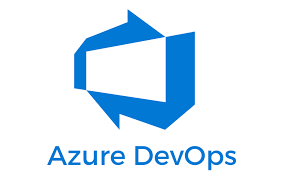
# **AZURE DEVOPS**

**Task-4**

**🔷 What is Azure DevOps?**

**Azure DevOps** is a suite of development tools offered by **Microsoft** that helps software teams **plan, develop, test, deliver, and monitor software** efficiently. It supports the entire software development lifecycle (SDLC) using modern DevOps practices.

It is built on the **cloud-based Azure platform**, but it can also be integrated with on-premises environments.



**🎯 Purpose of Azure DevOps**

The main goal of Azure DevOps is to:

* Enable **collaboration** among development, testing, and operations teams.
* Automate the software **build, test, and deployment** processes.
* Improve **code quality** and reduce time to market.
* Support **Agile, Scrum, and DevOps** practices seamlessly.

**🧰 Core Features of Azure DevOps**

Azure DevOps includes 5 main services:

**1. Azure Repos**

* Git repositories for source control.
* Branching, pull requests, and code reviews.

**2. Azure Pipelines**

* Build and release pipelines for **CI/CD**.
* Supports multiple languages: .NET, Java, Node.js, Python, etc.
* Compatible with Windows, Linux, and macOS agents.

**3. Azure Boards**

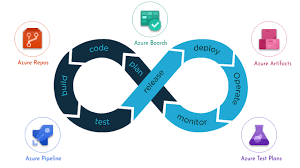
* Work tracking with **Kanban**, **Scrum**, and **Agile** boards.
* Manage user stories, sprints, backlogs, bugs, and tasks.

**4. Azure Test Plans**

* Manual and exploratory testing tools.
* Integrated test case management.

**5. Azure Artifacts**

* Package management for Maven, npm, NuGet, and Python.
* Create and share packages across projects.



**🔧 Additional Features**

* **Extensions Marketplace** – Add functionality from 1000+ extensions.
* **YAML-based Pipelines** – Store your build and release definitions as code.
* **Role-based access control (RBAC)** – Secure your environment.
* **Auditing and Logs** – Keep track of all activities and changes.
* **Multi-cloud support** – Deploy to **AWS, GCP**, or on-premise environments.

✅ **Key Benefits of Azure DevOps**

| **Benefit** | **Description** |
| --- | --- |
| **🔄 End-to-End DevOps** | Supports the complete development lifecycle – from planning to monitoring**.** |
| **☁️ Cloud-based or On-prem** | Available as a cloud service (Azure DevOps Services) or a self-hosted server (Azure DevOps Server). |
| **🤝 Seamless Collaboration** | Helps developers, testers, and project managers work together efficiently. |
| **⚙️ CI/CD Pipelines** | Built-in Continuous Integration and Continuous Delivery support. |
| **📊 Analytics & Insights** | In-depth dashboards and reporting for work items, builds, releases, etc**.** |
| **🔐 Secure & Compliant** | Microsoft-grade security and compliance for enterprise use. |
| **📁 Integrations** | Integrates with GitHub, Jenkins, Slack, Teams, and other tools. |

**📌 Why Use Azure DevOps?**

* **Centralized Toolset**: Everything in one platform.
* **Enterprise Ready**: Scales well for large organizations.
* **Flexible**: Works with any language, platform, or cloud.
* **Automated**: Automate everything from code commits to production deployments.
* **Agile-Friendly**: Plan sprints, monitor velocity, and improve continuously.

**🏗️ Architecture of Azure DevOps**

Azure DevOps follows a **modular service-oriented architecture**. Each DevOps function (like code, build, deploy, test, and monitor) is handled by a separate **service module**, all of which are tightly integrated into a single platform.

+---------------------+

| Azure DevOps UI | (Web Portal / Dashboard)

+---------------------+

|

+----------------+---------------------+

| | |

+----------------+ +----------------+ +----------------+

| Azure Boards | | Azure Repos | | Azure Pipelines |

+----------------+ +----------------+ +----------------+

| | |

| | |

| | +----------------+

| | | Build Agents |

| | +----------------+

| | |

+--------------------------------------------------+

| Shared Project, Code, Artifacts, Users |

+--------------------------------------------------+

|

+------------------+

| Azure Artifacts |

+------------------+

|

+------------------+

| Azure Test Plans |

+------------------+

↓

+---------------------------+

| Deployment Targets (Cloud,|

| On-prem, Multi-cloud) |

+---------------------------+

↓

+-------------------------+

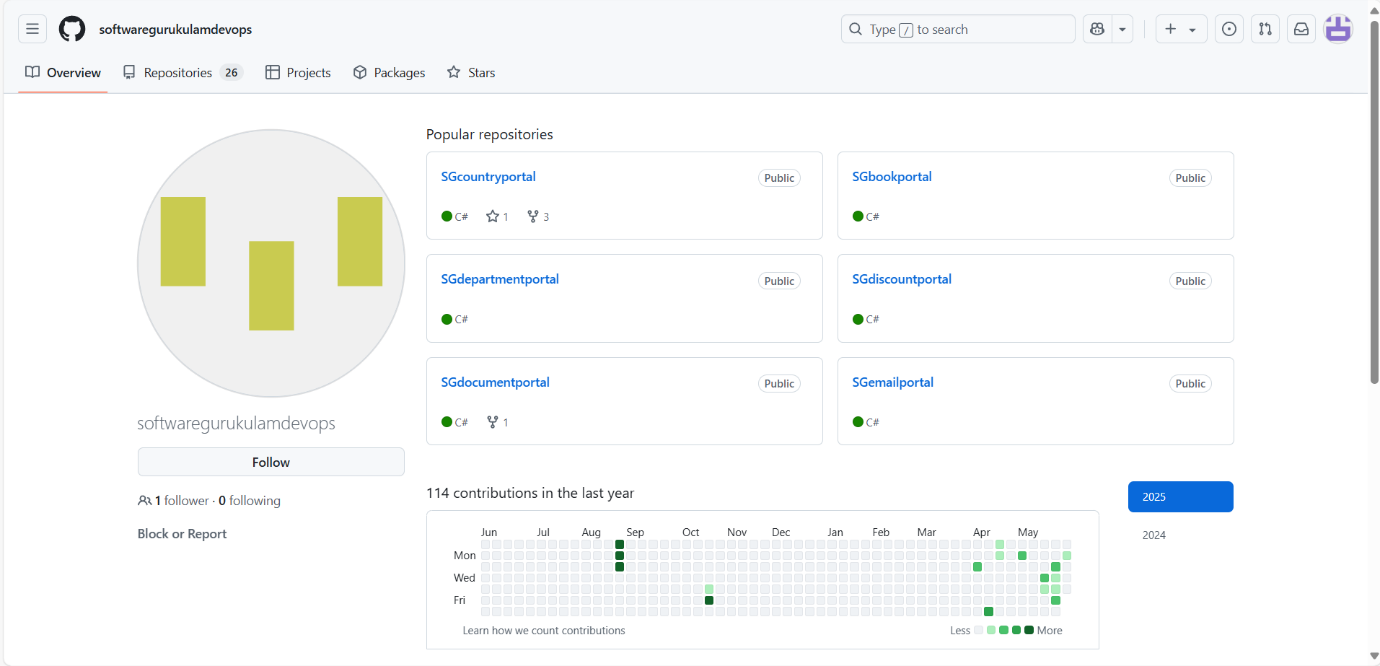
| Monitoring & Feedback |

+-------------------------+

🛠️ Steps to Push Code from GitHub to Azure DevOps

**🔁 Overview:**

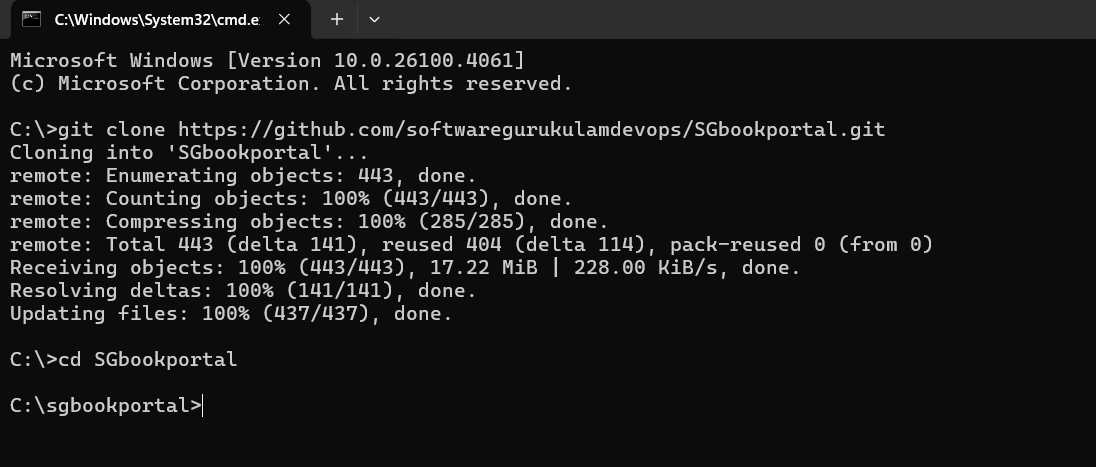
1. Clone the GitHub repo locally
2. Add Azure DevOps as a new remote
3. Push code to Azure DevOps



* GO to the GIT hub account and select your repo. click on the repo and copy the code .

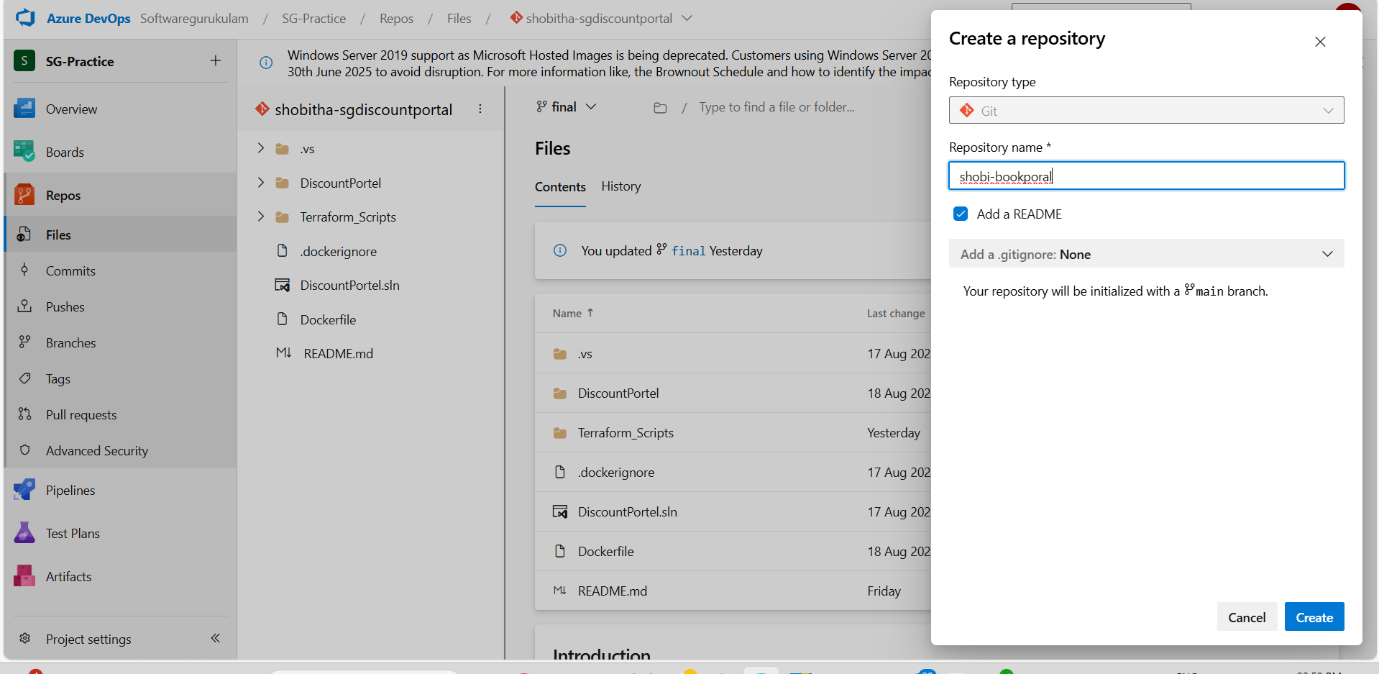
🧑‍💻 **Step 1: Clone the GitHub Repo Locally**

* Open your local repo and create a folder…….open cmd
* **git clone** https://github.com/your-username/your-github-repo.git
* **cd** your-github-repo



**🔧 Step 2: Create a New Repo in Azure DevOps**

1. Go to **Azure DevOps** → Your Project → **Repos**
2. Click **"New repository"**
3. Choose **Git**, give it a name, and click **Create**
4. Copy the **Azure Repo URL** (e.g., <https://dev.azure.com/ORG/PROJECT/_git/RepoName>)



➕ **Step 3: Add Azure DevOps as a Remote**

* **git remote add** azure https://dev.azure.com/ORG/PROJECT/\_git/RepoName

You now have 2 remotes:

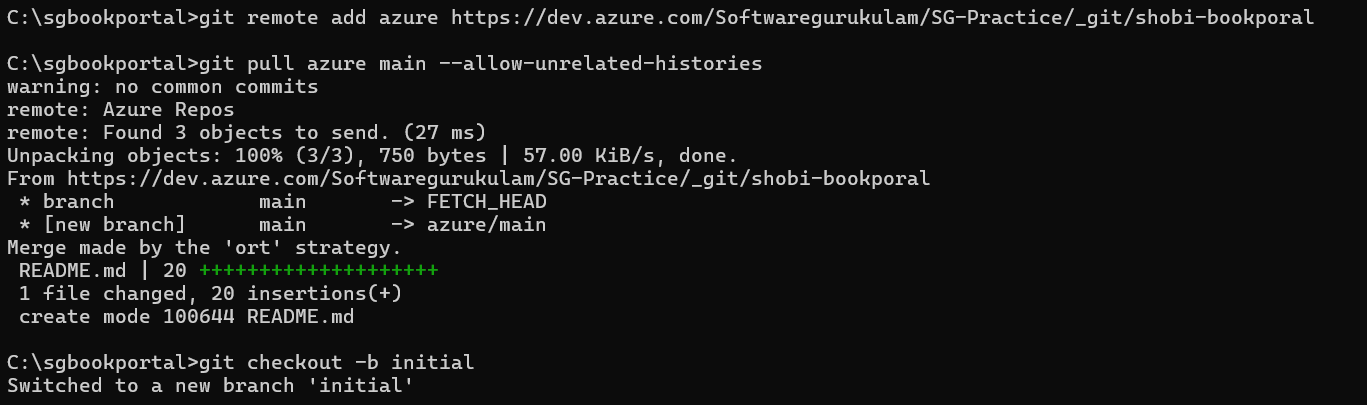
* origin → GitHub
* azure → Azure DevOps

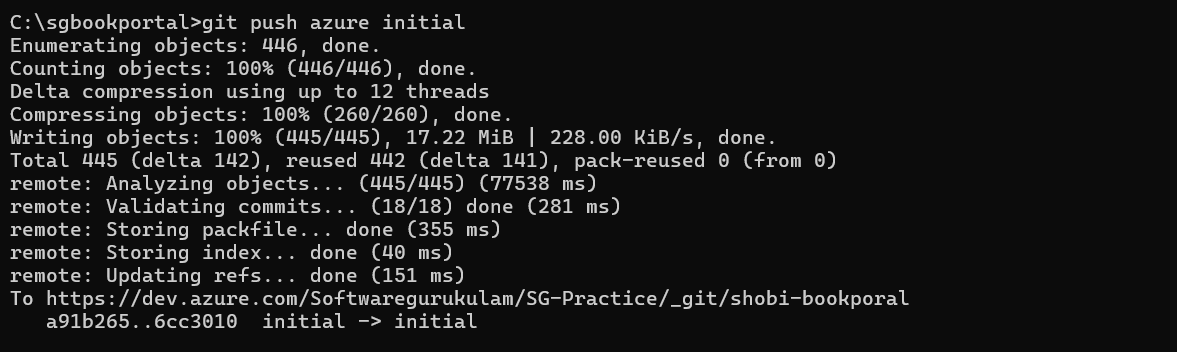
**🚀 Step 4: Push Code to Azure DevOps**

* git pull azure main --allow-unrelated-histories
* git push azure main --force

or

* git push azure main
* If you have permission to add files to main branch…..follow above commands
* If you don’t have permission to add files to main branch just create a another branch and send code to that branch (e.g…..initail,feature,) and give pull request(PR)……to main branch.
* All files go to main branch from feature(or) initial branch.
* **Git checkout -b** branch name
* **Git push azure** initial





**🔑 Step 5: Verify in Azure DevOps**

* Go back to Azure DevOps → Repos → Files
* You should see your GitHub code now available in Azure Repos.

