Understanding Azure DevOps and Git

What is Azure DevOps?

Azure DevOps is a cloud-based platform provided by Microsoft to support collaborative software development. It offers various services including version control, build automation, release management, and more.

What is Git?

Git is a distributed version control system used for tracking changes in source code during software development. It allows multiple developers to work on a project simultaneously and efficiently manage code changes.

Creating Azure DevOps Environment

1. **Sign up for Azure DevOps**: Go to the Azure DevOps website and sign up for an account if you haven't already.

2. **Create a Project:** Once logged in, create a new project by clicking on the "New Project" button. Give your project a name and select appropriate settings.

3. **Set up Git Repository:** Within your project, navigate to the "Repos" tab and create a new Git repository. Choose a name for your repository and configure any additional settings as needed.

Configuring Git on Local Machine

1. **Install Git:** If you haven't already, download and install Git on your local machine from the official Git website.

2. **Configure Git:** Open a terminal or command prompt and configure Git with your username and email using the following commands:

   git config --global user.name "Your Name"

   git config --global user.email your.email@example.com

3. **Clone Repository:** Clone the Git repository from Azure DevOps to your local machine using the following command:

   git clone <repository-url>

   Replace <repository-url> with the URL of your Azure DevOps Git repository.

4. **Add Test Files:** Create some test files in your local repository directory. You can do this by simply creating new text files or copying existing files into the directory.

5. **Add Files to Git:** Add the test files to the Git repository using the following command:

   git add .

6. Commit Changes: Commit the added files to the repository with a descriptive message using the following command:

git commit -m "Initial commit"

7. Push Changes: Push the committed changes to the Azure DevOps Git repository using the following command:

   git push origin master

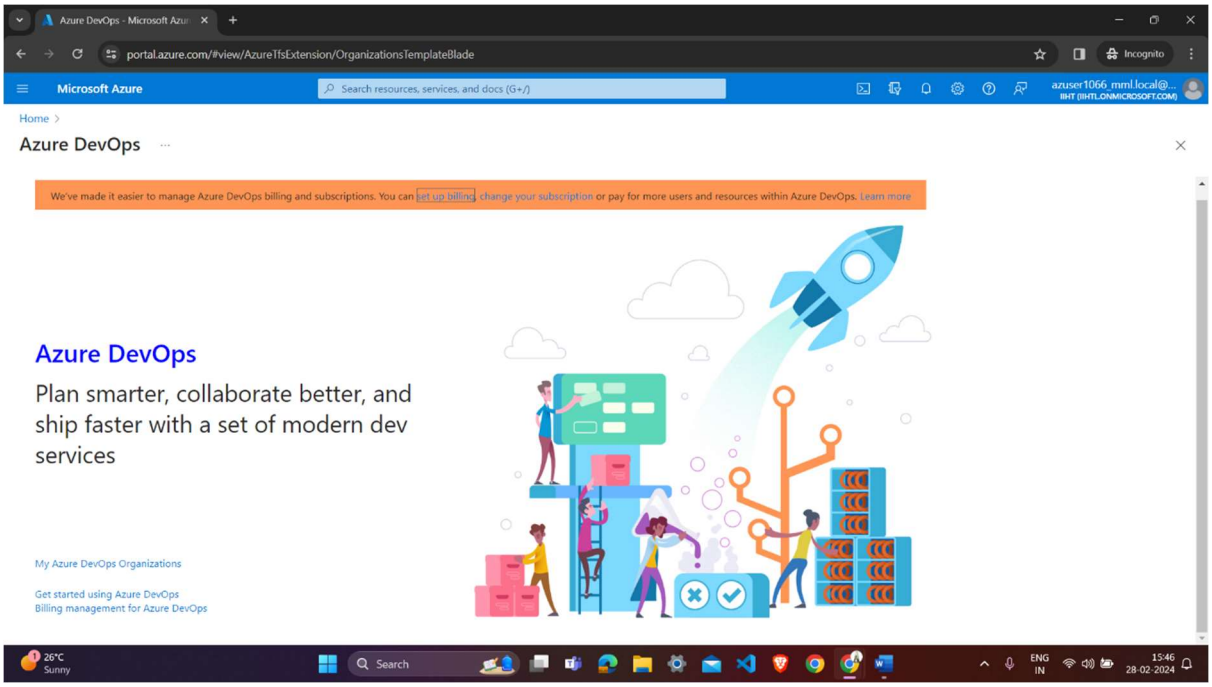
This command pushes the changes to the master branch of the remote repository.

Summary

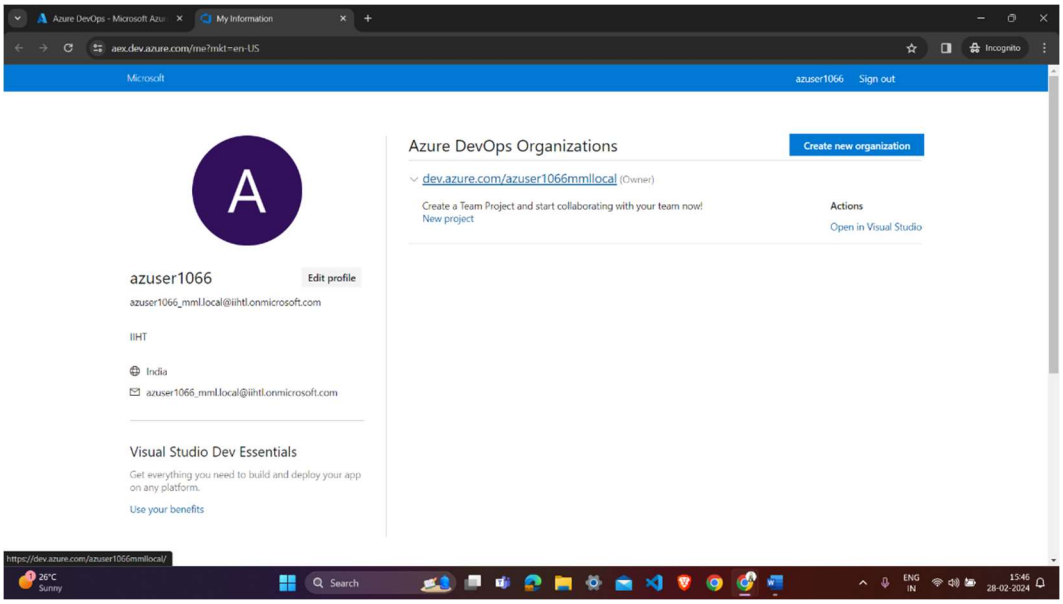
By following these steps, you've created an Azure DevOps environment, configured a Git repository, and set it up on your local machine. You've also added some test files to the repository. Now you're ready to collaborate on your project using Azure DevOps and Git!

Login to azure devops

Click on the my azure Devops organization

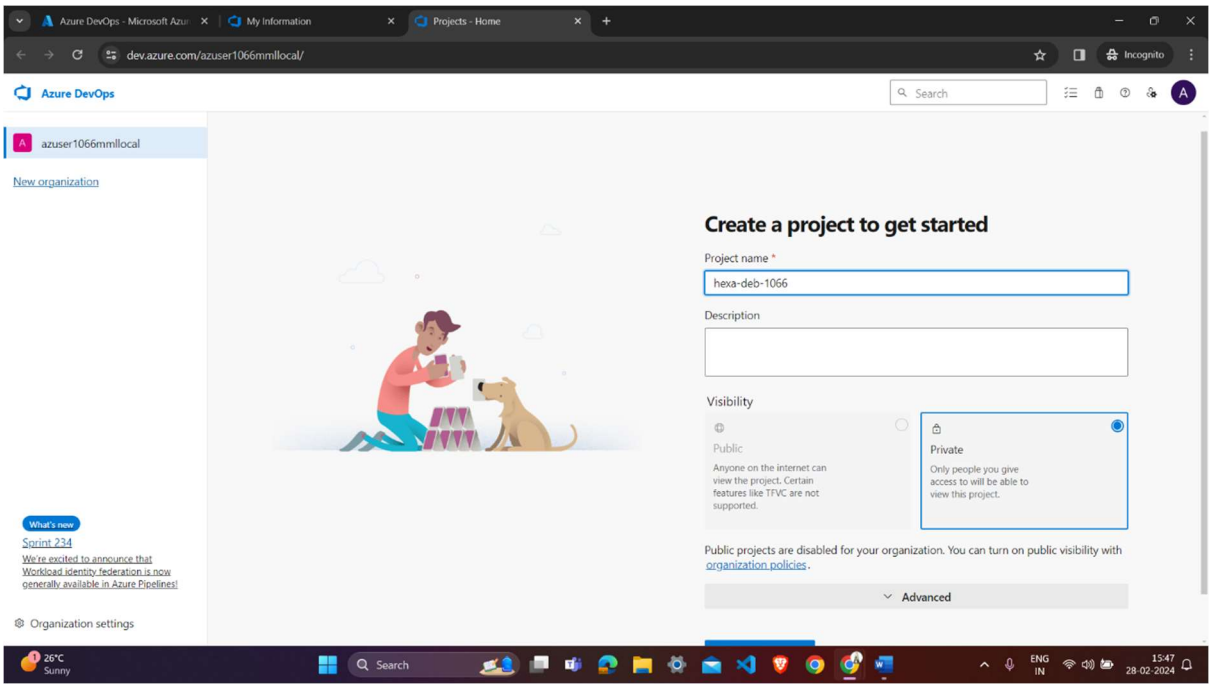


Create a new organization :

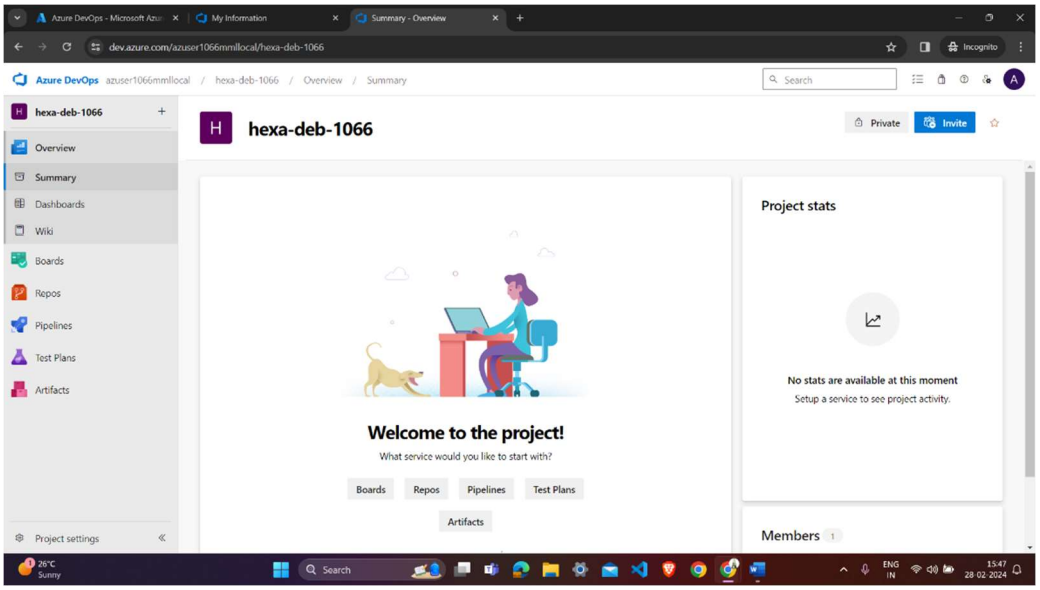


Create a project to get started

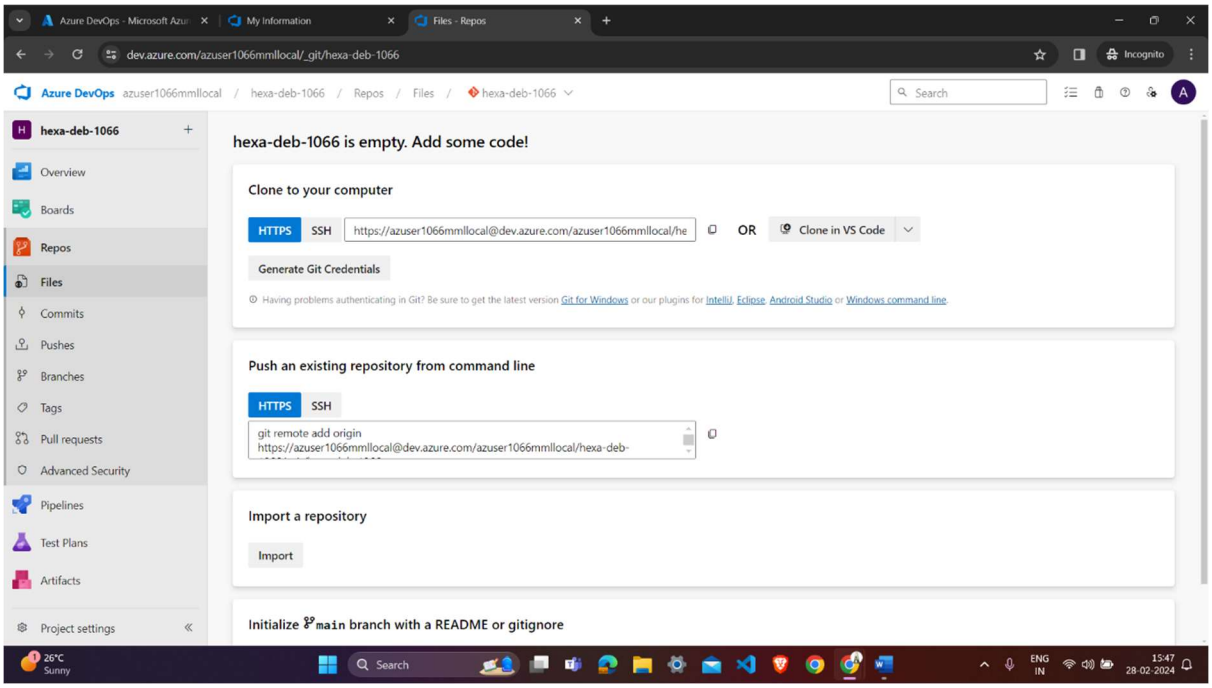
Name of the project is hexa-deb-1066



Project becomes created :-

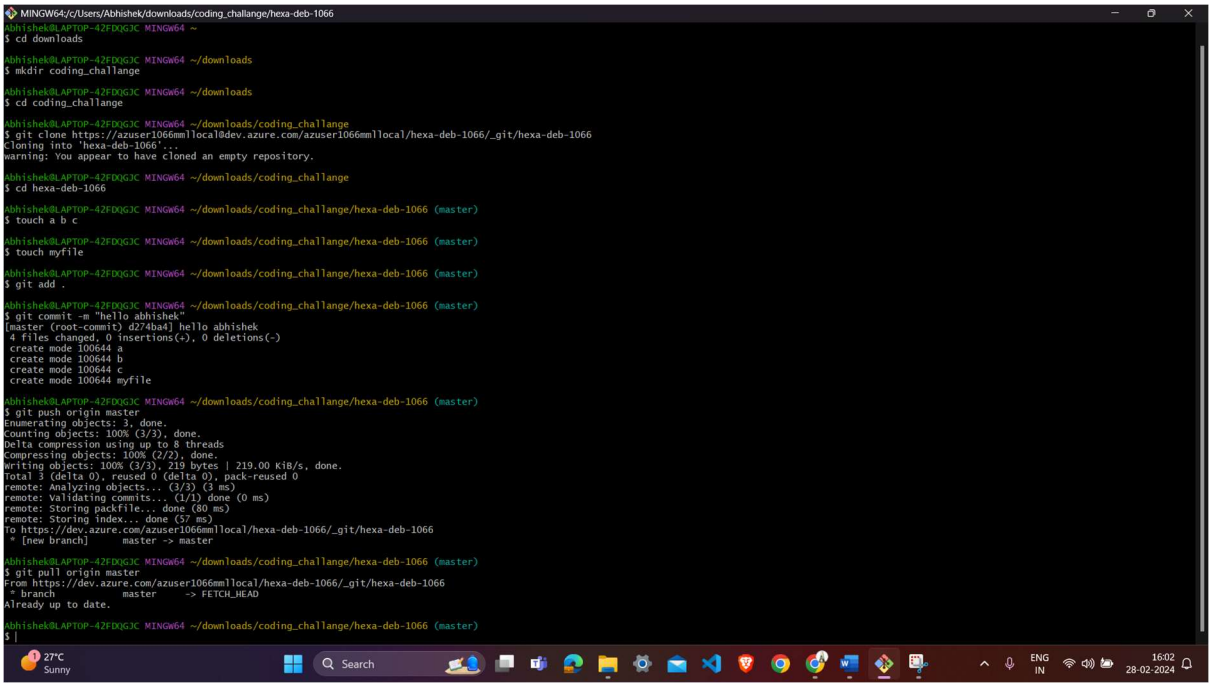


From clone to your computer we get link copy it for cloning it into the local:-

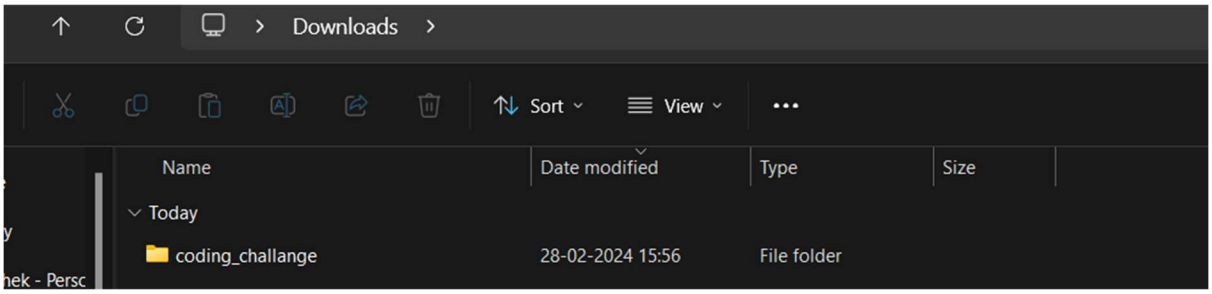


Then perform these command on gitbash

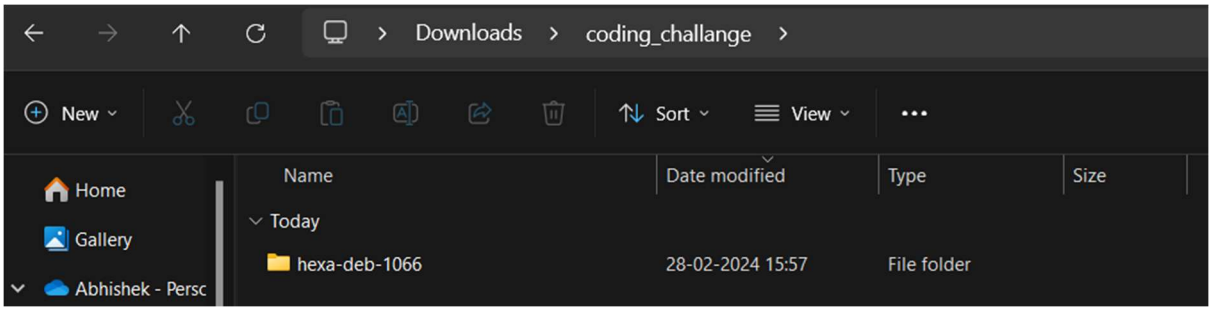
From these command we clone the repository on local and add file from local and then push these file on azure repo (shreen shot attach below)



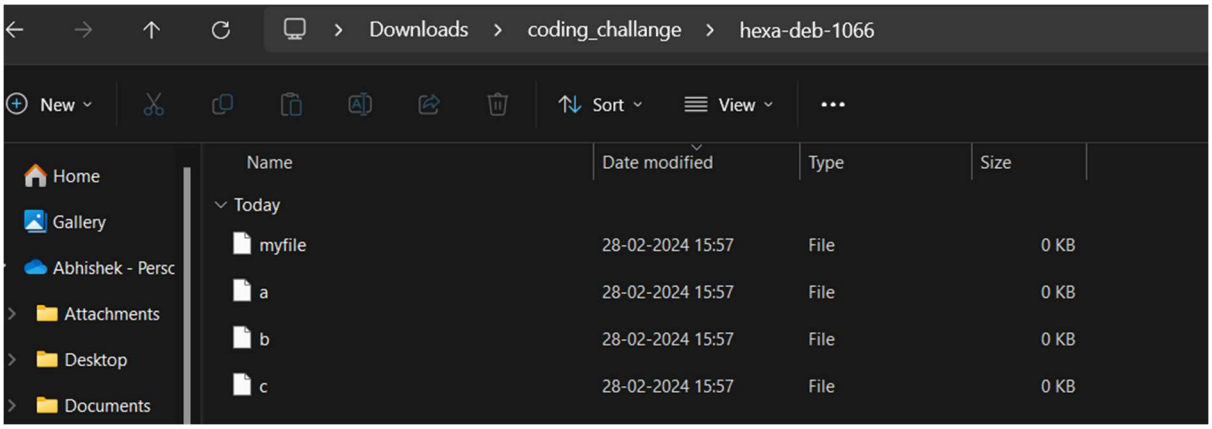
Create a directory in download of local:-



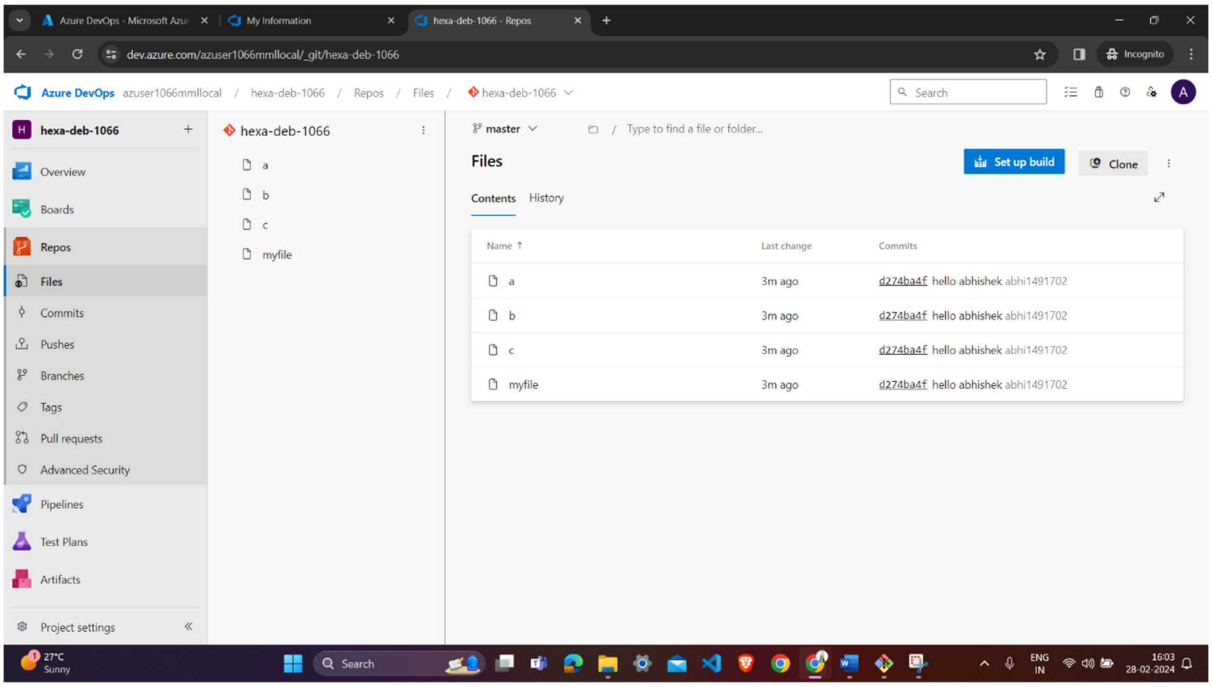
Clone hexa-deb-1066 locally from azure through the above command in the git bash



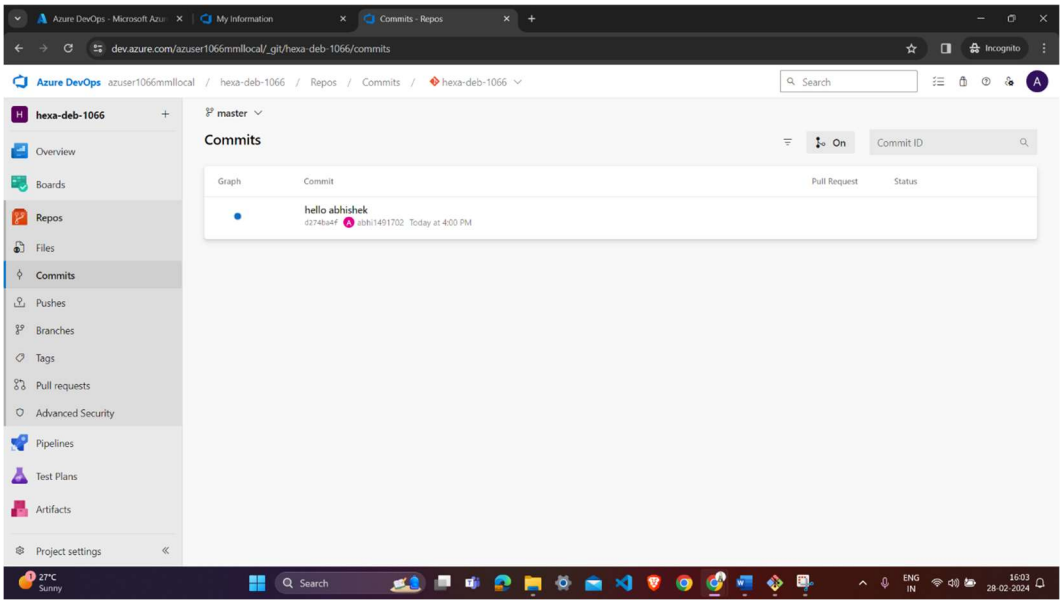
Add files a ,b,c, and myfile local through git bash



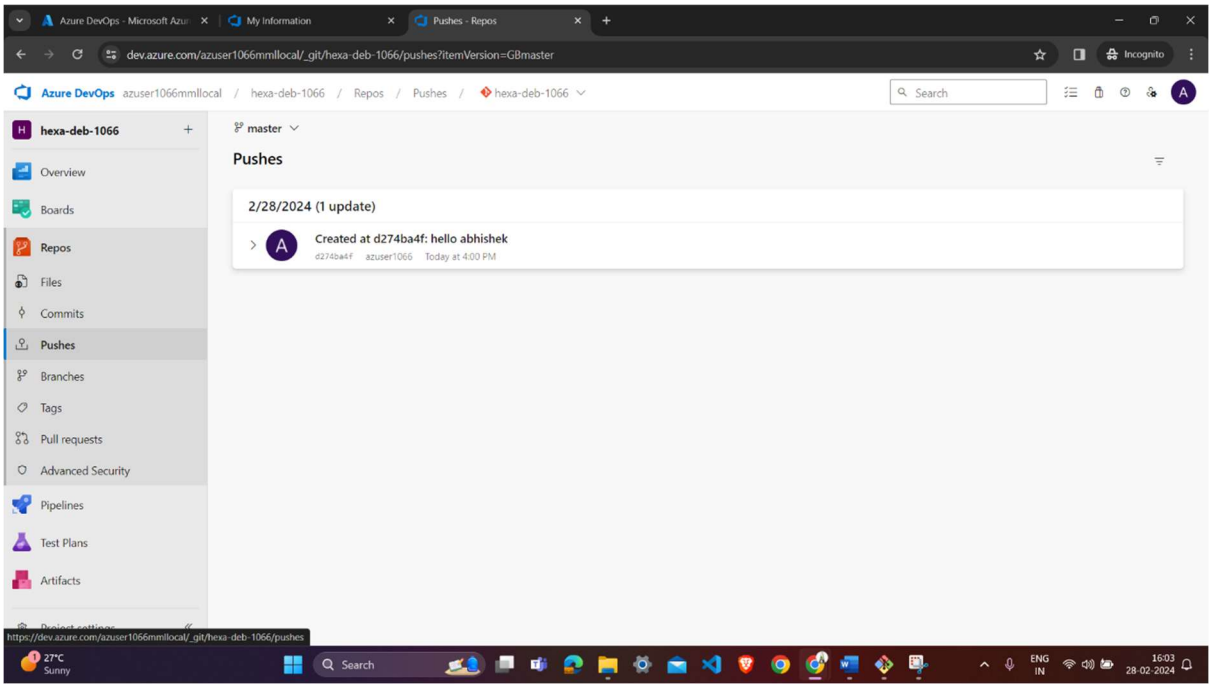
Here we can see that files are added to the Azure devops repository through git bash



Here we can see the commit done :-



Here we can see the pushes done by command on the git bash:-



In history we can also check the commit is done:-