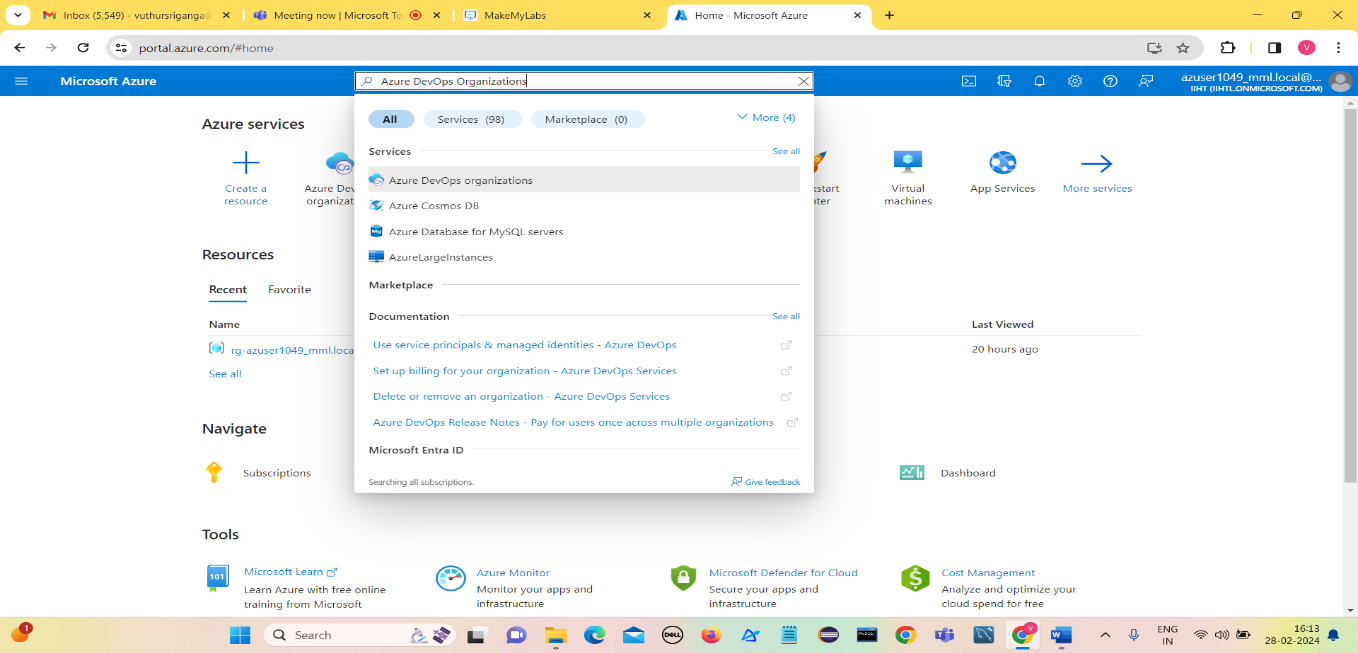
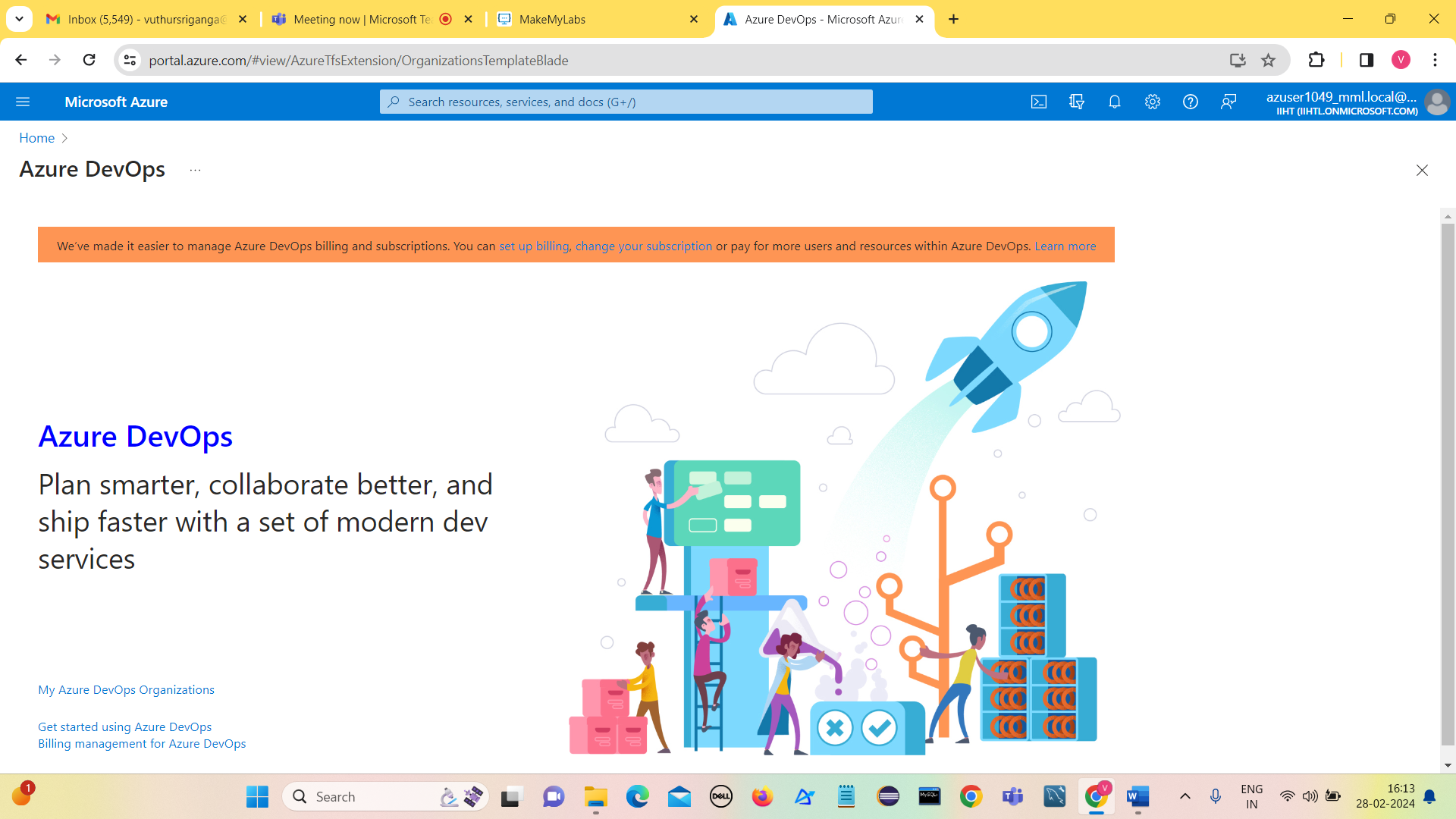
**Azure DevOps and Synapse Coding Assessment**

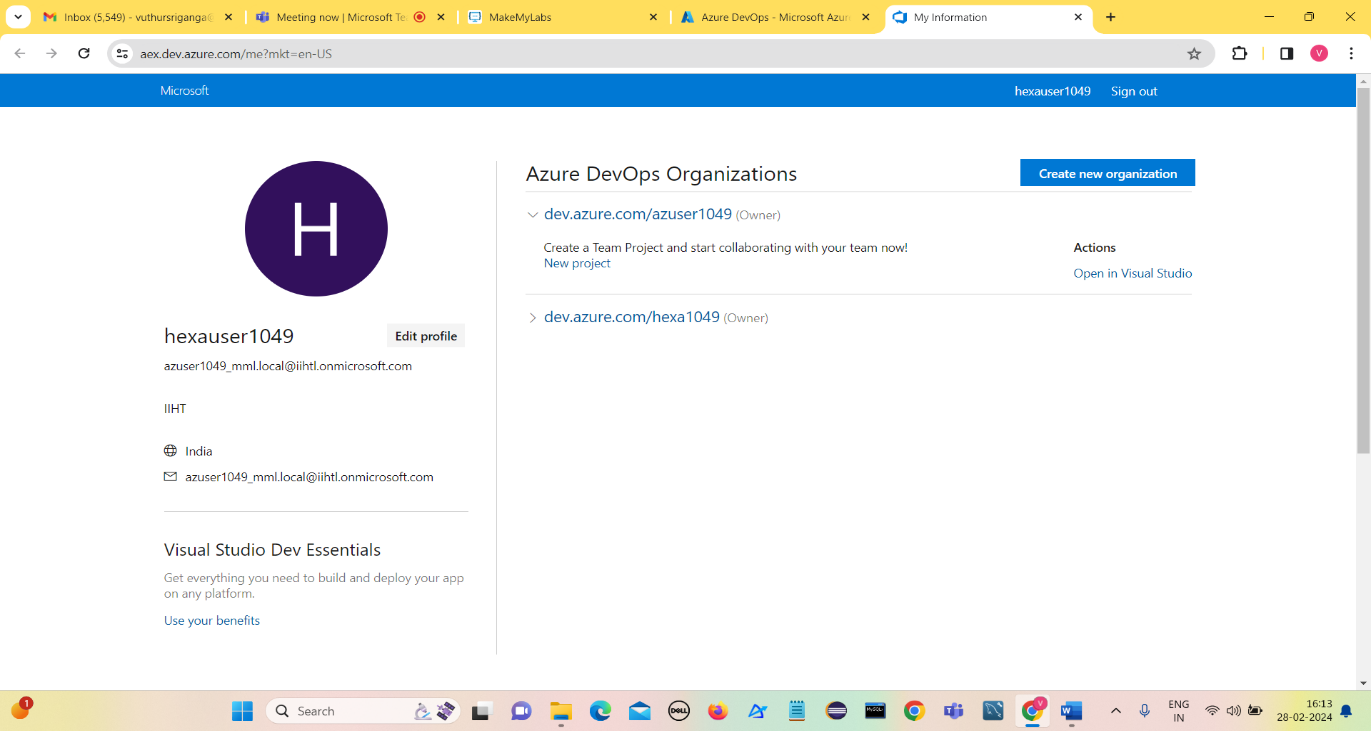
1. **Create Azure Devops Environment and configuring Azure Devops Git Repository ,configure on your local git to implement this upload few test files on same.**

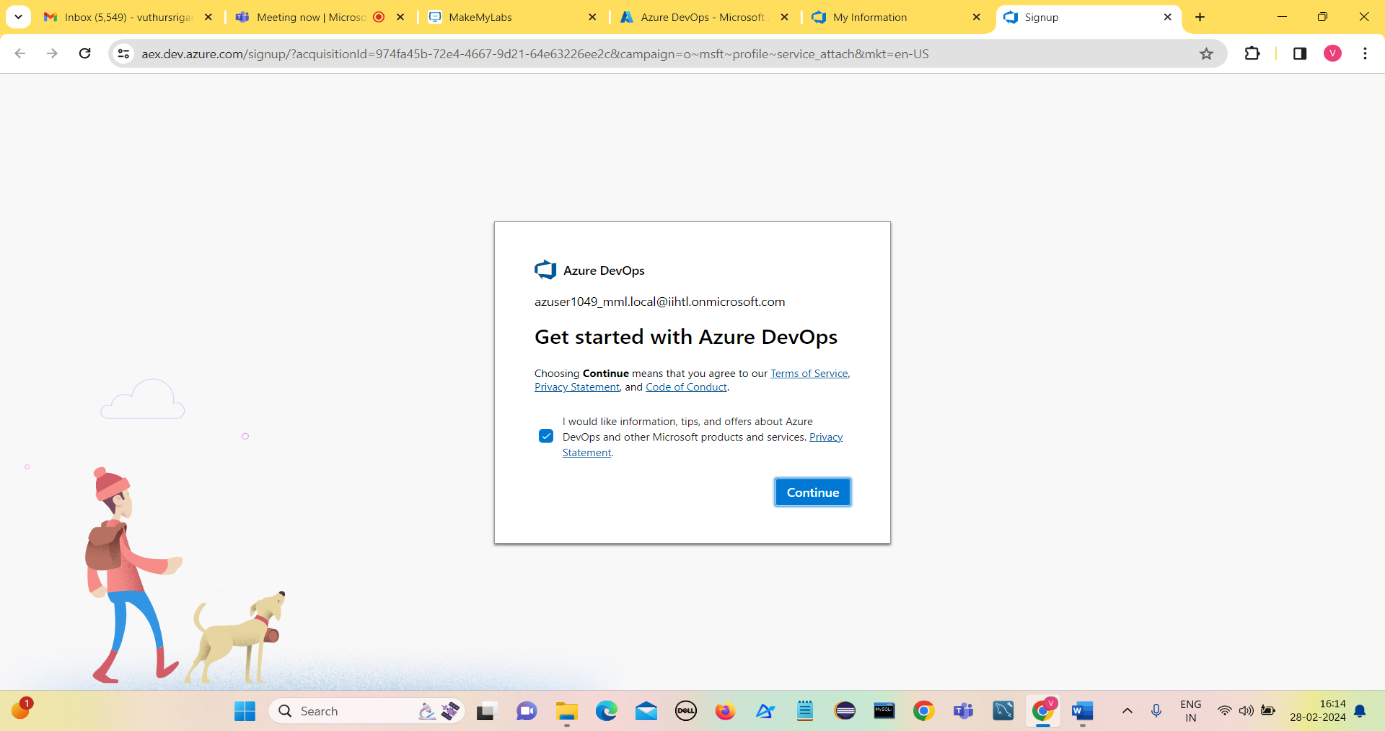
**Steps to be followed to set up an Azure DevOps Organization.**

Open Azure Portal and Search for Azure DevOps Organization.

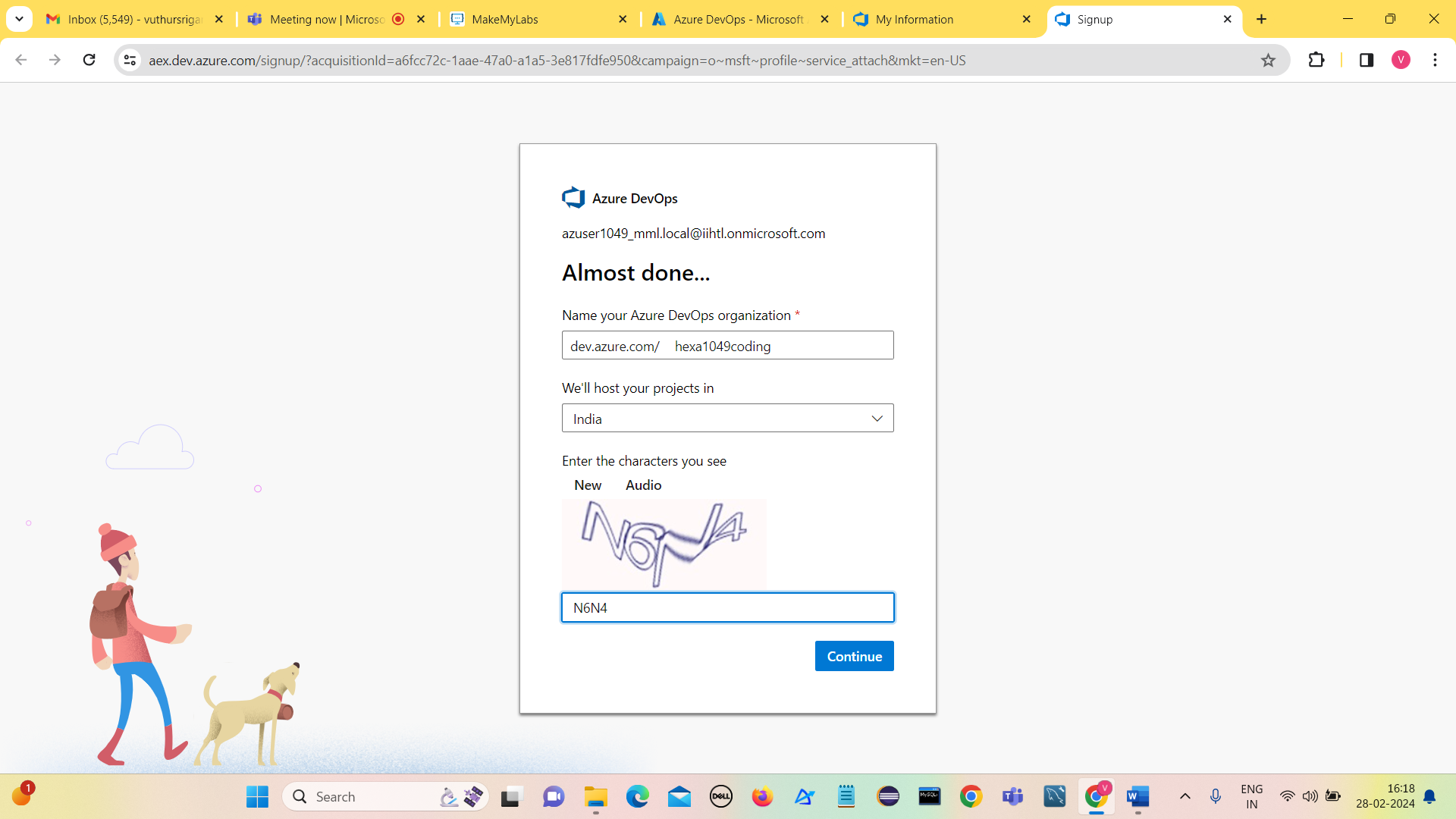


Click on My Azure DevOps Organizations

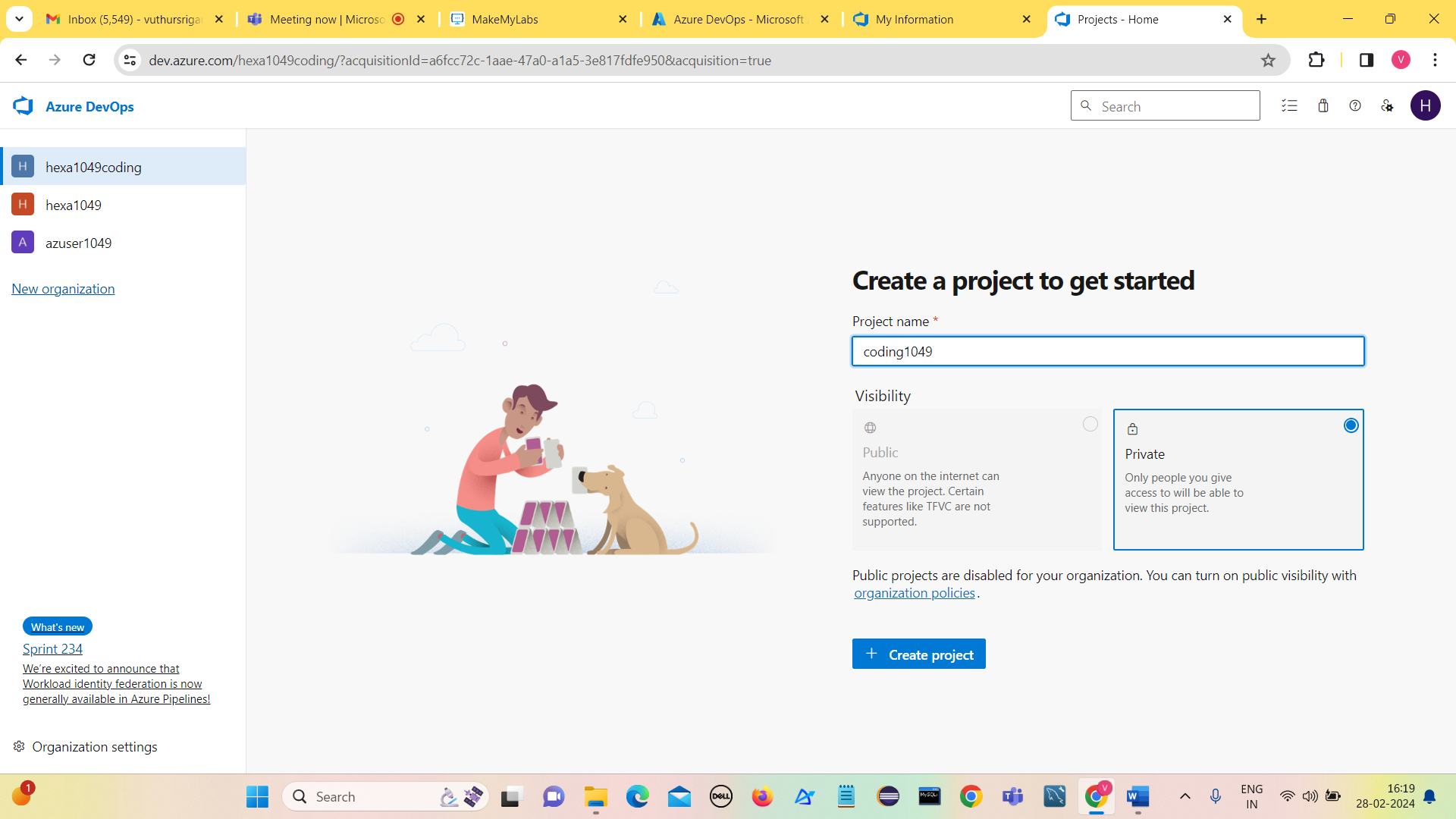


Click on Create new organization.

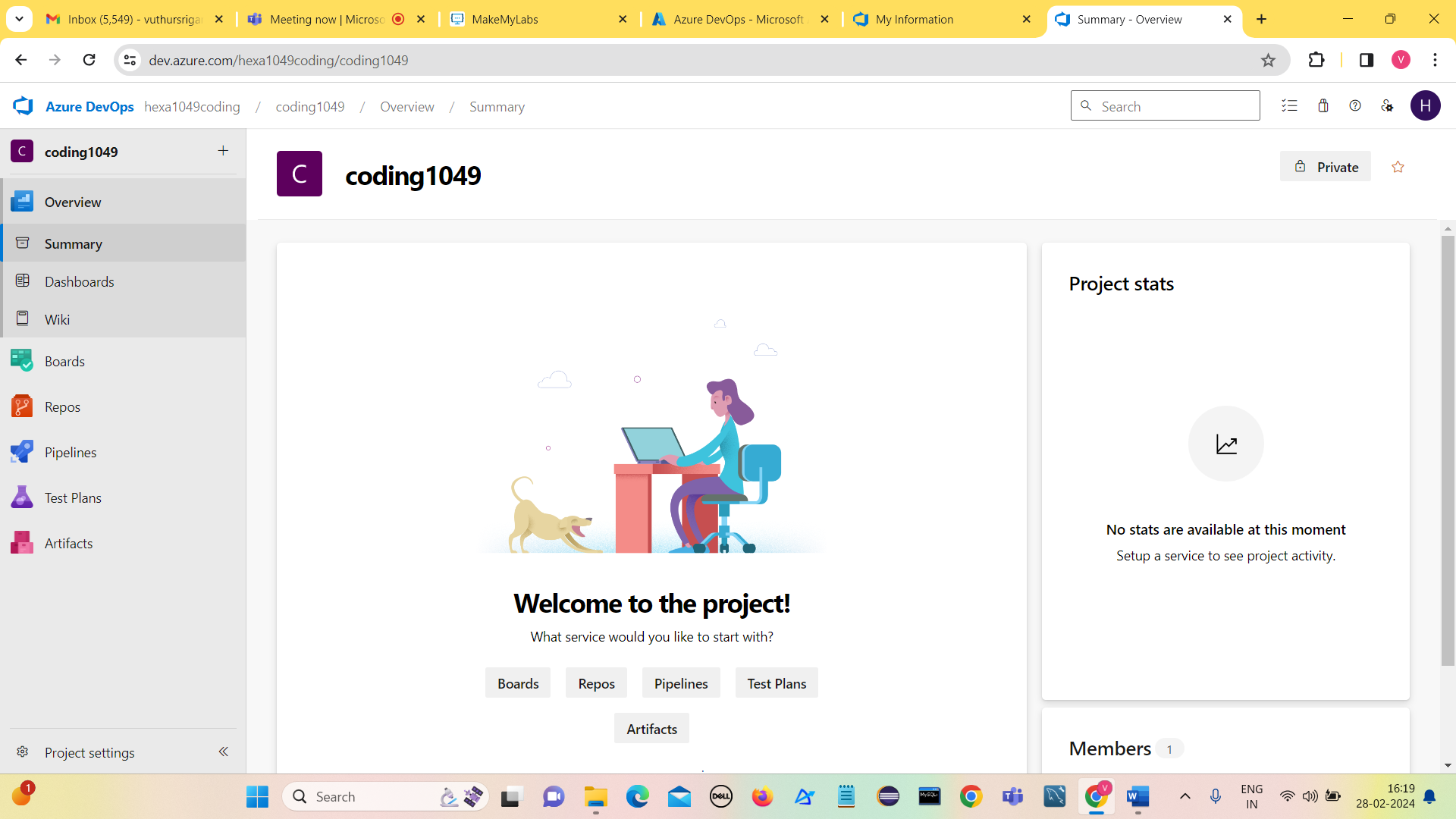
Click on Continue



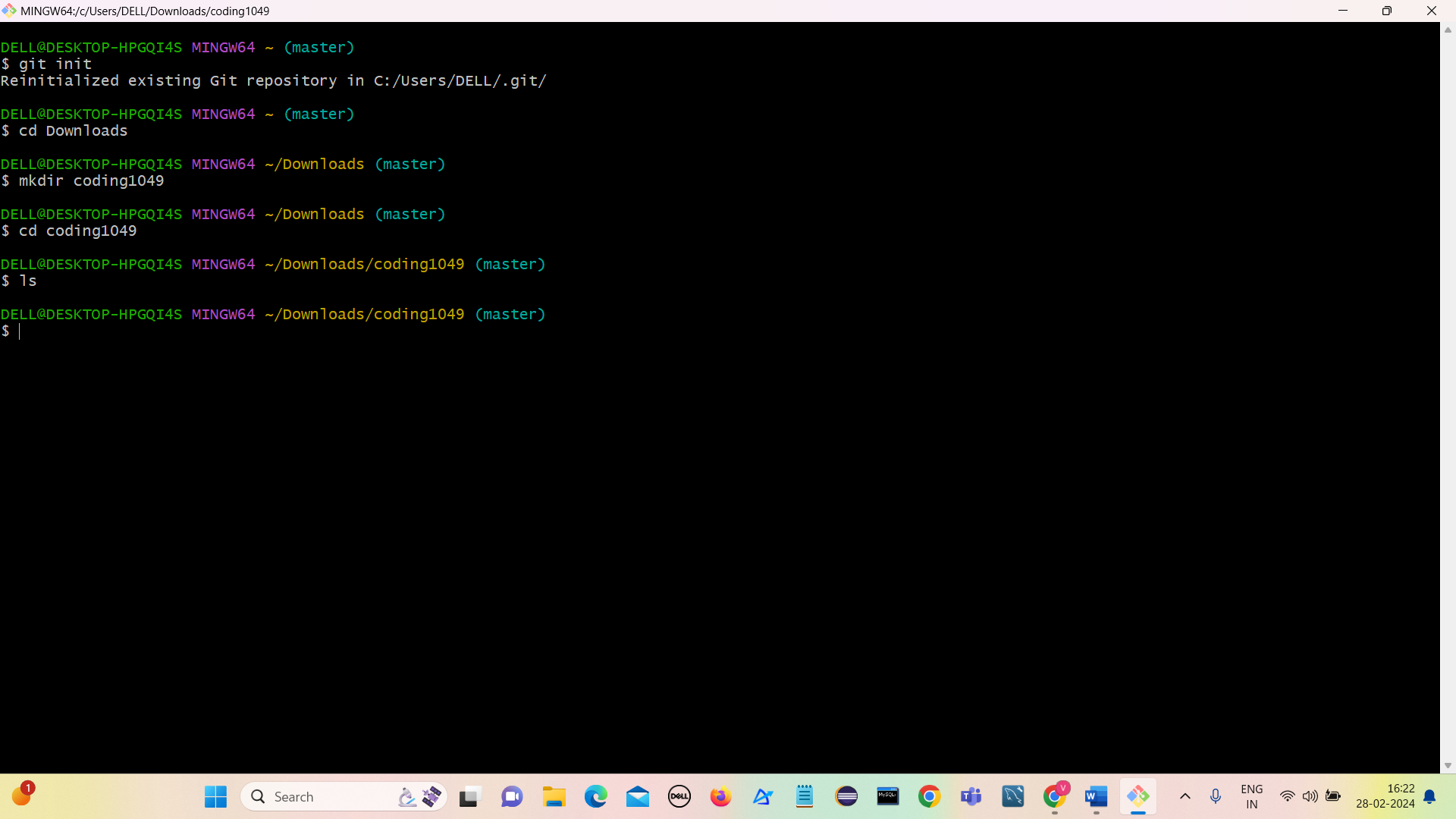
Give the name of Azure DevOps organization and the hosting country , Click on Continue.



Give the name of project and click on Create project.



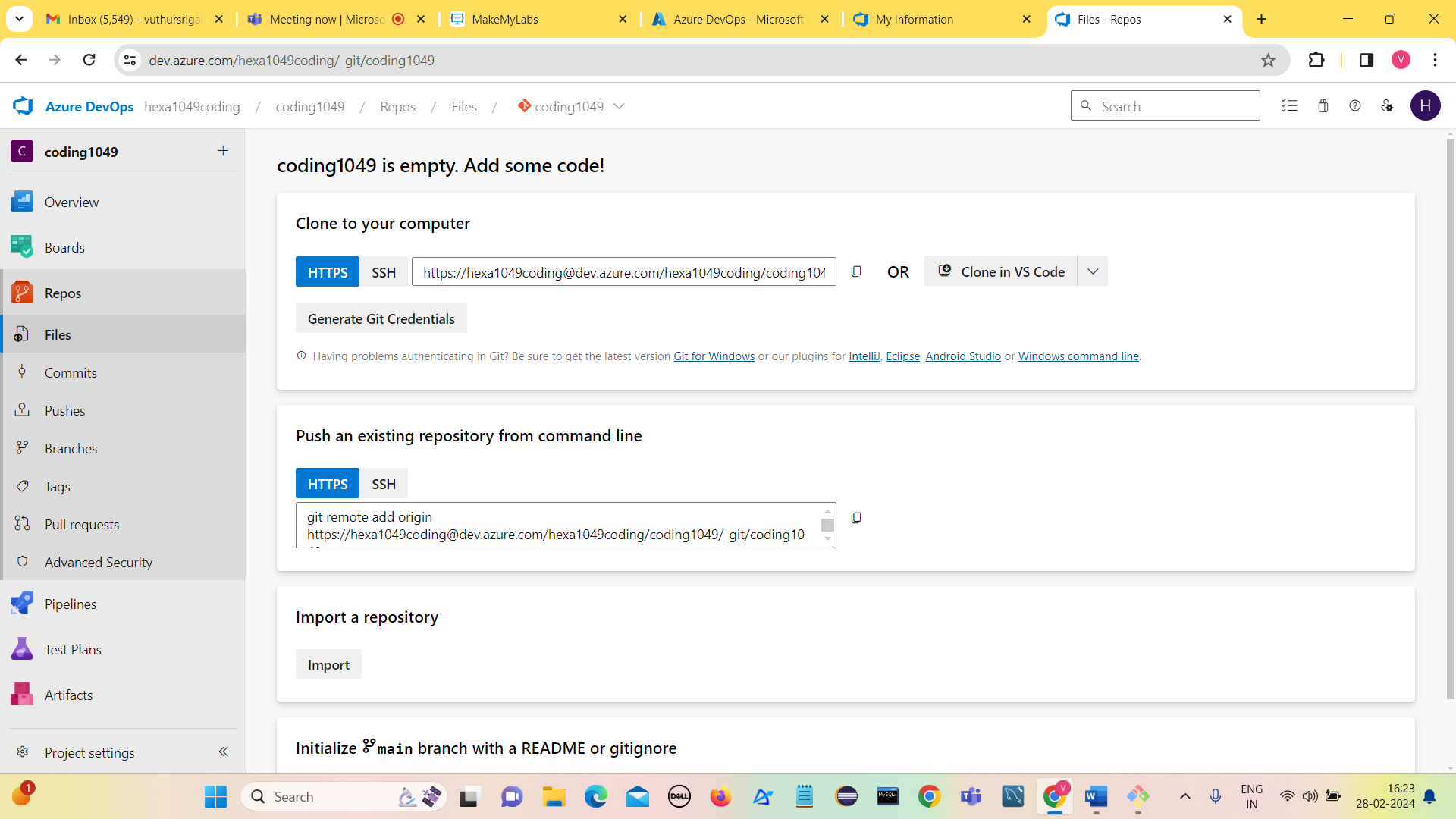
Go to Git Bash terminal and create a folder (coding1049) in Downloads using mkdir command



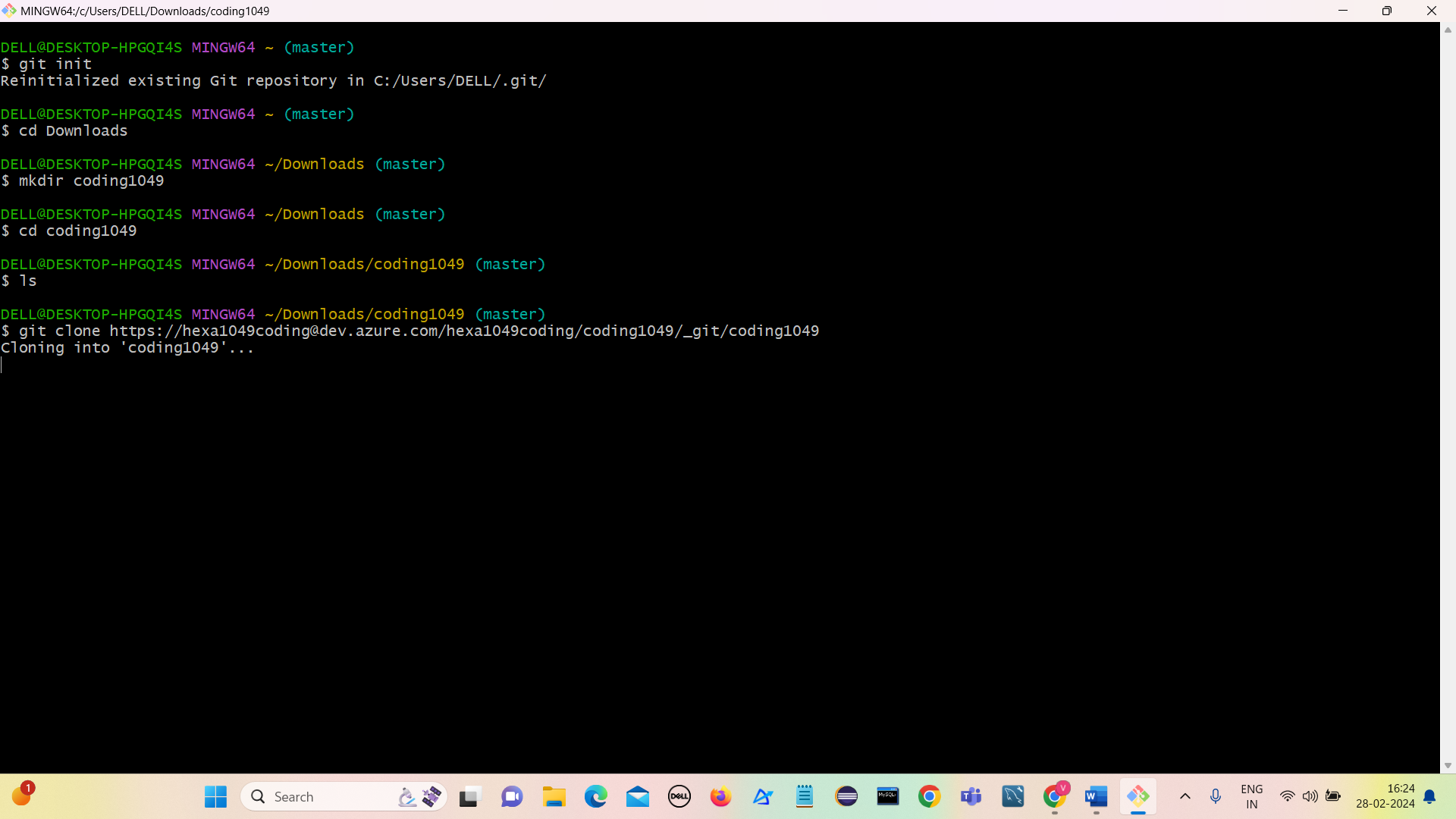
We observe that there are no files in the folder it is empty.

**Cloning the Repository to local:**

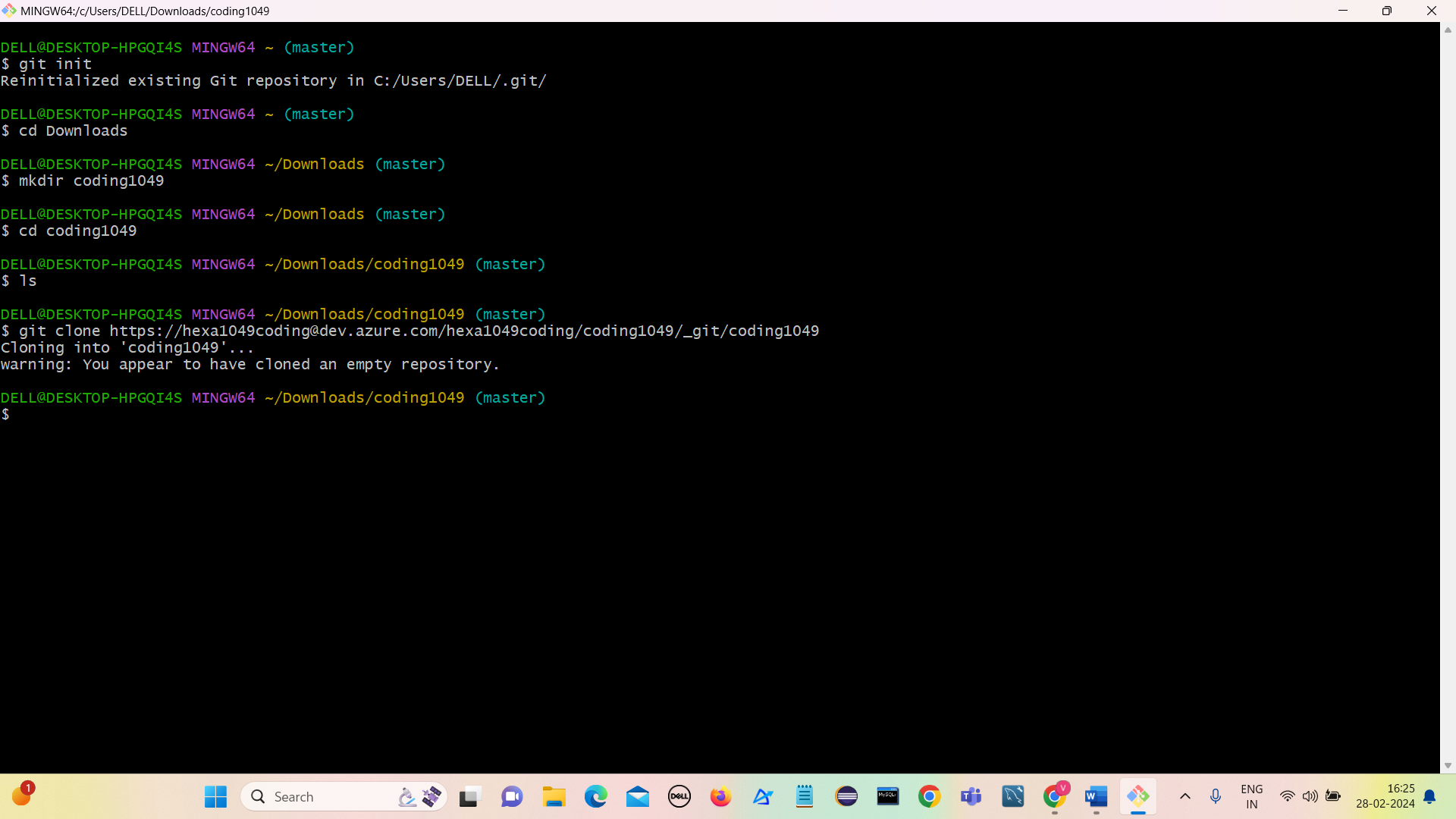
Copy url from repos in the DevOps project.



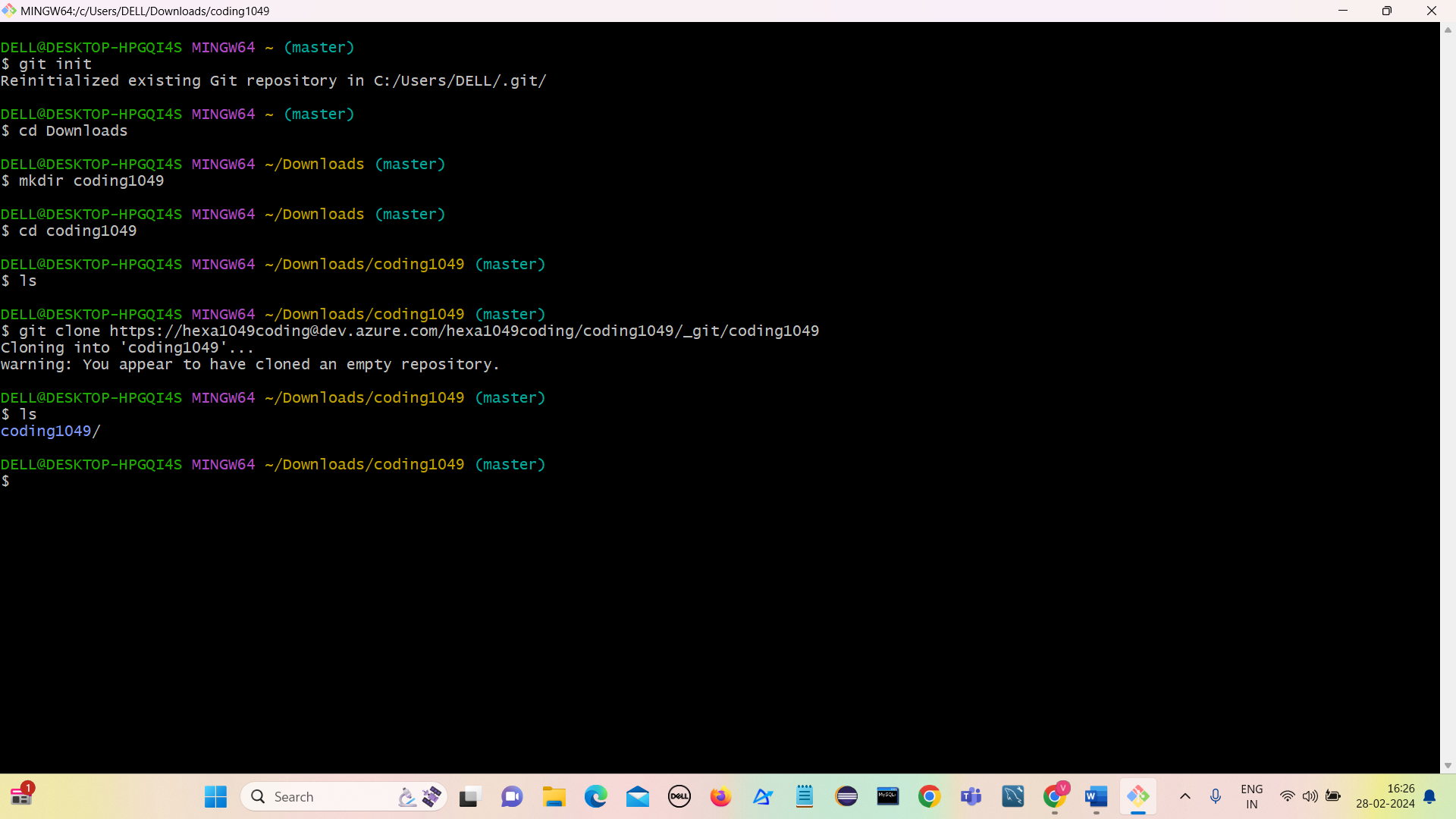
Use command **git clone url**



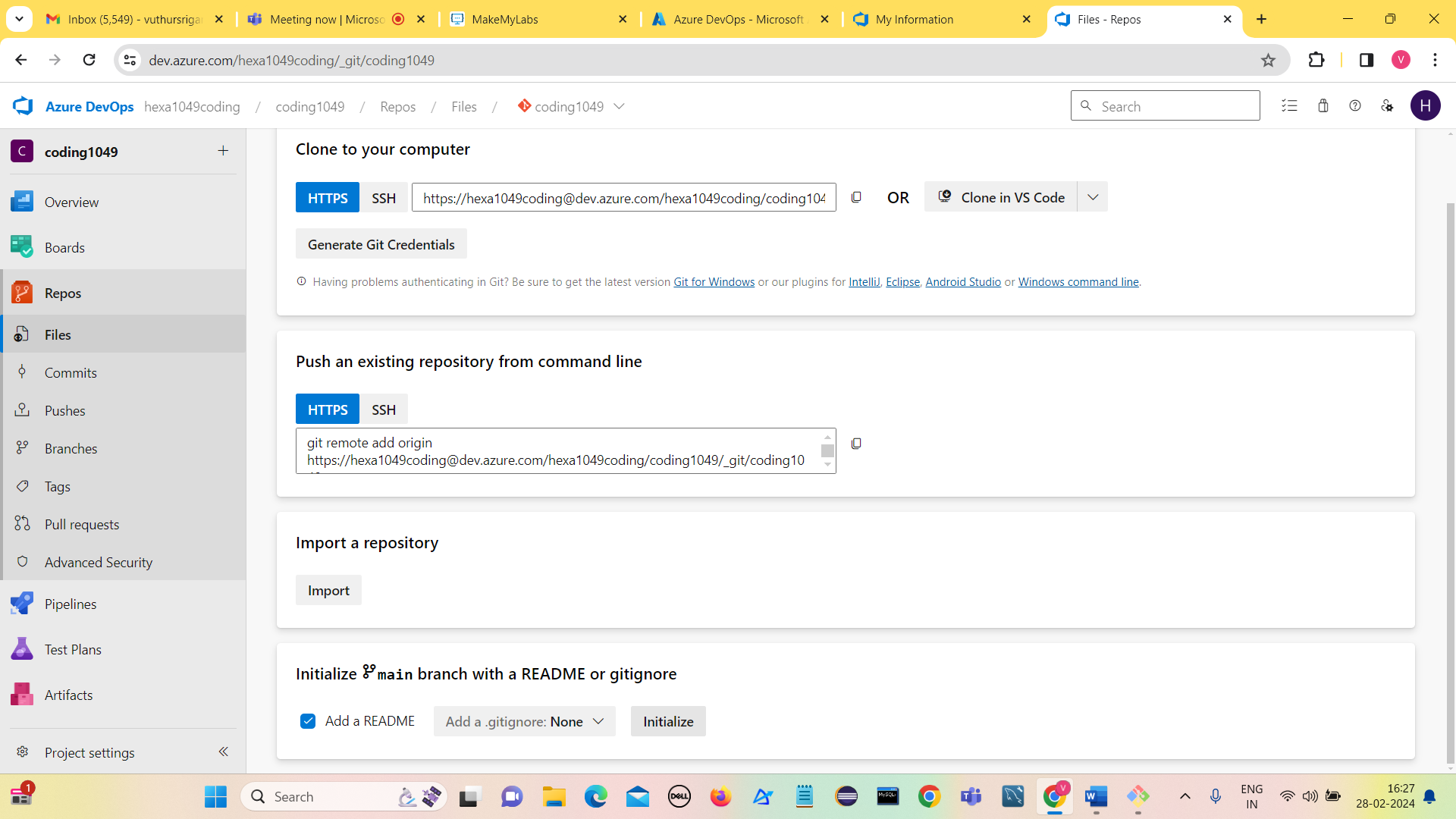
It shows we have cloned and empty repository.

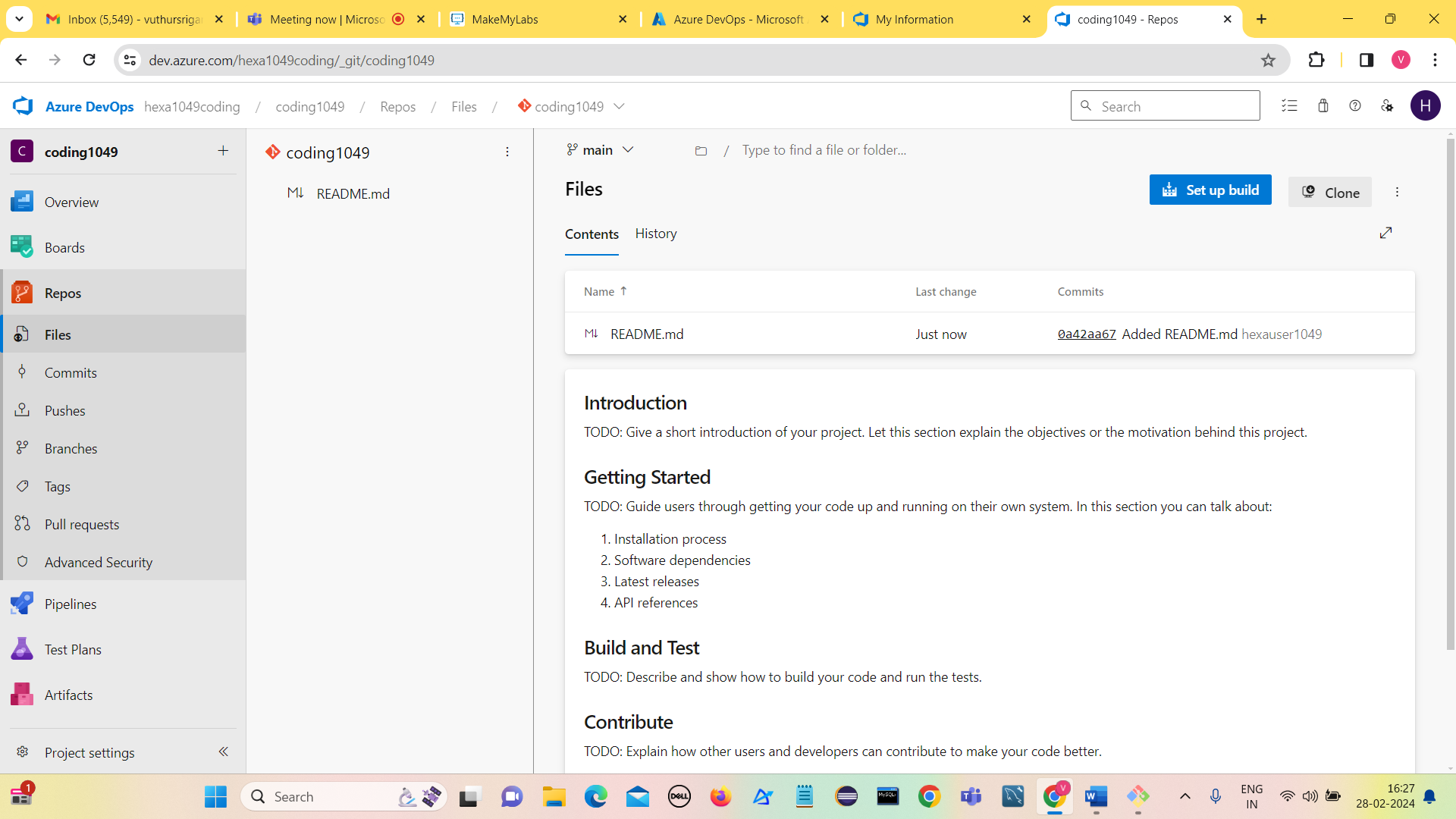


The folder got created in the coding1049 after cloning the repository.



Click on Initialize to add a readME file into the project.

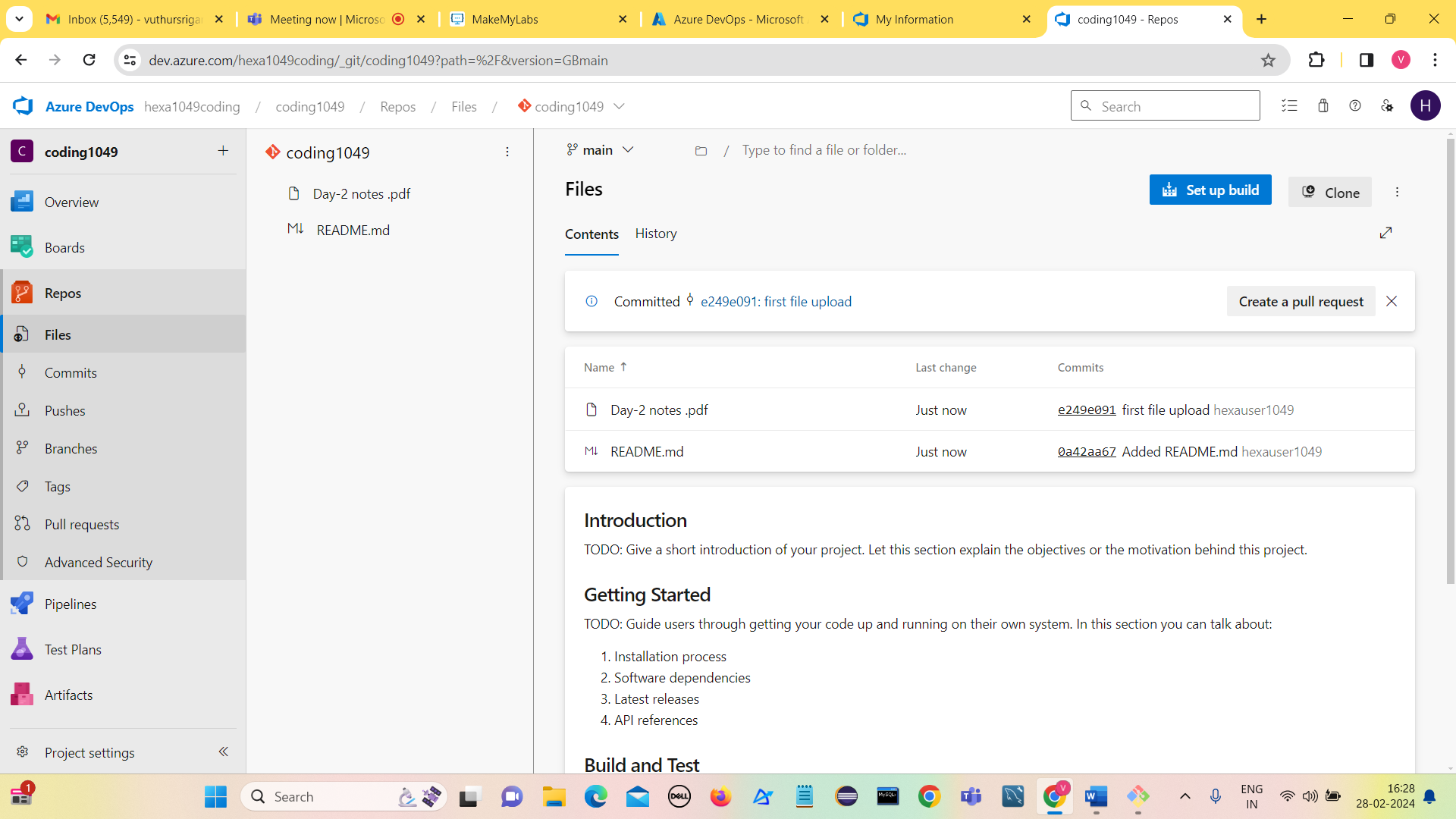




Uploading a file Day-2 notes.pdf into the project.



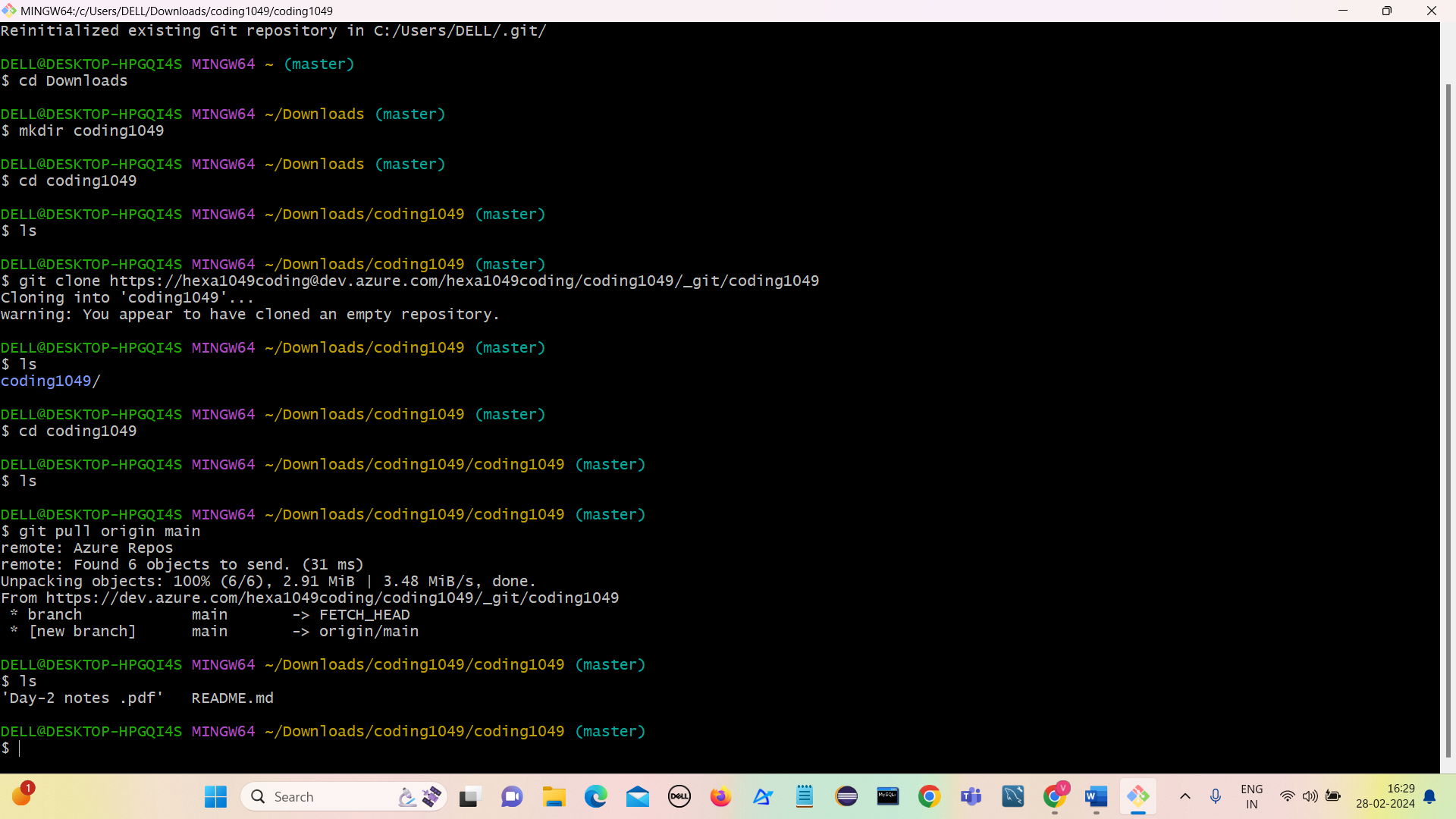
Click on Commit

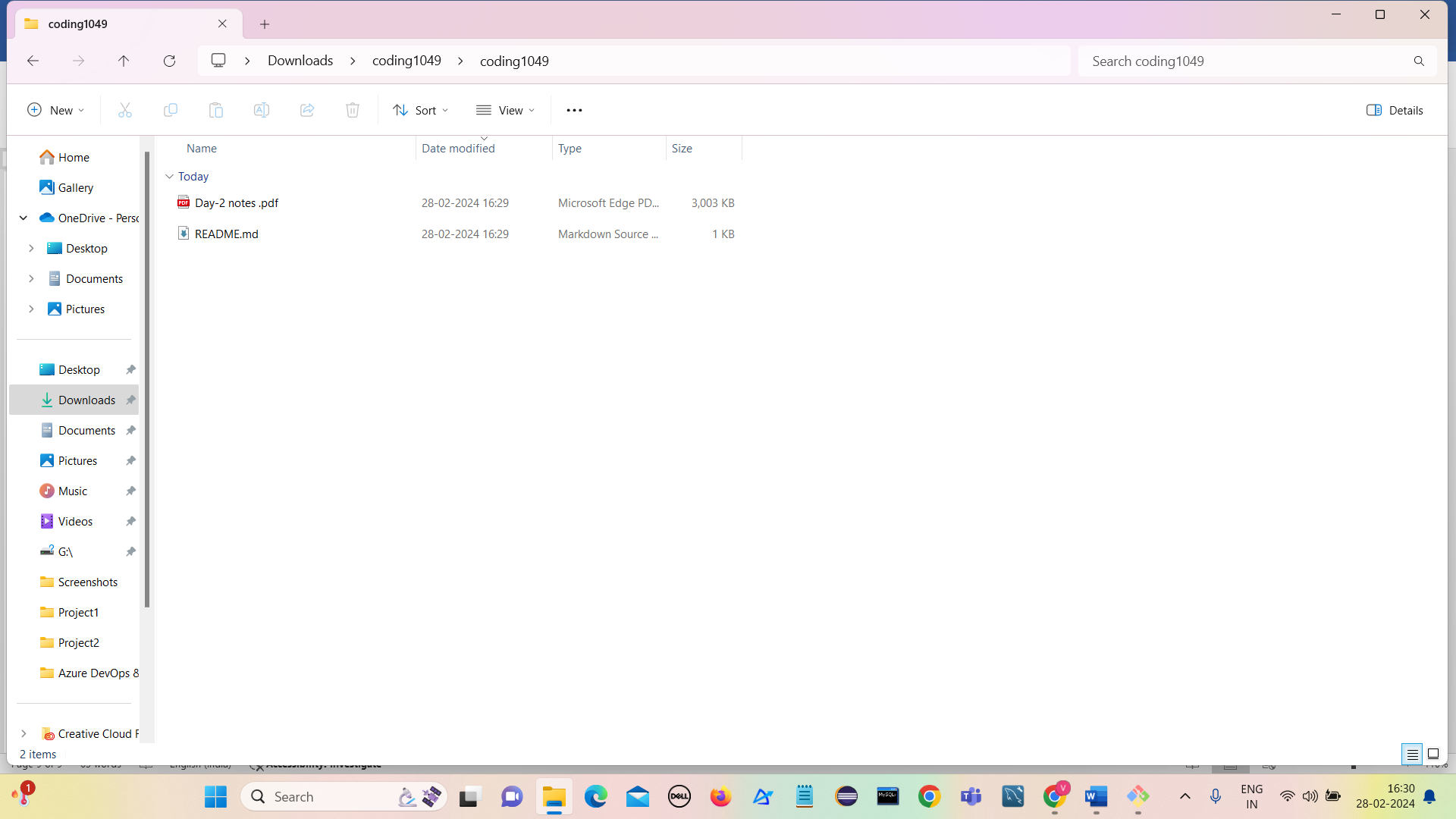


We observe that file got added.

**Pull the uploaded file from project to local**

Use the command git pull origin main



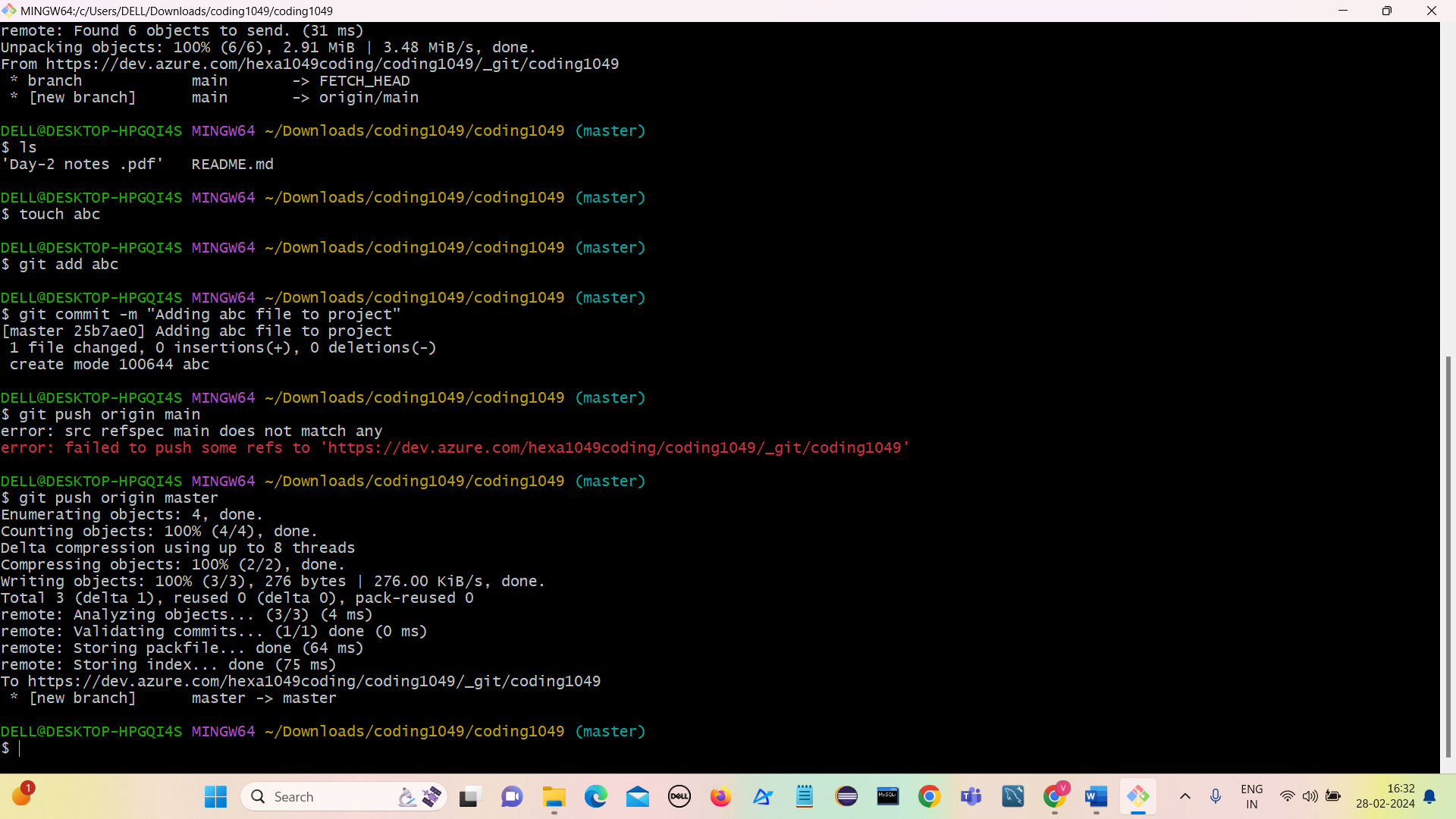


We observe that file got pulled into local from the repository.

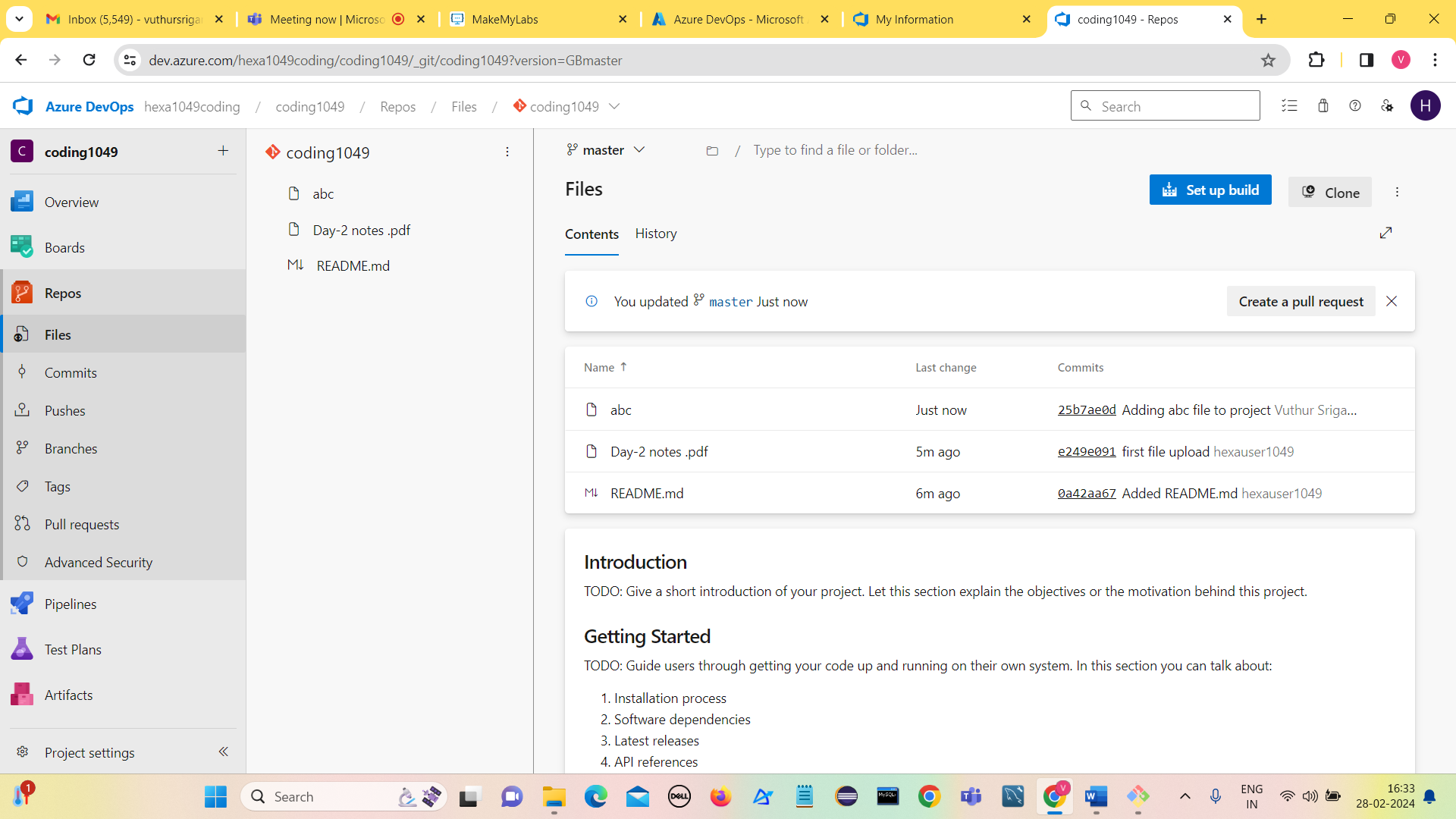
**Add files in project using git**

**Commands used are:**

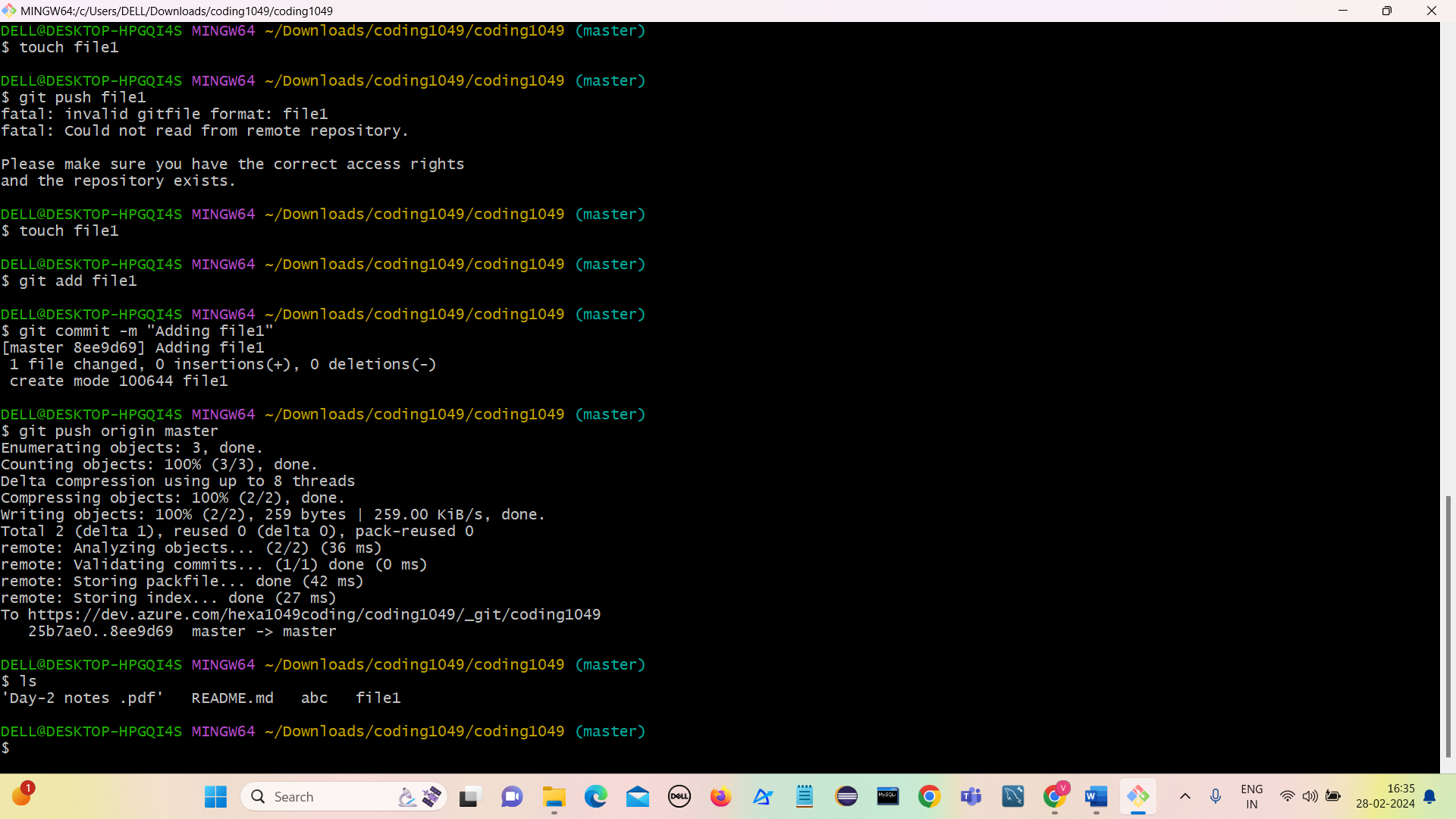
1. touch abc
2. git add abc
3. git commit -m “Adding abc file to project”
4. git push origin master



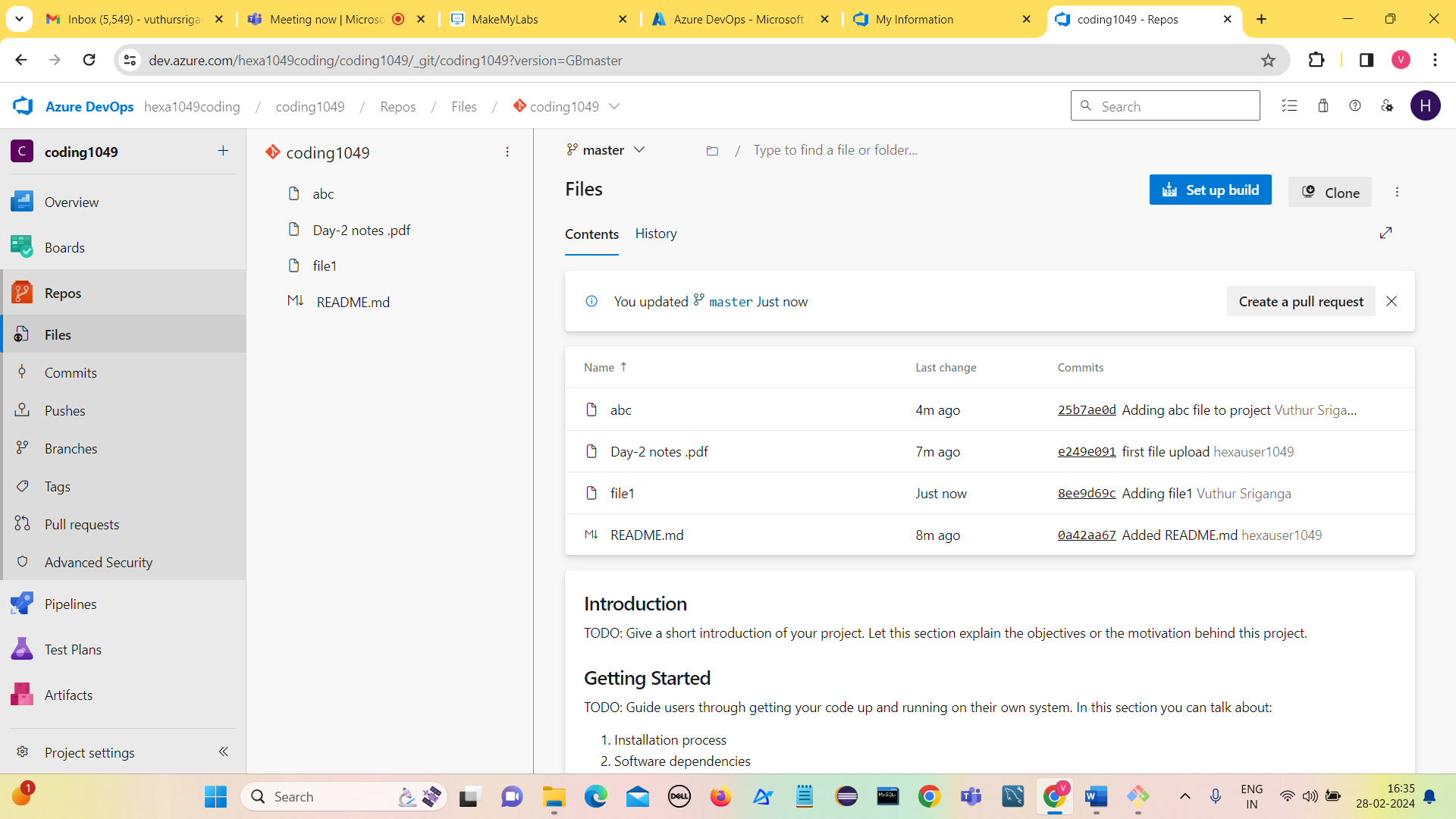
abc file got added to project

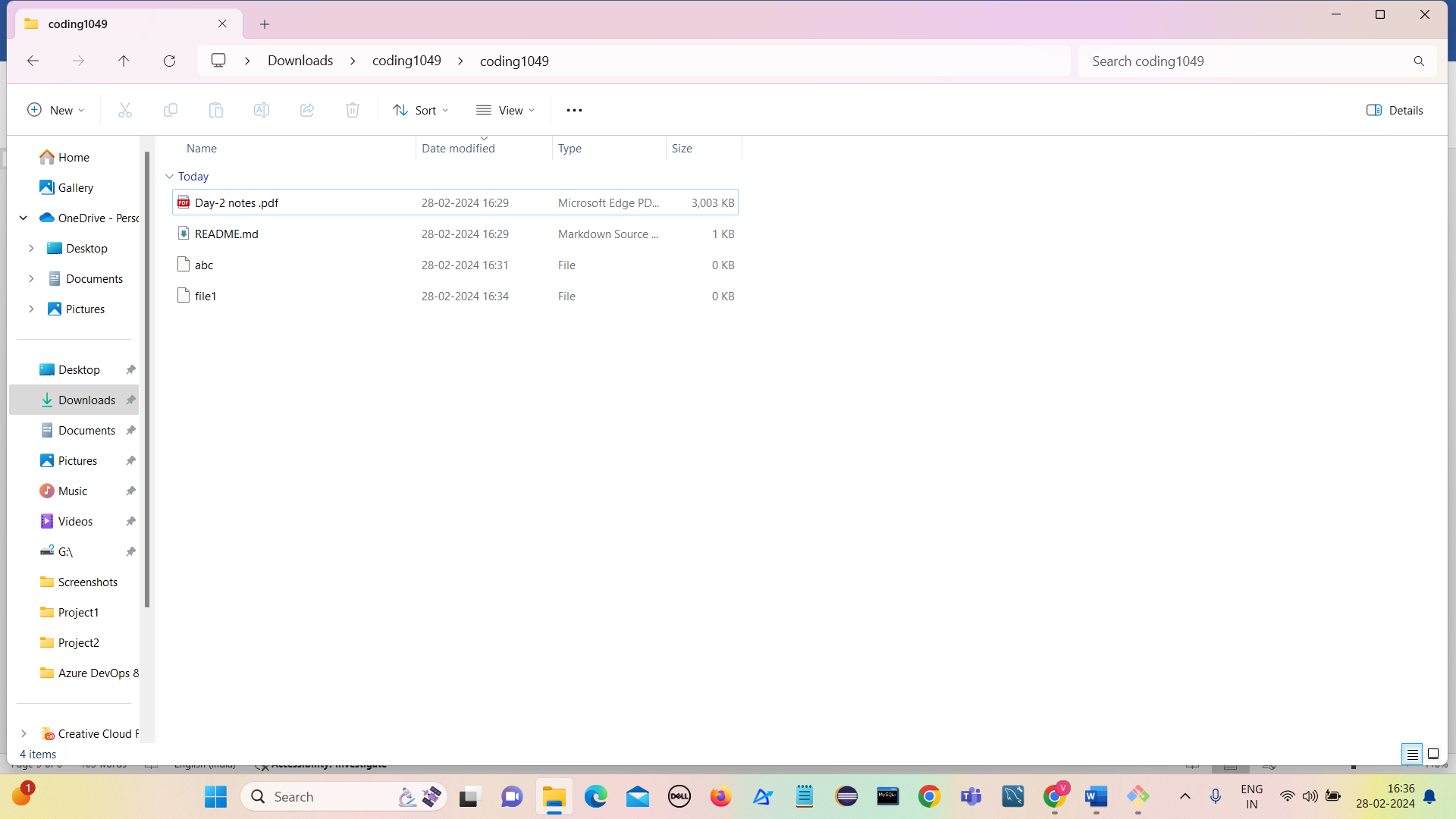


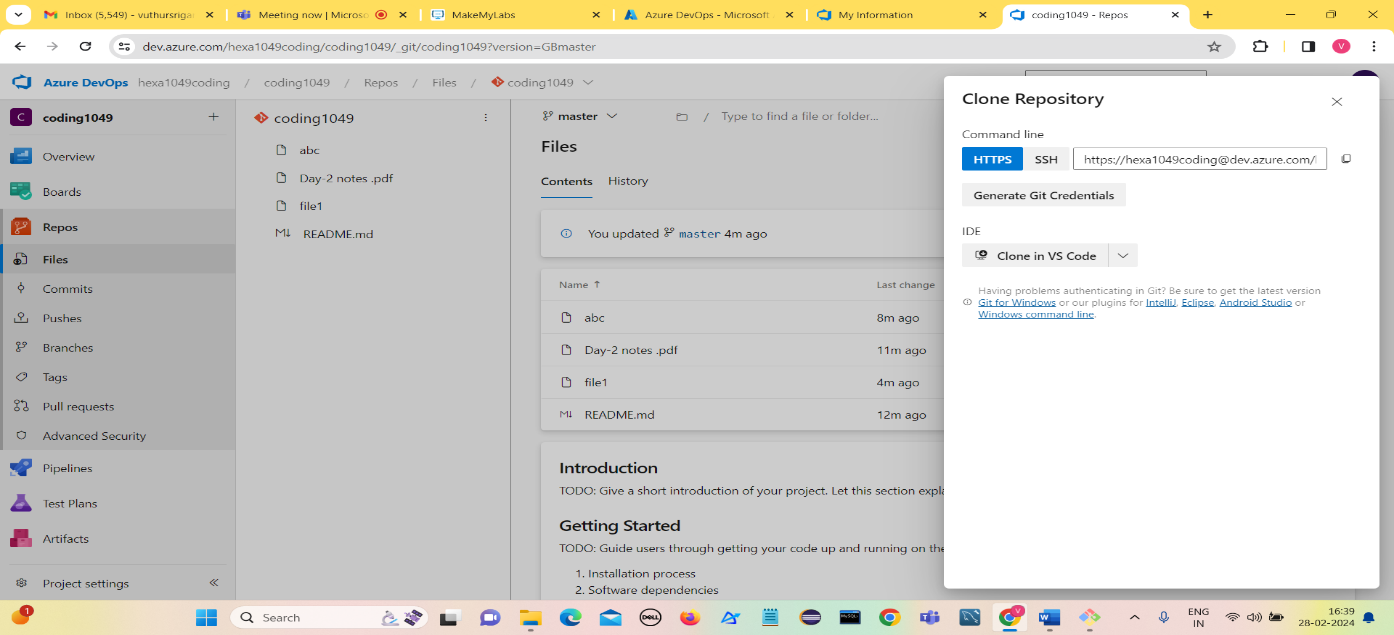
Adding one more file file1

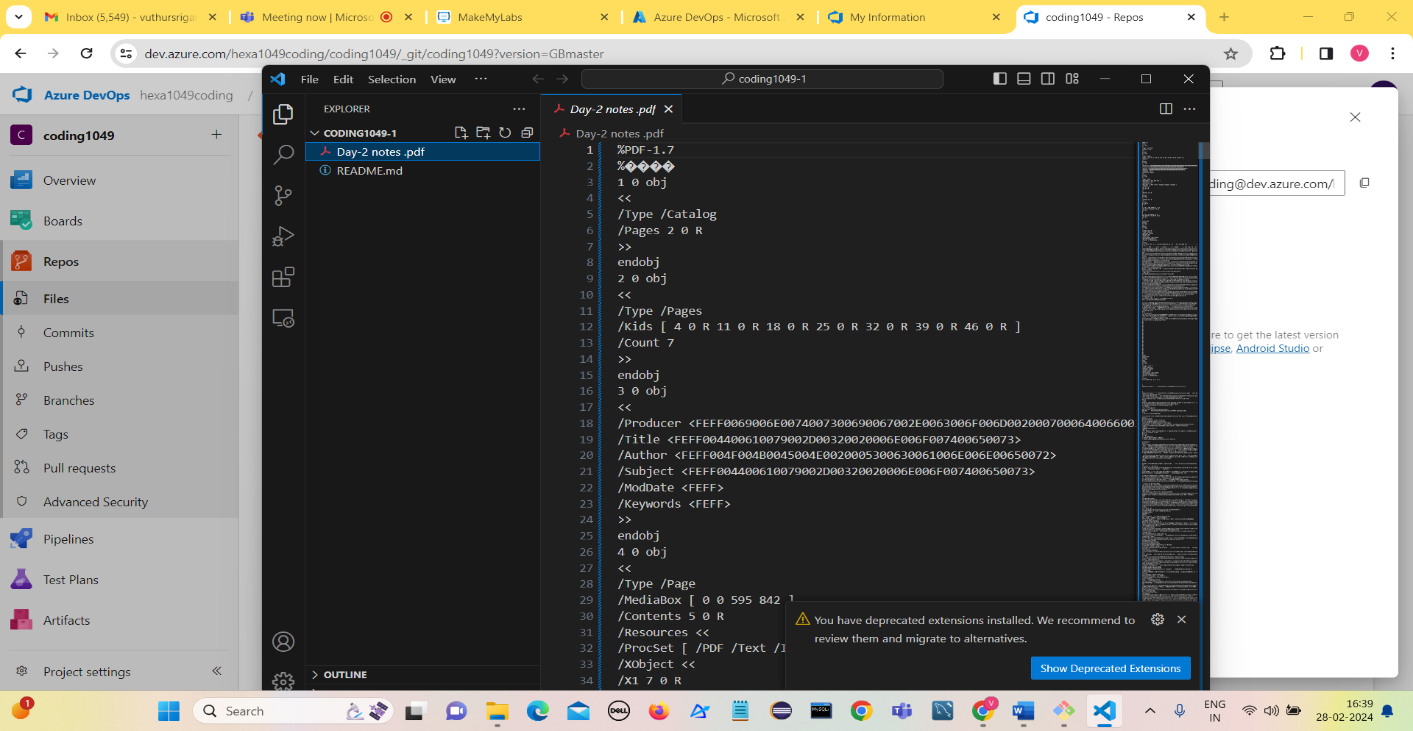


File1 got added





**Cloning using VS code**



**git commands used:**git init

touch filename  
git add (filename)  or git add .  
git commit -m "samplecommitmsg"  
git push origin main/master (local to azure repos)  
git pull origin main/master (azure repos to local)

git push -u origin --all

git checkout main

1. **Leverage the practices of CI/CD Using Azure Data Engineering**

**Leverage the practices of CI/CD Using Azure Data Engineering:**

* CI stands for Continuous Integration and CD stands for Continuous Deployment.
* CI/CD in deployment of data pipeline mainly focuses on automating data operations and transformations.
* Using CI/CD for data pipeline automation has become more critical in ensuring the development velocity of processes.
* They include training machine learning models, supporting a data science team, doing large-scale data analysis, business intelligence or data visualization, supporting the growth of unstructured data collection, and other business needs.

**Example:**

We are building a toy train track. Every time we add a new piece, we want to ensure it fits perfectly and train doesnot come out from track.

**Continuous Integration:** Every time we add a new track piece, we immediately test it by running the toy train through it. This ensures that adding new piece has not caused any problems. If there's an issue, we can fix it immediately before it becomes a bigger problem.

**Continuous Deployment:** Once it is confirmed that our new piece fits and the train runs smoothly, we will show it everyone stating it is working perfectly. In other words, as soon as your changes are verified, they're made live and functional in the main track (production environment).

**Continuous Integration:**

CI checks and tests every new piece of code (or data transformation logic) you add to your data pipeline.

**Continuous Deployment:**

CD ensures that once tested and approved, this code gets added to the live system without manual intervention.

**Uses of Continuous Integration:**

1. Automated Testing
2. Version Control
3. Consistent Environment
4. Data Quality Checks

**Uses of Continuous Deployment:**

1. Automated Deployment
2. Monitoring and Alerts
3. Rollbacks
4. Infrastructure as Code (IaC)
5. **Explain the architecture of the Azure Synpase** .

* **Azure Synapse Analytics is designed to streamline the process of ingesting, preparing, managing, and analyzing data at scale, empowering organizations to derive valuable insights and make data-driven decisions more efficiently.**
* **It is a cloud-based analytics service provided by Microsoft Azure.**
* **It integrates big data and data warehousing functionalities, allowing users to analyze and visualize large volumes of data for business intelligence and decision-making purposes.**
* **It combines enterprise data warehousing, big data analytics, and data integration into a single service, providing a unified experience for data engineers, data scientists, and business analysts.**

**Components of Azure Synapse Architecture:**

1. **Synapse SQL**

Azure Synapse SQL is a distributed SQL engine designed for large-scale data processing. It is based on Microsoft SQL Server and can handle large data workloads. Azure Synapse SQL allows users to query data stored in various sources, including data lakes,

data warehouses, and operational databases, using standard SQL syntax. It also supports both serverless and dedicated options for processing queries, depending on the user’s needs.

1. **Dedicated SQL Pool**

* The aim of utilizing Dedicated SQL Pools is to effectively store a large amount of data and able to perform queries efficiently.
* It stores the data in a columnar format for easy retrieval.
* Its performance is determined by the Data Warehouse Units (DWUs) we choose while creating the resource.
* One of the benefits is that we can temporarily halt its resources to save on costs and start it again whenever required.
* However, you should note that storage costs still apply even when the Dedicated SQL Pool is paused.

1. **Serverless SQL Pool**

* The Serverless SQL Pool is a tool for querying data in your data lake.
* We can use it to access your data with T-SQL syntax, which lets us to query the data without needing to copy or load it into a specific storage location.
* We can also connect to the tool through the T-SQL interface and use various business intelligence and ad-hoc querying tools, including commonly used drivers.

1. **Spark**

* Spark in Azure Synapse Analytics is fully managed and comes with pre-installed libraries for Python, R, and Scala.
* The Spark component in Azure Synapse Analytics is highly scalable, enabling users to process large amounts of data quickly and efficiently.
* It also integrates with other Azure services such as Azure Data Factory, Azure Stream Analytics, and Azure Event Hubs.

1. **Synapse Pipelines**

* Azure Synapse Pipelines is a cloud-based data integration service that allows users to create, schedule, and manage data pipelines.
* It is part of the Azure Synapse Analytics workspace and provides an graphical interface for building and managing data pipelines.
* With Synapse Pipelines, users can connect to various data sources, transform and process data using a wide range of built-in transformations and custom code, and load the data into various destinations, including Azure Synapse Analytics, Azure Data Lake Storage, and Azure SQL Database.

1. **Synapse Studio**

* Azure Synapse Studio is a web-based integrated development environment (IDE) that provides a unified experience for data integration, data warehousing, big data, and AI tasks.
* It is a key component of the Azure Synapse Analytics architecture and provides a collaborative workspace for data engineers, data scientists, and business analysts.
* Synapse Studio allows users to create and manage multiple Synapse workspaces from a single location.

**Vuthur Sriganga**

**vuthursriganga@gmail.com**