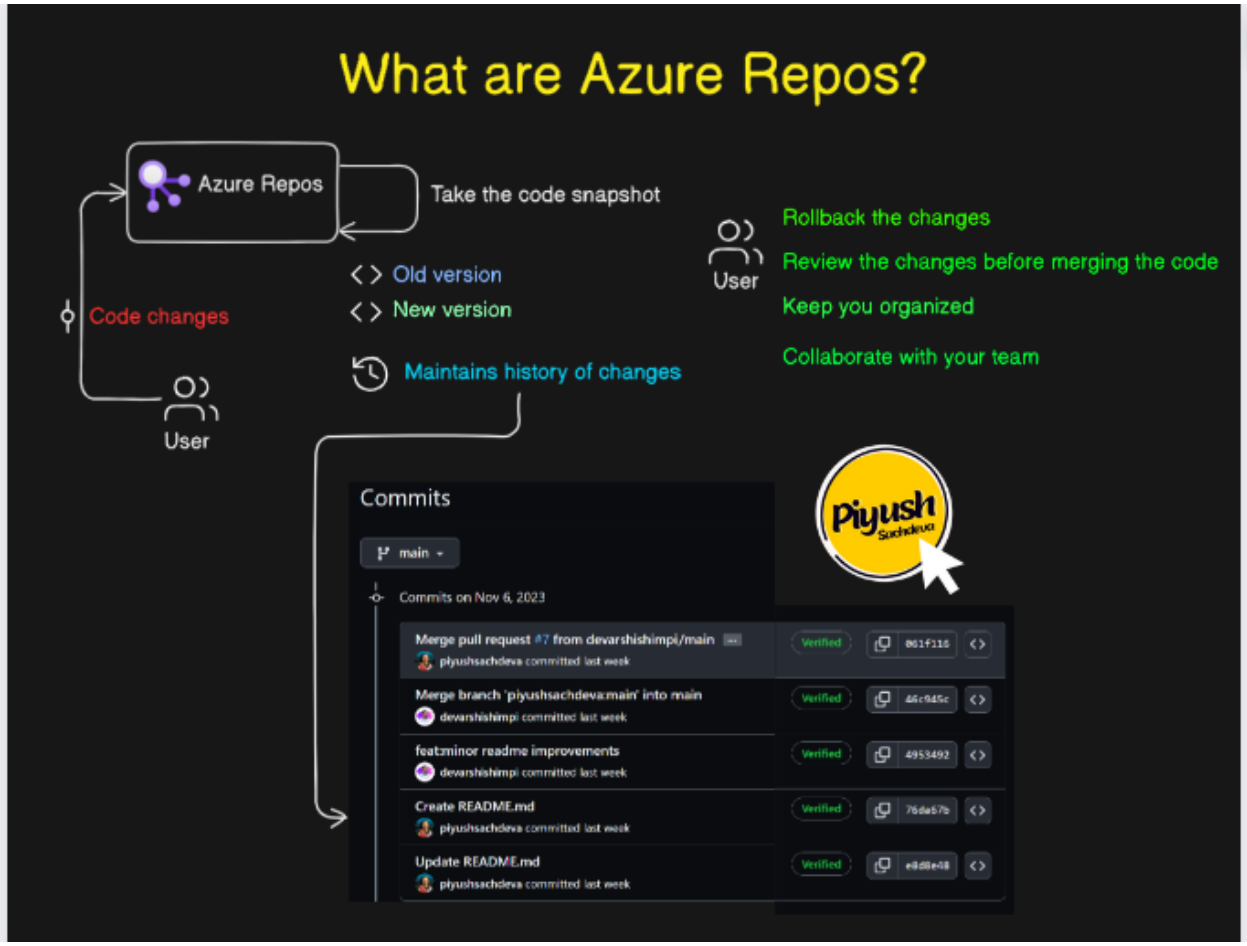
Azure Repos

Azure Repos is a set of version control tools that you can use to manage your code, just like GitHub. Whether your software project is large or small, using version control as soon as possible is a good idea.



Version control systems are software that helps you track changes you make in your code over time. As you edit your code, you tell the version control system to take a snapshot of your files. The system saves that snapshot permanently so you

can recall it later if you need it. Use version control to save your work and coordinate code changes across your team.

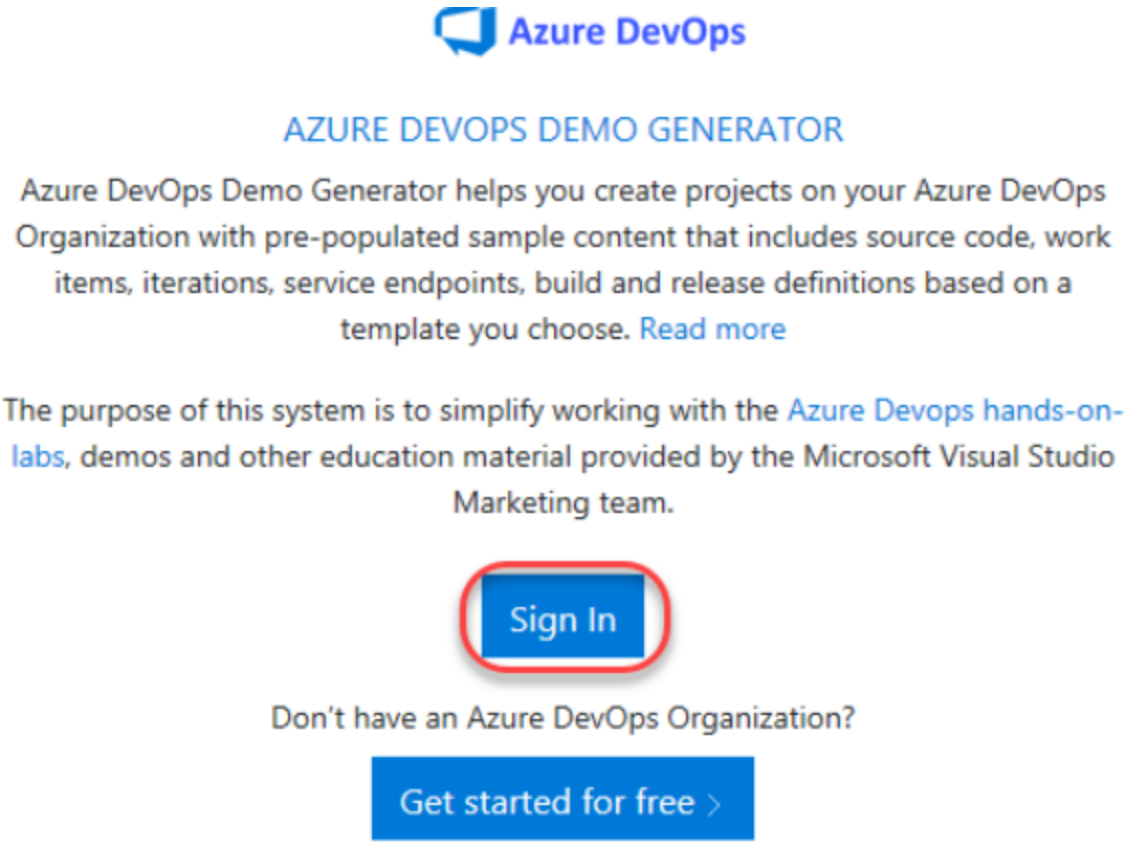
Even if you are working on a personal project, version control helps you stay organized as you fix bugs and develop new features. Version control keeps your development history so you can quickly review and even roll back to any code version.

- Helps you track changes in the codebase
- Maintains the history of your codebase, who made the changes, what changes were made, why the changes were made, etc
- Helps you stay organized
- Gives you the ability to rollback the changes as needed

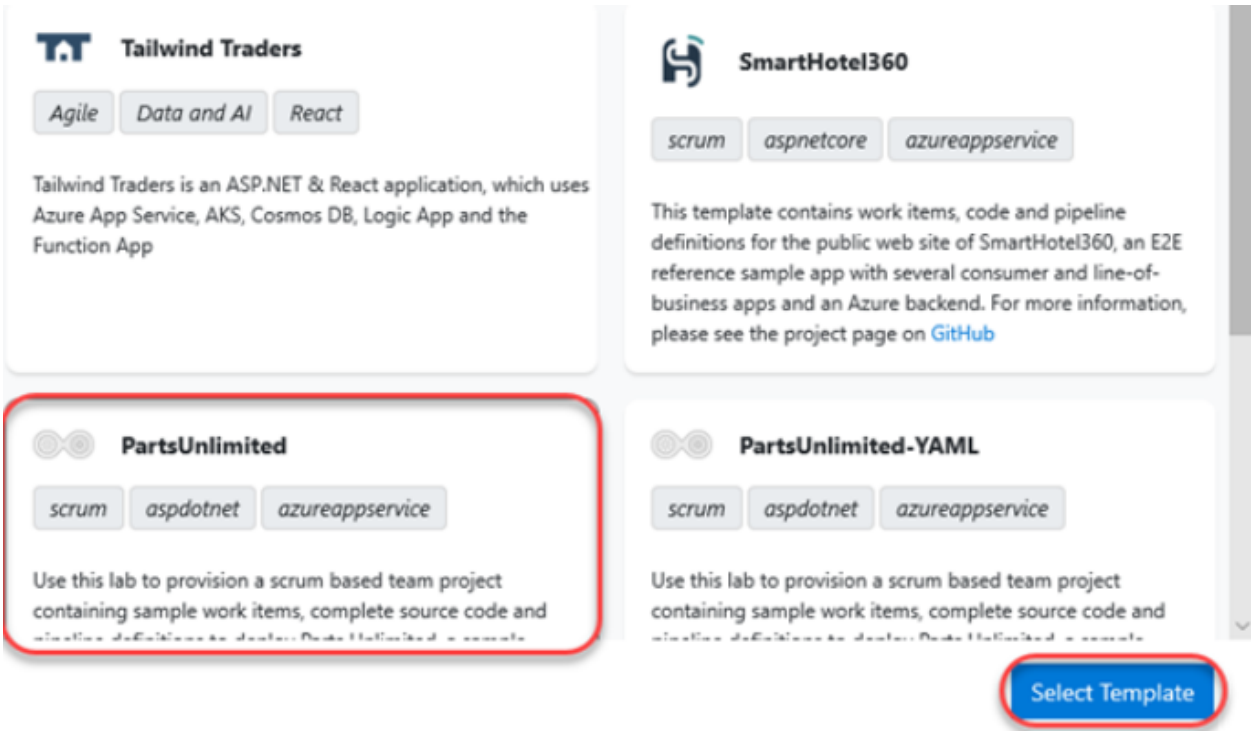
Azure DevOps Demo Generator

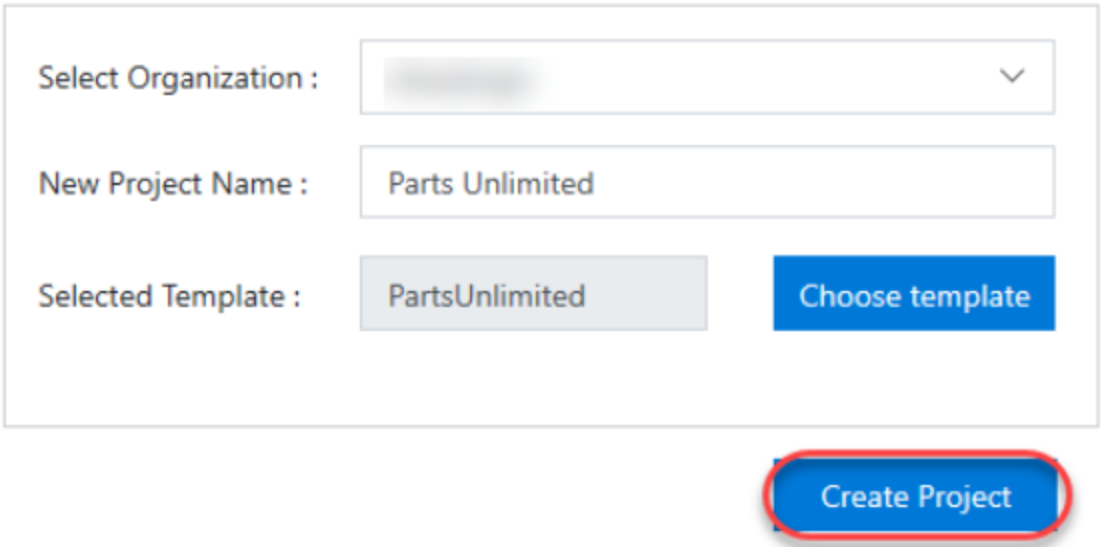
Use these steps to load dummy data into your Azure DevOps project. We will use this data in the demo.

1. Navigate to https://azuredevopsdemogenerator.azurewebsites.net. This utility site will automate the creation of a new Azure DevOps project within your account that is prepopulated with content (work items, repos, etc.) required for the lab.
2. Sign in using the Microsoft account associated with your Azure DevOps subscription.



1. Accept the permission requests for accessing your subscription.
2. Select the PartsUnlimited template and click Select Template.
3. Click Create Project and wait for the process to complete.

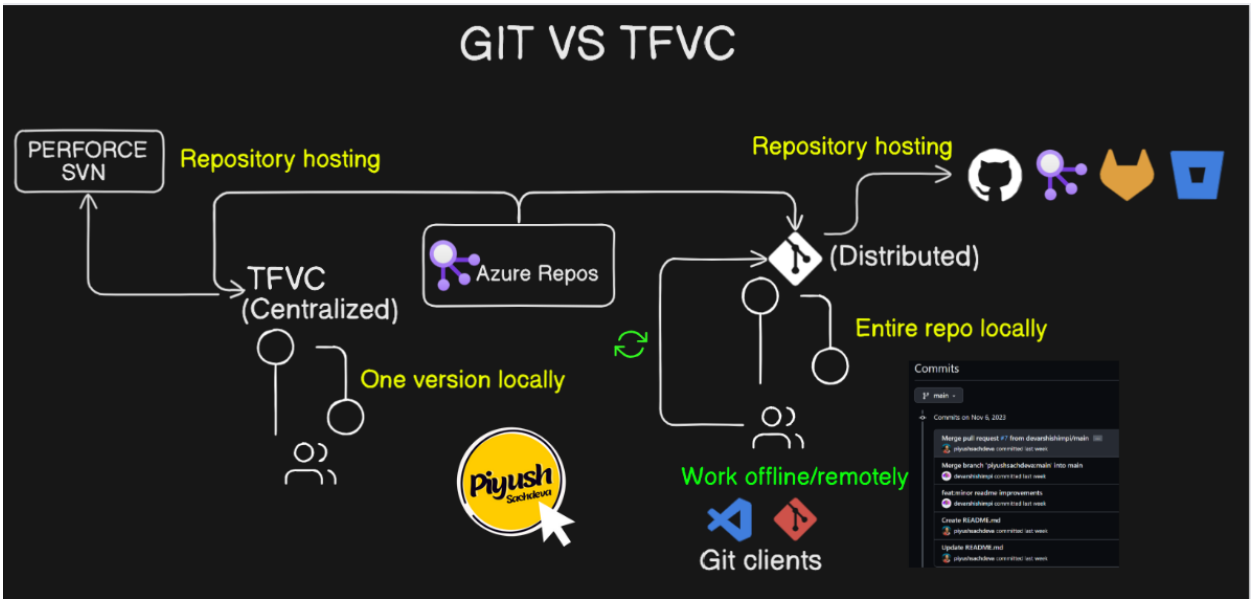




Git vs. TFVC

Azure Repos supports two types of Version Control:

- Git
- TFVC (Team Foundation Version Control)



Git

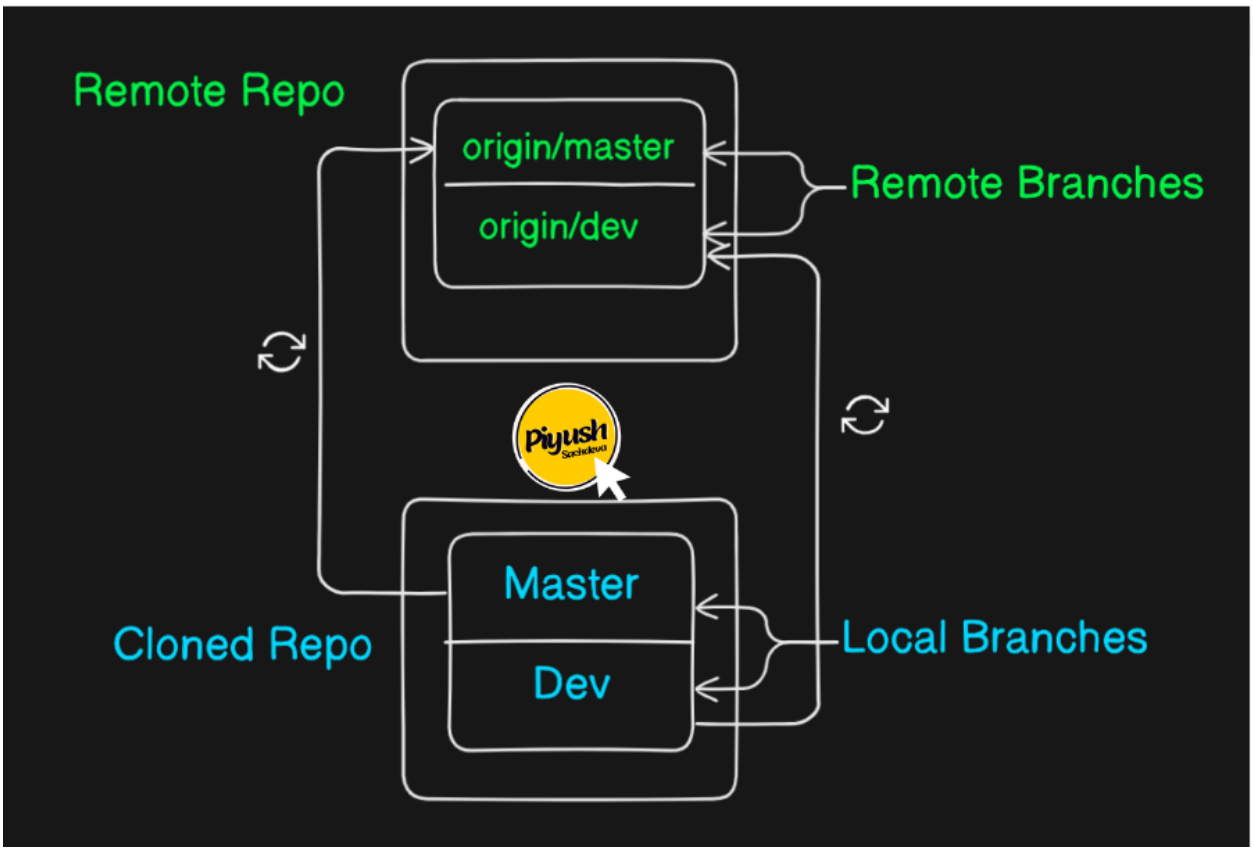
Git is the most popular distributed version control system. It allows developers to download the entire code repository locally with all the versions and make the changes remotely/offline. Afterward, changes can be synced to the remote server.

To interact with Git, you can make use of Git clients such as Git for Windows, VSCode, etc Git provides a version control system, but you need a hosting service to host your codebase(repositories). You can use git hosting services such as GitHub, Azure Repos, Bitbucket, Gitlab, etc.

TFVC

Another type of version control system is TFVC, a centralized version control. In TFVC, only a single codebase version is downloaded locally, and historical data is maintained on the central server. You can host TFVC using hosting services such as Perforce, SVC, Azure Repos, etc.

Working with branches



A **local** branch exists solely on your local machine. All the changes you make and commit to your local repository are stored only on your local system. To create a branch, use the command below.

```
git branch <branchname>
git checkout -b <branchname>
git branch <branchname> <tag>
git branch <branchname> <commit id>
```

Remote branches are how developers collaborate on the same project simultaneously. A remote branch exists in a remote repository (most commonly called origin by convention) and is hosted on a platform such as GitHub and Azure Repos.

```
git remote -v
git remote add origin git@github.com:github URL
git push origin master
```

When a Git repository is cloned, the default remote repository created is called the "**origin**" remote. It's typically the central repository for sharing changes and collaboration with other developers.