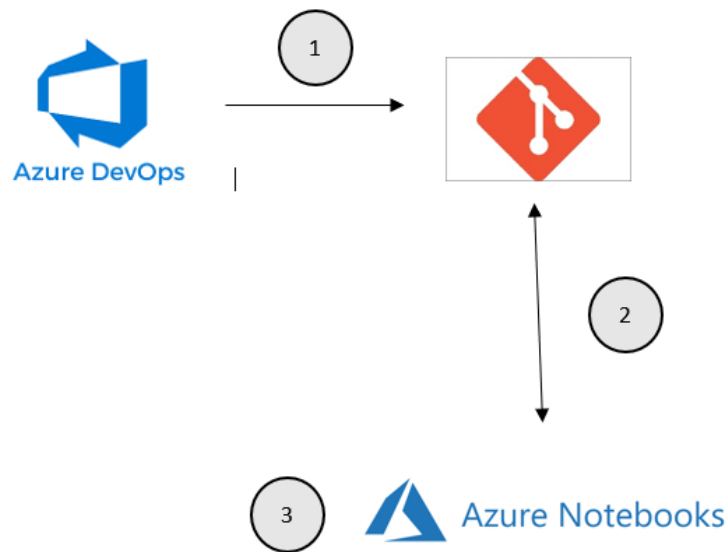


## Getting a project started using Azure DevOps, Git and Jupyter Lab

This document will help guide you on how to get a project started using Azure DevOps to manage your project, a Git repository to manage your code and a Jupyter Lab as your selected Integrated Deveopment Environment (IDE).

This guide is ideal if you want to get a small project started such as for a proof of concept or you would like to develop a better idea on how to use and integrate these tools. If you would like to use these tools for a substantial project within a larger team please refer to Roles and taks for an overview of the different roles and repsonsibilities.

This page will give you further links for detailed instructions on your role and how to get started with your project. Details covered include how to get started with Azure DevOps, creating a project repository and setting up security control—adding project members and configuring permissions.



*An overview of the tools used for this solution*

1. Azure DevOps Use Azure DevOps to create and manage your project. You can use this to break down your project into smaller tasks for e.g. ingest data, preprocess data, build features, etc.

2. **Git Repository** Create a git repo for your project to manage your code. Git is a distributed version control system. Git repositories can live locally (such as on a developer's machine). Each developer has a copy of the source repository on their dev machine. Developers can commit each set of changes on their dev machine and perform version control operations such as history and compare without a network connection.
3. **Azure ML Notebooks** Launch an Azure Machine Learning (ML) workspace, and clone the git repo onto your directory in the workspace's shared workspace file system. Once cloned you can use Jupyter notebooks to write your code. With your workspace now connected to your git repo you can **commit** (save) changes and **push** them back to your git repo in Azure DevOps. If files are added to the git repo you and colleagues can **pull** these changes onto your workspace making collaborating with colleagues easy.

## 1. Create a project in Azure DevOps

### Prerequisites

- You need an organization before you can create a project.

If you haven't created an organization yet, create one by following the instructions in Sign up, sign in to Azure DevOps, which also creates a project.

Or see Create an organization or project collection. If you have a Microsoft account follow the instructions for the heading **Sign up with a personal Microsoft account** here and then refer back to this documentation.

If you are going to sign up with a newly created Microsoft account (MWS), your project is automatically created and named based on your sign-in.

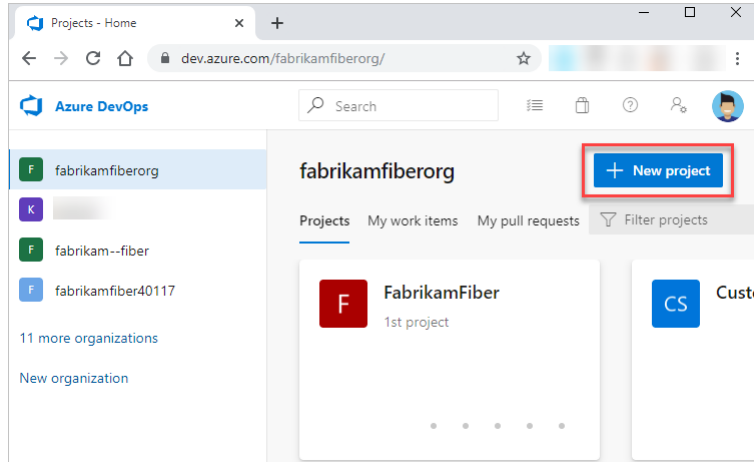
- To create a new project you must be a member of the Project Collection Administrators group or have the **Create new projects** permission set to **Allow**. If you're the Organization Owner, you're automatically added to the Project Collection Administrators group. If you aren't a member, get added before following this documentation. For more information, see Set permissions at the project- or collection-level.

[Important] To create a public project, or to make a private project public, see Create a public project in your organization or Change the project visibility, public or private. Additional policy settings must be enabled to work with public projects.

### Create a project

1. Navigate to your organization page <https://dev.azure.com/{yourorganization}>.
2. Select **Azure DevOps** to open the **Projects page**.

3. Choose the organization, and then select **New Project**.



4. Enter information into the form provided. Provide a name for your project. Your project name can't contain special characters, such as / : \ ~ & % ; @ ' " ? < > | # \$ \* } { , + = [ ] , can't begin with an underscore, can't begin or end with a period, and must be 64 or fewer characters. Enter an optional description. Choose the visibility, **Git** as the source control type, and **Agile** as the work item process. For more information, see [Choosing the right version control for your project](#) and [Choose a process](#).

## Create new project



Project name \*

Fabrikam Test



Description

Test project for testing new features before roll out.

Visibility



Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.



Private

Only people you give access to will be able to view this project.



^ Advanced

Version control ?

Git



Work item process ?

Agile

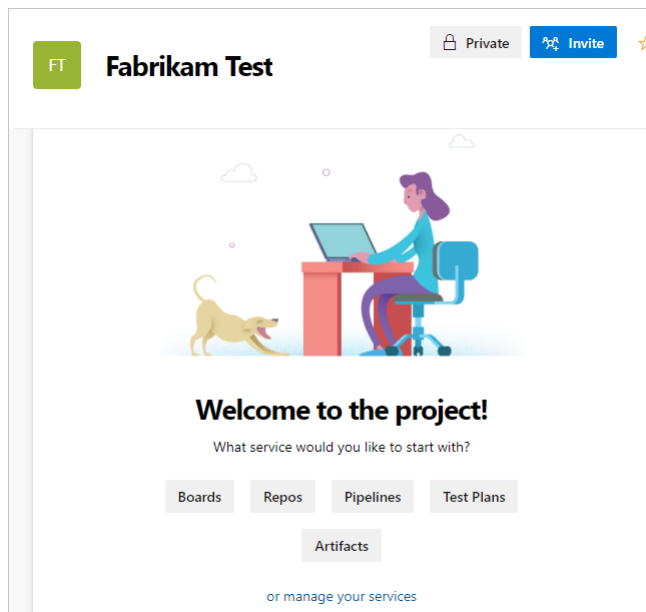


Create

Cancel

Select visibility of either public or private. When you choose public visibility, anyone on the internet can view your project. With private visibility, only people who you give access to can view your project. For more information about public projects, see [Create a public project in your organization](#). If the **Public** option isn't available, you need to change the policy.

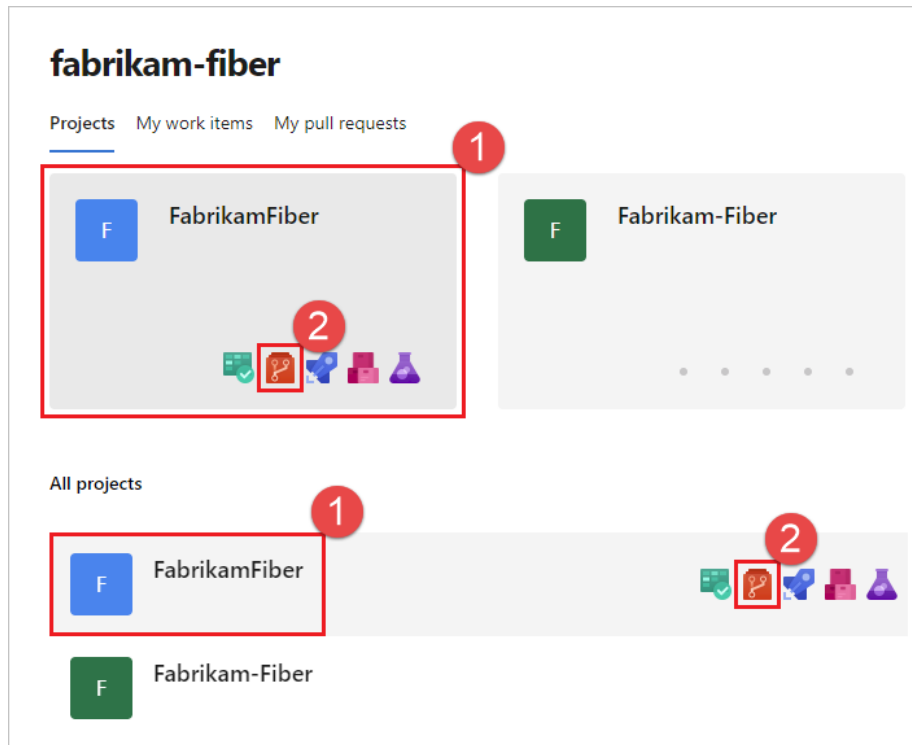
5. Select **Create**. The welcome page appears.



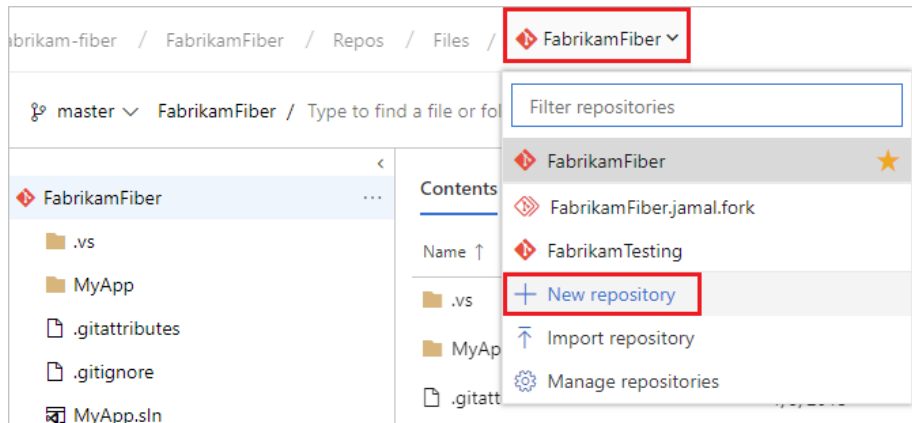
- **Invite:** on the top right if you want to add others to your project. See Add users to a project or team. You can only invite users who are already in your organization.

## 2. Add a Git repo to your project

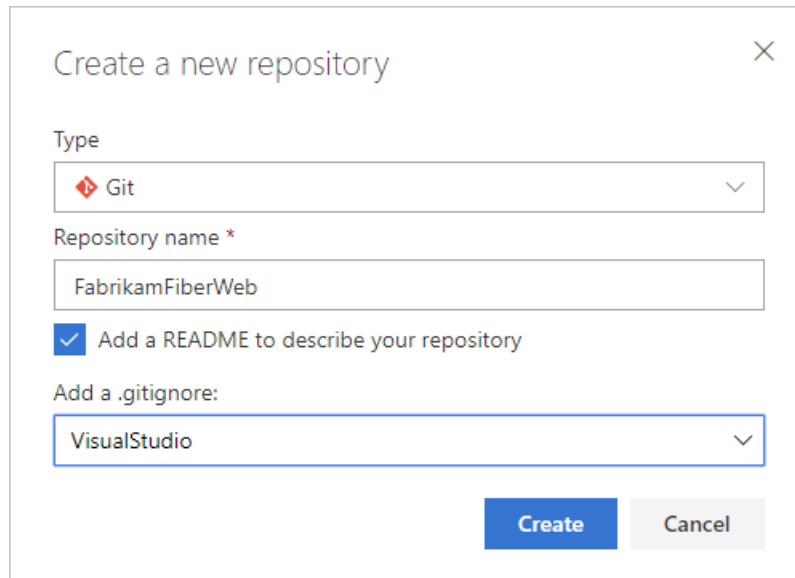
1. Navigate to the **Repos** page in your project, hover your mouse over the name of your project and select the **Repos** icon.



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.



Create a new repository

Type

Git

Repository name \*

FabrikamFiberWeb

☒ Add a README to describe your repository

Add a .gitignore:

VisualStudio

Create Cancel

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

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

### 3. Run a notebook in your workspace

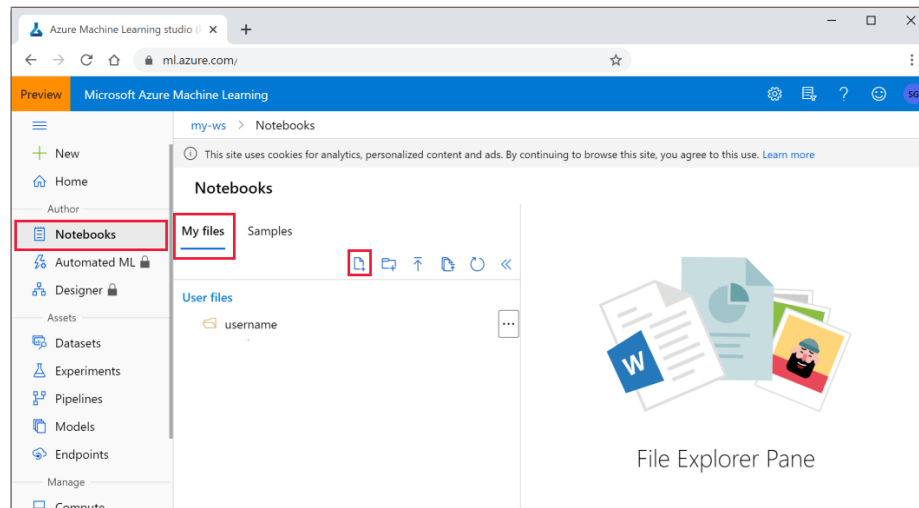
For details on how to create and manage an Azure ML workspace click [here](#).

#### Create notebooks

In your Azure Machine Learning workspace, create a new Jupyter notebook and start working. The newly created notebook is stored in the default workspace storage. This notebook can be shared with anyone with access to the workspace.

To create a new notebook:

1. Open your workspace in Azure Machine Learning studio.
2. On the left side, select **Notebooks**.
3. Select the **Create new file** icon above the list **User files** in the **My files** section.



4. Name the file.
5. For Jupyter Notebook Files, select **Notebook** as the file type.
6. It is recommended that you select your file directory so that others will not make collisions directly on your working branch when pushing your code to **git**.
7. Select **Create**.

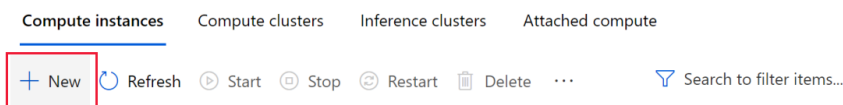
You can create text files as well. Select **Text** as the file type and add the extension to the name (for example, myfile.md or myfile.txt)

You can also upload folders and files, including notebooks, with the tools at the top of the Notebooks page. Notebooks and most text file types display in the preview section. No preview is available for most other file types.

### Create a compute instance

1. In your workspace in Azure Machine Learning studio, select **Compute**, then select **Compute Instance** on the top.
2. If you see a list of compute resources, select **+New** above the list.

#### Compute



3. Fill out the form for your compute instance



New compute instance

Compute name \* ⓘ

Region \* ⓘ

centralus

Virtual machine type \*

CPU (Central Processing Unit)

Virtual machine size \* ⓘ

Standard\_DS3\_v2 4 Cores, 14 GB (RAM), 28 GB (Disk)

☐ Enable SSH access ⓘ

> Advanced settings

Field	Description
Compute name	* Name is required and must be between 3 to 24 characters long.

- Valid characters are upper and lower case letters, digits, and the - character
- Name must start with a letter.
- Name needs to be unique across all existing computes within an Azure region. You will see an alert if the name you choose is not unique.
- If - character is used, then it needs to be followed by at least one letter later in the name. | Virtual machine type | Choose CPU or GPU. thus type cannot be changed after creation. Select CPU for general use unless you know your code will use GPU (example for training deep learning models) | Virtual machine size | Supported virtual machine sizes might be restricted in your region. Check the availability list | Enable/disable SSH access | SSH access is disabled by default. SSH access cannot be changed after creation. Make sure to enable access if you plan to debug interactively with VS Code Remote | Advanced settings | Optional. Configure a virtual

network. Specify the **Resource group**, **Virtual network**, and **Subnet** to create the compute instance inside an Azure Virtual Network (vnet). For more information, see these network requirements for vnet.

4. Select **Create**
5. View the status of the create operation by selecting the compute target from the list

Compute

Compute instances

Compute clusters

Inference clusters

Attached compute

+ New

↺ Refresh

▶ Start

⏸ Stop

↺ Restart

🗑 Delete

⋮

🔍 Search to filter items...

Name	Status	Application URI	Virtual machine size
sdg-compute-instance	Creating		STANDARD_DS3_V2

## Clone Git repositories into your workspace file system

Azure Machine Learning provides a shared file system for all users in the workspace. To clone a Git repository into this file share: \* Follow the instructions here to copy the details needed to clone the Git repo.

- Go back to your Azure ML workspace
- Open a terminal. Once the terminal is opened, you have access to a full Git client and can clone and work with Git via the Git CLI experience.
- It is advised that you clone the repository into your users directory so that others will not make collisions directly on your working branch. To do this type `cd` e.g. `cd david`
- Run `git clone` followed by the followed by the path copied from the Clone URL in the previous section, as shown in the following example. `git clone https://dev.azure.com/fabrikam-fiber/MyFirstProject/_git/`
- Next it will ask for a password, paste the password shown when you cloned your git repo as shown here and press Enter.

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

Switch your directory to the repository that you cloned. For e.g. `cd fabrikam-fiber`

For details on how to work with git such as using `commit` to save changes and using `push` to share code refer to the **command line** examples shown under docuemnts titles **Commits**, **push**, **fetch**, **pull** shown on the left here

## Notes on using a Compute Instance

**Accessing files** Notebooks and R scripts are stored in the default storage account of your workspace in Azure file share. These files are located under your “User files” directory. This storage makes it easy to share notebooks between compute instances. The storage account also keeps your notebooks safely preserved when you stop or delete a compute instance.

The Azure file share account of your workspace is mounted as a drive on the compute instance. This drive is the default working directory for Jupyter, Jupyter Labs, and RStudio. This means that the notebooks and other files you create in Jupyter, JupyterLab, or RStudio are automatically stored on the file share and are accessible from all compute instances in the same workspace.

**Managing a compute instance** Azure RBAC allows you to control which users in the workspace can create, delete, start, stop, restart a compute instance. All users in the workspace contributor and owner role can create, delete, start, stop, and restart compute instances across the workspace. However, only the creator of a specific compute instance, or the user assigned if it was created on their behalf, is allowed to access Jupyter, JupyterLab, and RStudio on that compute instance. A compute instance is dedicated to a single user who has root access, and can terminal in through Jupyter/JupyterLab/RStudio. Compute instance will have single-user log in and all actions will use that user’s identity for Azure RBAC and attribution of experiment runs. SSH access is controlled through public/private key mechanism.

These actions can be controlled by Azure RBAC:

- *Microsoft.MachineLearningServices/workspaces/computes/read*
- *Microsoft.MachineLearningServices/workspaces/computes/write*
- *Microsoft.MachineLearningServices/workspaces/computes/delete*
- *Microsoft.MachineLearningServices/workspaces/computes/start/action*
- *Microsoft.MachineLearningServices/workspaces/computes/stop/action*
- *Microsoft.MachineLearningServices/workspaces/computes/restart/action*

## Collaborate on Jupyter Notebooks by adding contributors

- Navigate to your Azure ML workspace on the Azure portal
- Select **Access control (IAM)** on the left for your machine learning resource
- Click the option **Add** for **Add role assignment** on the right
- Select *Contributor* for **role**
- Leave **Assign access to** as the default
- Search for the person you want to select as a *contributor* in the search bar for **Select**
- Select **Save**

The person has now been added as a contributor and can assess and contribute

to your work. You can, however, collaborate on Jupyter Notebooks