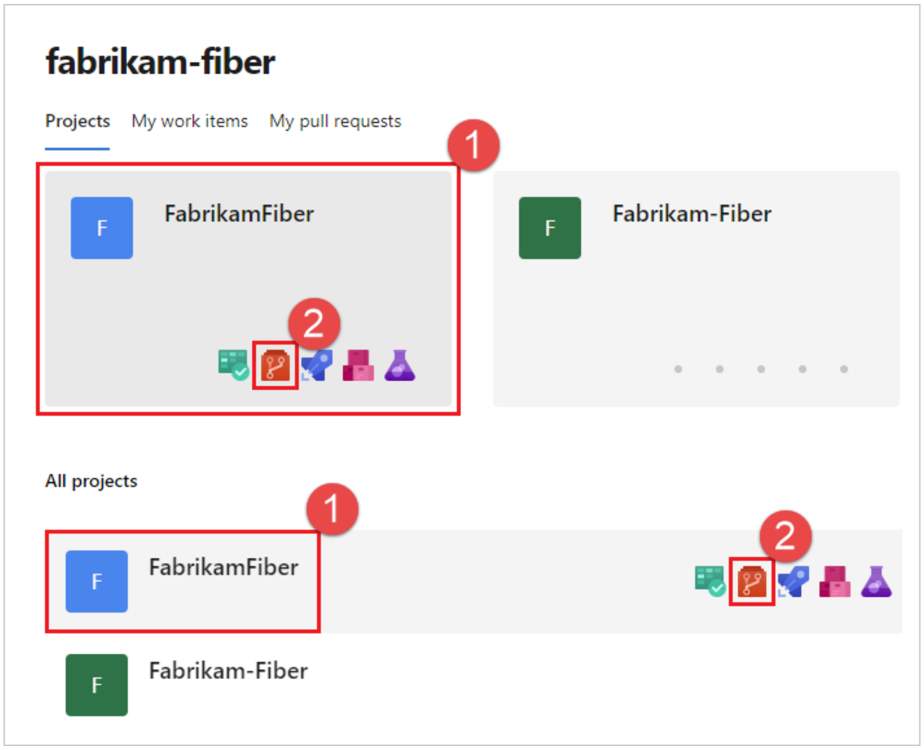
Create a new Git repo in your project

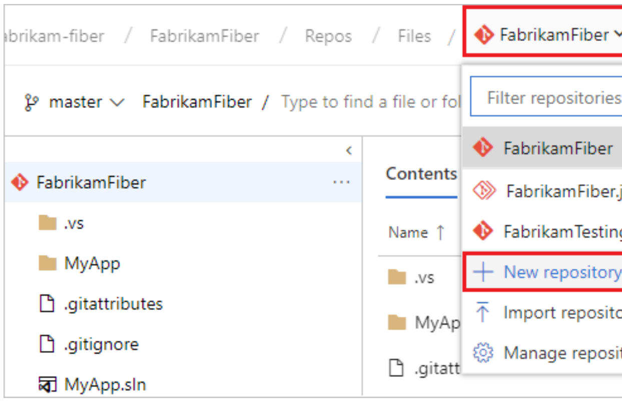
Azure DevOps Services contain Git repositories, work items, builds, and releases. You can use Git repos in your projects to manage your source code, particularly as your project grows. This document will detail how to create a Git repo using the web portal for Azure DevOps Services.

Prerequisites

- An organization in Azure DevOps. If you don't have one, you can sign up for one for free. Each organization includes free, unlimited private Git repositories.
- A project in Azure DevOps as detailed here. # Create a repo using the web portal

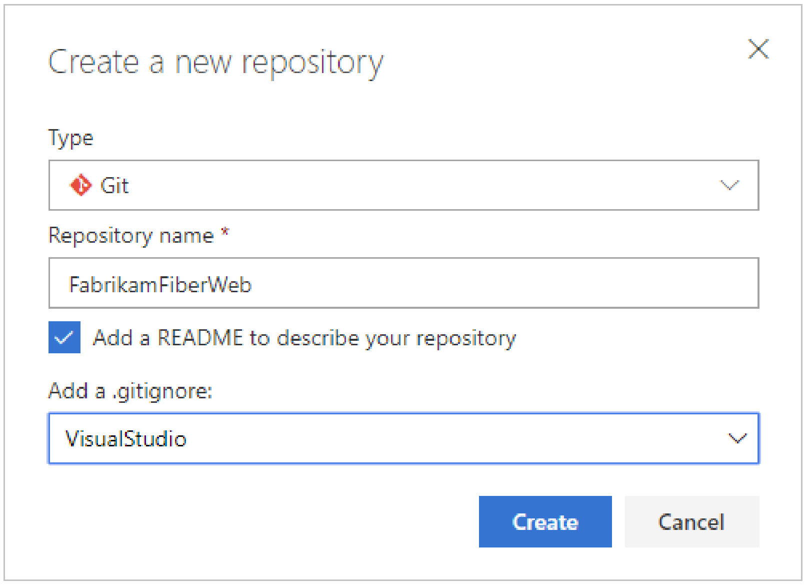
1. Navigate to the **Repos** page in your project, hover your mouse over the name of your project and select the **Repos** icon.





2. From the repo drow-down, select **New repository**.

3. In the **Create a new repository** dialog, verify that Git is the repo type and enter a name for your new repo. You can also choose to add a README and create a .gitignore for the type of code you plan to manage in the repo. A README contains information about the code in your repo, and a .gitignore file tells Git which types of files to ignore, such as temporary build files from your development environment.



Type: Git

Repository name *: FabrikamFiberWeb

☑ Add a README to describe your repository

Add a .gitignore: VisualStudio

4. When you're happy with the repo name and choices, select **Create**.

A new empty Git repo is now created in your project.

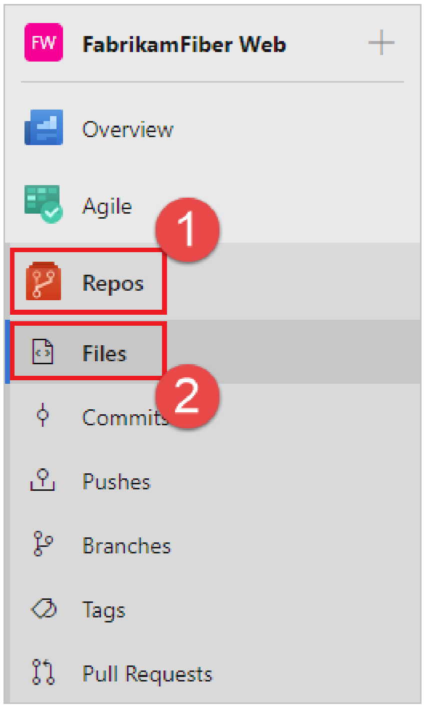
- If you created an empty repo (no README or .gitignore), you'll see instructions on how to clone the repo to your computer or push code in an existing repo into the newly created one. In this example you created a README and a .gitignore, so you'll see an overview of the files in your repo, and you can **clone** the repo using the Clone link on the upper right of the page to get working with a local copy of the repo immediately.

For details on how to clone the repo to your Azure ML JupyterLab click here. If you are working with another IDE such as Visual Code and want to work with a local cope ony our compute follow the steps below.

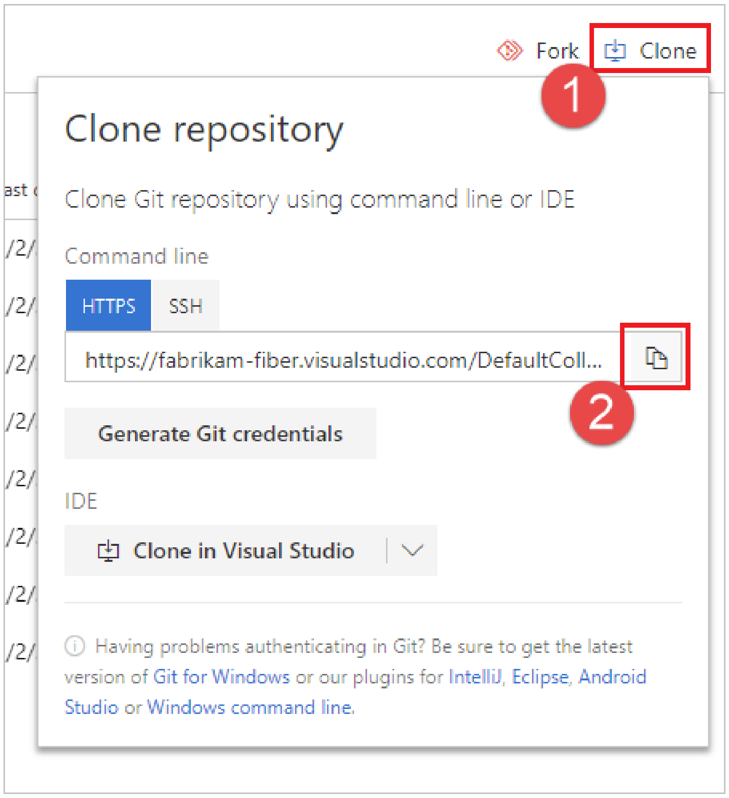
Clone the repo to your computer

To work with a Git repo, you clone it to your computer or other developing workspace. Cloning a repo creates a complete local copy of the repo for you to work with, and downloads all commits and branches in the repo and sets up a named relationship with the repo on the server. Use this relationship to interact with the existing repo, pushing and pulling changes to share code with your team.

1. From your web browser, open the team project for your organization in Azure DevOps and select **Repos** > **Files**. If you don't have a team project, create one now.



2. Select **Clone** in the upper-right corner of the **Files** window and copy the clone URL.



Clone repository

Clone Git repository using command line or IDE

Command line

https://fabrikam-fiber.visualstudio.com/DefaultColl...

Having problems authenticating in Git? Be sure to get the latest version of Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

3. Open the Git command window (Git Bash on Git for Windows) and browse to the folder where you want the code from the repo stored on your computer. Run `git clone` followed by the path copied from the **Clone URL** in the previous section, as shown in the example below.

```
git clone https://dev.azure.com/fabrikam-fiber/MyFirstProject/_git/
```

Git downloads a copy of the code, including all commits and branches from the repo, into a new folder for you to work with.

4. Switch your directory to the repository that you cloned.

```
cd fabrikam-fiber
```

Keep this command window open, because you'll use it in the following steps.

Example of working with the code

In this step, we'll make a change to the files on your computer, commit the changes locally, push the commit up to the repo that is stored on the server, and view the changes there.

4. Browse to the folder on your computer where you cloned the repo and open the `README.md` file in your editor of choice—a good edito to use when working with Git is Visual Code.

2. Make some changes, for example add `This is my first edit.` to the file, and save and close the file.

3. In the Git command window, navigate to your directory—in this example it would be `fabrikam-fiber`, by entering the follwing command:

```
cd fabrikam-fiber
```

4. Commit your changes by entering the following command in the Git command window:

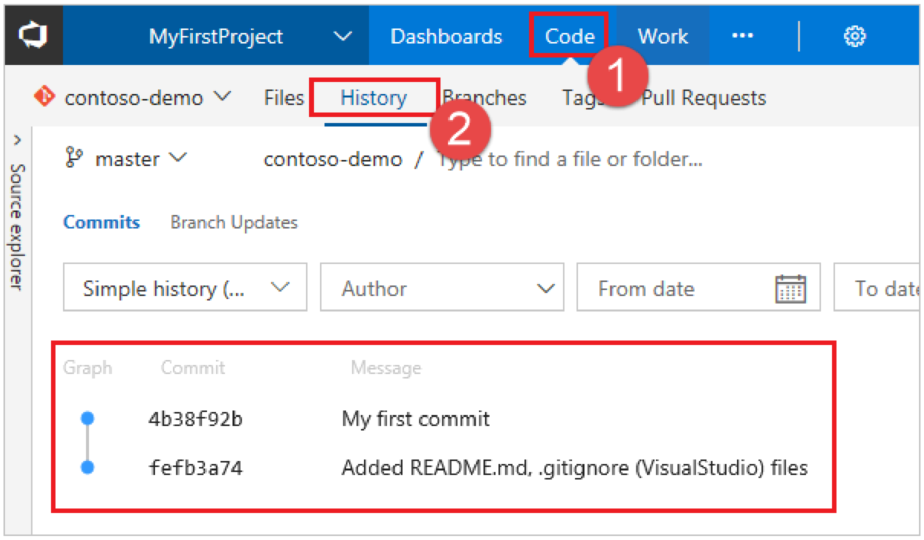
```
git commit -a -m "My first commit"
```

When using `git commit`, `-a` means to commit all changed files, and `-m` specifies a commit message.

5. Push your changes up to the Git repo on the server by entering the following command into the Git command window:

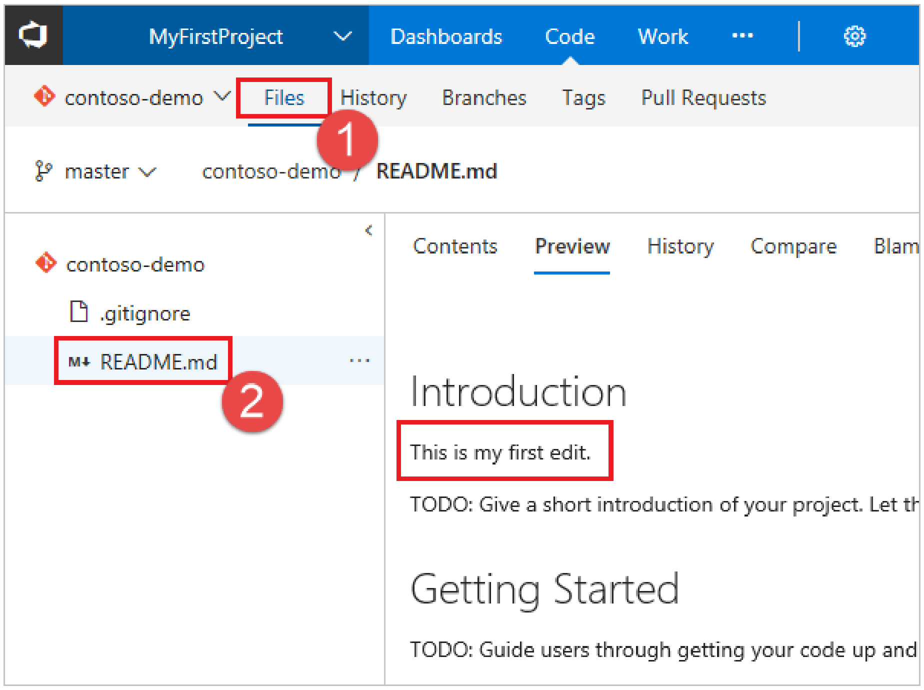
```
git push
```

6. Switch back to the web portal and select **History** from the **Code** view to view your new commit. The new repo has two commits: the first commit where the README and .gitignore were added when the repo was created, and the commit you just made.



Graph	Commit	Message
●	4b38f92b	My first commit
●	fefb3a74	Added README.md, .gitignore (VisualStudio) files

7. Switch to the **Files** tab and click on the README file to view your changes.



Introduction

This is my first edit.

TODO: Give a short introduction of your project. Let th

Getting Started

TODO: Guide users through getting your code up and

Congratulations! You now know how to create your first Git repo on Azure DevOps.