced' section with two dropdown menus: 'Version control' set to 'Git' and 'Work item process' set to 'Agile'; and 'Cancel' and 'Create' buttons at the bottom right.

Create new project

Project name *

AUFLiteDemo ✓

Description

Visibility

☐ Public ⓘ
Anyone on the internet can view the project. Certain features like TFVC are not supported.

☒ Private ⓘ
Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

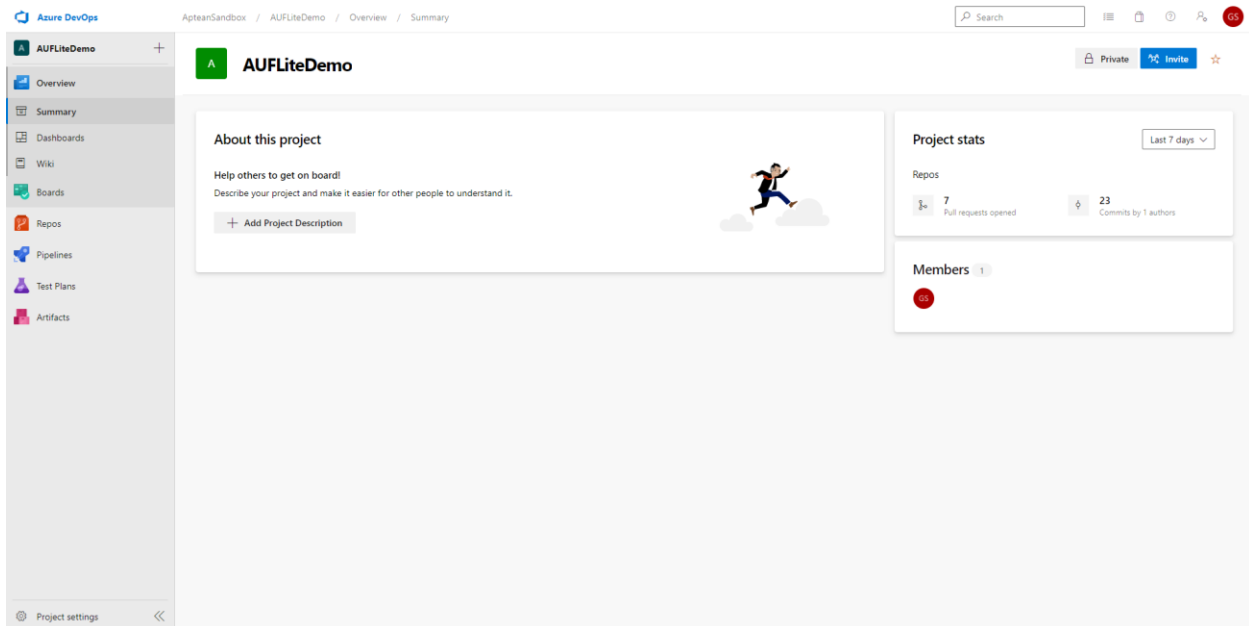
Advanced

Version control ⓘ
Git

Work item process ⓘ
Agile

Cancel Create

After clicking on **Create** button, new project will be created successfully.



Create Git repo and initial commit:

Prerequisite: Git should be installed on your local machine

1.) Open git bash in local machine



2.) Execute these two commands to add your username and email to git global configuration

Git config --global user.name <<Username>>

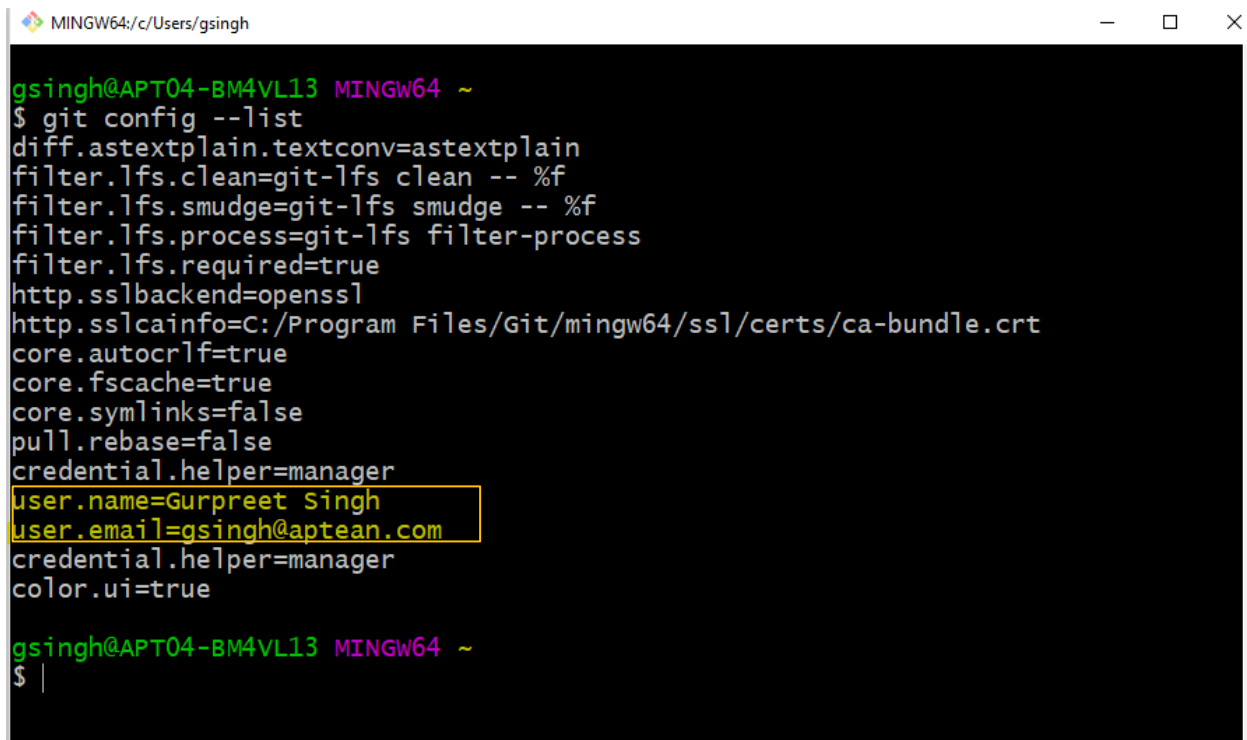
Example: Git config --global user.name Gurpreet Singh

Git config --global user.email <<email id>>

Example: Git config --global user.email gsingh@aptean.com

3.) Check if username and email added successfully by executing this command

Git config --list

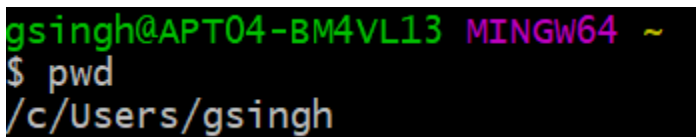


```
MINGW64: c:/Users/gsingh
gsingh@APT04-BM4VL13 MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
user.name=Gurpreet Singh
user.email=gsingh@aptean.com
credential.helper=manager
color.ui=true

gsingh@APT04-BM4VL13 MINGW64 ~
$ |
```

4.) Check working directory:

Command → pwd



```
gsingh@APT04-BM4VL13 MINGW64 ~
$ pwd
/c/Users/gsingh
```

5.) Change directory where you want to create git repo

Command → cd "location"

```

gsingh@APT04-BM4VL13 MINGW64 ~
$ cd "C:\Automation\GitTraining"

gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining
$ |

```

6.) Create folder with the repo name and go in this folder:

Command → mkdir "repo name" and cd "location"

```

gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining
$ mkdir AUFLiteDemo

```

```

gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining
$ cd AUFLiteDemo

gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo
$ |

```

7.) Create git repo at this location:

Command → git init

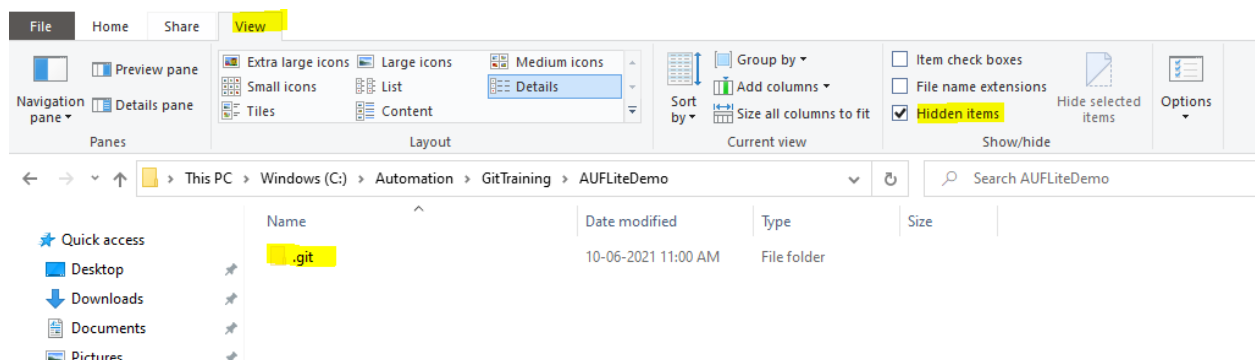
```

gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo
$ git init
Initialized empty Git repository in C:/Automation/GitTraining/AUFLiteDemo/.git/

```

8.) After executing this command, go to your repo → view → check hidden items checkbox as shown in the image below.

After performing these steps, you will see .git folder. That means git repo has been created successfully in the local machine.



9.) Now copy and paste all AUF Lite code inside this git repo.

Windows (C:) > Automation > GitTraining > AUFLiteDemo

10.) Go to git bash and execute these commands:

Command → git status

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .classpath
        .gitignore
        .project
        .settings/
        AllBackUpTestSpecs.json
        CRMTestSpecs.json
        DashboardTemplate/
        Environments.json
        README.md
        TestData/
        TestSpecs.json
        TestSpecs_DemoERP.json
        pom.xml
        src/
        system.properties
        testng.xml

nothing added to commit but untracked files present (use "git add" to track)
```

Command → git add .

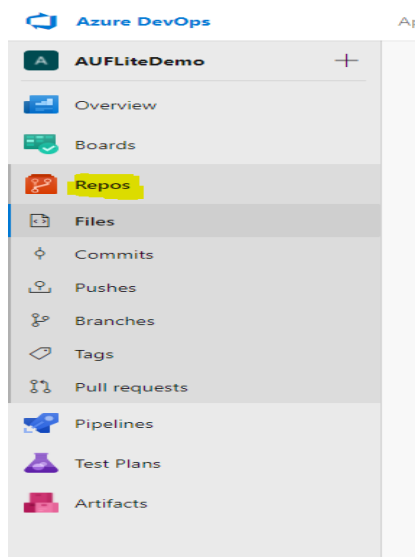
```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo (master)
$ git add .
```

Command → `git commit -m "Initial Commit"`

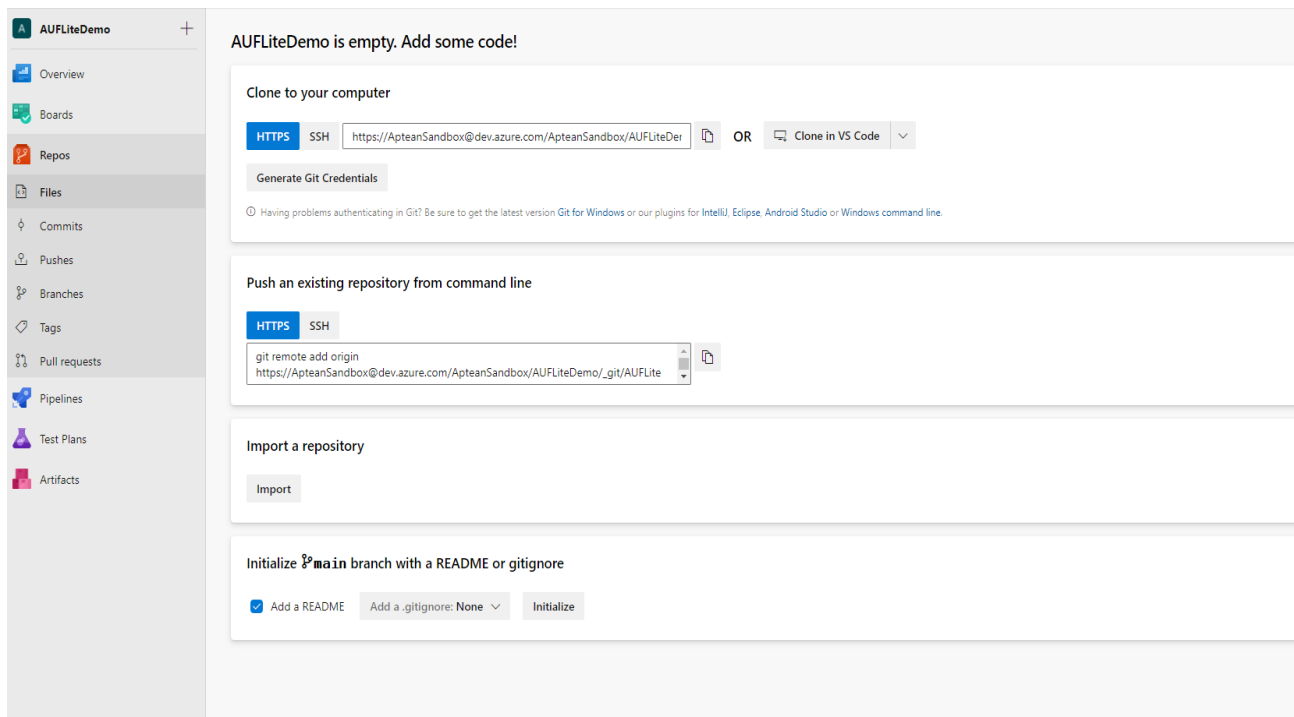
```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo (master)
$ git commit -m "Initial Commit"
[master (root-commit) 7aee090] Initial Commit
196 files changed, 32423 insertions(+)
create mode 100644 .classpath
create mode 100644 .gitignore
create mode 100644 .project
create mode 100644 .settings/org.eclipse.core.resources.prefs
create mode 100644 .settings/org.eclipse.jdt.core.prefs
create mode 100644 .settings/org.eclipse.m2e.core.prefs
create mode 100644 AllBackUpTestSpecs.json
create mode 100644 CRMTTestSpecs.json
create mode 100644 DashboardTemplate/MacroTemplate-V1.xlsx
create mode 100644 Environments.json
create mode 100644 README.md
create mode 100644 TestData/ApiPro - Gina.xlsx
create mode 100644 TestData/ApiPro - Julia.xlsx
create mode 100644 TestData/ApiPro - Murugan.xlsx
create mode 100644 TestData/ApiPro.xlsx
create mode 100644 TestData/Apprise.xlsx
create mode 100644 TestData/ApteanPay.xlsx
create mode 100644 TestData/ApteanPayAssessment.xlsx
create mode 100644 TestData/CRM.xlsx
create mode 100644 TestData/Covid19.xlsx
create mode 100644 TestData/DemoERP_Inventory.xlsx
create mode 100644 TestData/DemoERP_Inventory_Driver.xlsx
```

11.) Push this local repo to azure git server:

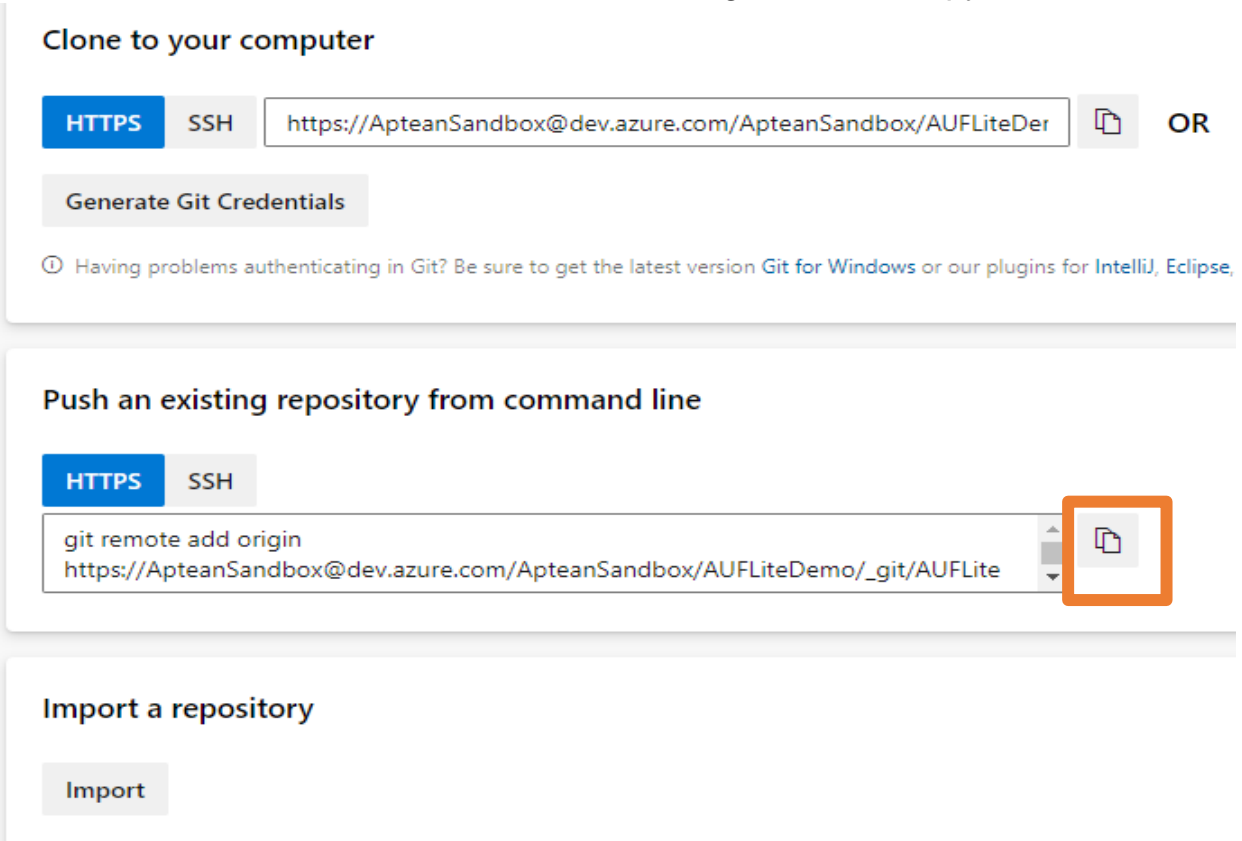
Go to azure project and click on repo in the left panel.



12.) After clicking on repo, this page will open.



Click on the outlined button as shown in the image below to copy the command



13.) Go to git bash, paste this command and execute

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo (master)
$ git remote add origin https://ApteanSandbox@dev.azure.com/ApteanSandbox/AUFLiteDemo/_git/AUFLiteDemo

gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo (master)
$ git push -u origin --all
Enumerating objects: 296, done.
Counting objects: 100% (296/296), done.
Delta compression using up to 8 threads
Compressing objects: 100% (253/253), done.
Writing objects: 100% (296/296), 750.95 KiB | 8.34 MiB/s, done.
Total 296 (delta 86), reused 0 (delta 0), pack-reused 0
remote: Analyzing objects... (296/296) (732 ms)
remote: Storing packfile... done (94 ms)
remote: Storing index... done (59 ms)
To https://dev.azure.com/ApteanSandbox/AUFLiteDemo/_git/AUFLiteDemo
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemo (master)
$ |
```

Go to azure project, refresh the page and you will see the code inside the master branch.

The screenshot shows the Azure DevOps web interface for a project named 'AUFLiteDemo'. The left sidebar has a 'Repos' section with 'Files' selected. The main area displays a file explorer view of the 'master' branch. The file explorer shows a list of files and folders, including .settings, DashboardTemplate, src, TestData, .classpath, .gitignore, .project, AllBackUpTestSpecs.json, CRMTTestSpecs.json, Environments.json, pom.xml, MI README.md, system.properties, testng.xml, TestSpecs_DemoERP.json, and TestSpecs.json. Each file entry shows its last change time (9m ago) and the commit hash (7aee0905) with the commit message 'Initial Commit Gurpreet Singh'.

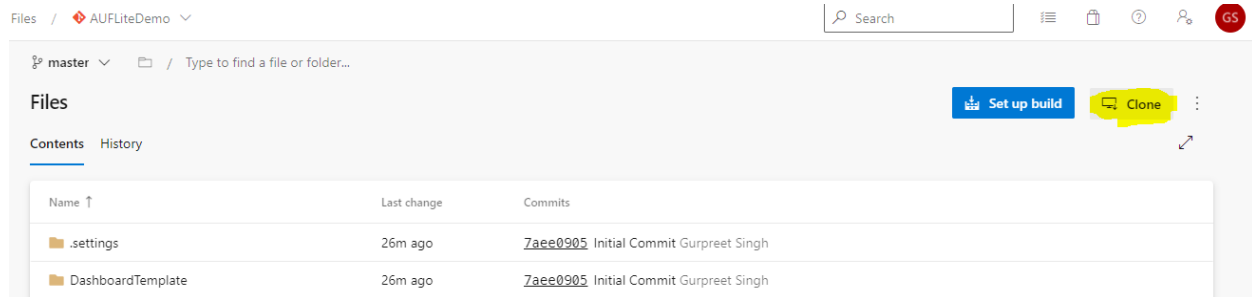
| Name | Last change | Commits |
|-------------------------|-------------|--|
| .settings | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| DashboardTemplate | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| src | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| TestData | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| .classpath | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| .gitignore | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| .project | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| AllBackUpTestSpecs.json | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| CRMTTestSpecs.json | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| Environments.json | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| </> pom.xml | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| MI README.md | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| system.properties | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| </> testng.xml | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| TestSpecs_DemoERP.json | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |
| TestSpecs.json | 9m ago | 7aee0905 Initial Commit Gurpreet Singh |

Git flow

1.) Clone the repo in local machine.

Go to the repo you want to clone.

Click on the clone button.



Clone Repository

Command line

HTTPS

SSH

https://ApteanSandbox@dev.azure.com/ApteanSandbox/AUFLiteDemo/_git/AUFLiteDemo

Generate Git Credentials

IDE



Clone in VS Code



Having problems authenticating in Git? Be sure to get the latest version
① [Git for Windows](#) or our plugins for [IntelliJ](#), [Eclipse](#), [Android Studio](#) or [Windows command line](#).

2.) Copy the URL, open git bash in the location where you want to clone the code and execute this command.

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone
$ git clone https://ApteanSandbox@dev.azure.com/ApteanSandbox/AUFLiteDemo/_git/AUFLiteDemo
Cloning into 'AUFLiteDemo'...
remote: Azure Repos
remote: Found 296 objects to send. (27 ms)
Receiving objects: 100% (296/296), 750.75 KiB | 1.04 MiB/s, done.
Resolving deltas: 100% (87/87), done.
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone
$ |
```

3.) After executing this command, local git repo will be created at the specified location.

(C:) > Automation > GitTraining > AUFLiteDemoClone > AUFLiteDemo

Search AUFLiteDemo

| Name | Date modified | Type | Size |
|--------------------|---------------------|-----------------|------|
| .git | 10-06-2021 11:44 AM | File folder | |
| .settings | 10-06-2021 11:44 AM | File folder | |
| DashboardTemplate | 10-06-2021 11:44 AM | File folder | |
| src | 10-06-2021 11:44 AM | File folder | |
| TestData | 10-06-2021 11:44 AM | File folder | |
| .classpath | 10-06-2021 11:44 AM | CLASSPATH File | 2 KB |
| .gitignore | 10-06-2021 11:44 AM | Text Document | 1 KB |
| .project | 10-06-2021 11:44 AM | PROJECT File | 1 KB |
| AllBackUpTestSpecs | 10-06-2021 11:44 AM | JSON File | 5 KB |
| CRMTestSpecs | 10-06-2021 11:44 AM | JSON File | 1 KB |
| Environments | 10-06-2021 11:44 AM | JSON File | 2 KB |
| pom | 10-06-2021 11:44 AM | XML Document | 8 KB |
| README | 10-06-2021 11:44 AM | MD File | 1 KB |
| system | 10-06-2021 11:44 AM | PROPERTIES File | 1 KB |
| testng | 10-06-2021 11:44 AM | XML Document | 1 KB |
| TestSpecs | 10-06-2021 11:44 AM | JSON File | 1 KB |
| TestSpecs_DemoERP | 10-06-2021 11:44 AM | JSON File | 1 KB |

4.) Create Branch.

Command → git branch <<branch name>>

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone/AUFLiteDemo (master)
$ git branch feature
```

5.) Checkout to specific branch.

Command → git checkout <<branch name>>

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone/AUFLiteDemo (master)
$ git checkout feature
Switched to branch 'feature'
```

6.) Check the status of your local repo.

Command → git status

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone/AUFLiteDemo (feature)
$ git status
On branch feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   TestSpecs.json

no changes added to commit (use "git add" and/or "git commit -a")
```

7.) Stage changes for commit:

Command → git add .

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone/AUFLiteDemo (feature)
$ git add .
```

8.) Commit changes:

Command → git commit -m "commit message"

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone/AUFLiteDemo (feature)
$ git commit -m "feature created"
[feature c3dde86] feature created
1 file changed, 4 insertions(+), 4 deletions(-)
```

9.) Get log of the current branch:

Command → git log

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone/AUFLiteDemo (feature)
$ git log
commit c3dde86f3726d3f18495f16e2ef941fb72761df7 (HEAD -> feature)
Author: Gurpreet Singh <gsingh@aptean.com>
Date: Thu Jun 10 11:52:50 2021 +0530

    feature created

commit 7aee0905e61b259002161e6401f03c9a33c2f638 (origin/master, origin/HEAD, master)
Author: Gurpreet Singh <gsingh@aptean.com>
Date: Thu Jun 10 11:14:52 2021 +0530

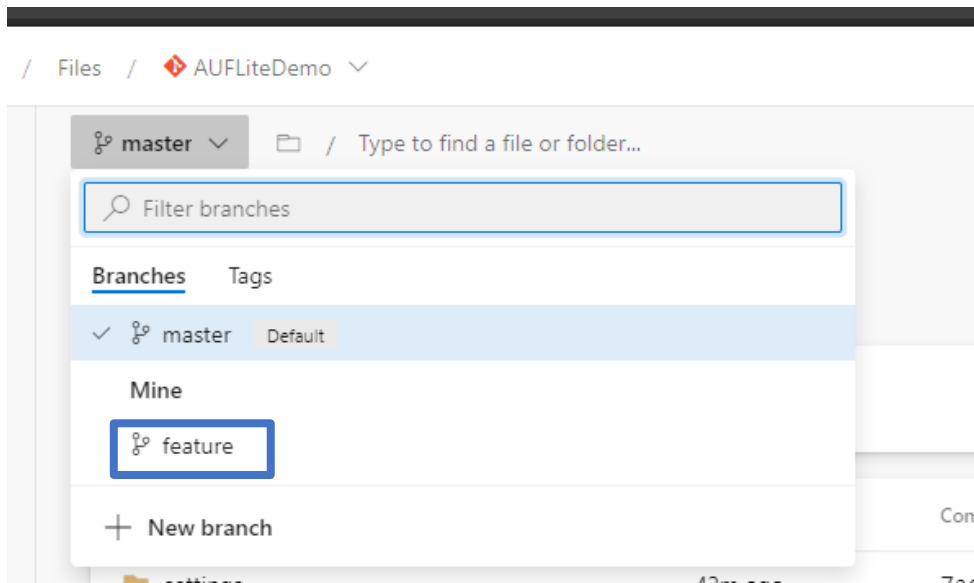
    Initial Commit
```

10.) Push current branch and all the changes to azure git server:

Command → git push -u origin <<branch name>>

```
gsingh@APT04-BM4VL13 MINGW64 /c/Automation/GitTraining/AUFLiteDemoClone/AUFLiteDemo (feature)
$ git push -u origin feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Analyzing objects... (3/3) (87 ms)
remote: Storing packfile... done (57 ms)
remote: Storing index... done (50 ms)
To https://dev.azure.com/ApteanSandbox/AUFLiteDemo/_git/AUFLiteDemo
 * [new branch] feature -> feature
Branch 'feature' set up to track remote branch 'feature' from 'origin'.
```

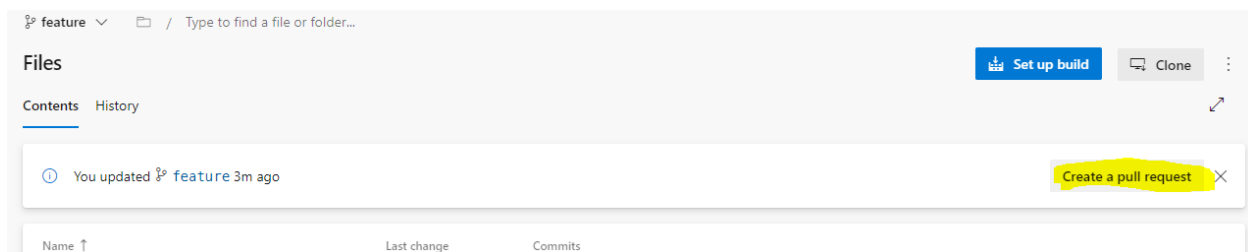
11.) Go to azure project and check if the branch has been created or not.



12.) Merge feature branch in master branch

Create a pull request

Click on the button highlighted in the image shown below



Click on **create** button to create pull request.

New pull request

🔗 feature ▾ into 🔗 master ▾ ⇄

Overview Files 1 Commits 1

Title

feature created

Description

feature created

[Markdown supported.](#) Drag & drop, paste, or select files to insert.

[Link work items.](#)



feature created

Reviewers

Add required reviewers

 Search users and groups to add as reviewers

Work items to link

Search work items by ID or title

Tags

Create ▾

Click on approve and complete button to complete the pull request.

feature created

Active

1115

Gurpreet Singh

feature into master

Overview

Files

Updates

Commits

✓

No merge conflicts

Last checked Just now

Description

feature created

Show everything (1)

✱

Add a comment...

✱

Gurpreet Singh created the pull request

Just now

Approve

Complete

Reviewers

Add

Required

No required reviewers

Optional

No optional reviewers

Tags

+

No tags

Work items

+

No work items

Select these options and click on complete merge.

Complete pull request

Merge type

Merge (no fast forward)



Post-completion options

- ☒ Complete associated work items after merging
- ☒ Delete feature after merging
- ☐ Customize merge commit message

Cancel

Complete merge

13.) Retrieve the latest meta-data info from the origin:

Command: `git fetch`

14.) Update the local version of a repository from a remote:

Command: `git pull`

15.) Moving or combining a sequence of commits to a new base commit:

Command: `git rebase <<branch name>>`

If there are some conflicts in the branch, resolve them, and perform below commands to continue changes:

`git status`

`git rebase --continue`