5.2 AZURE DEVOPS INTRODUCTION

Azure DevOps is a cloud-based platform that provides integrated tools for software development teams. It includes everything you need to plan work, collaborate on code, build applications, test functionality, and deploy to production.

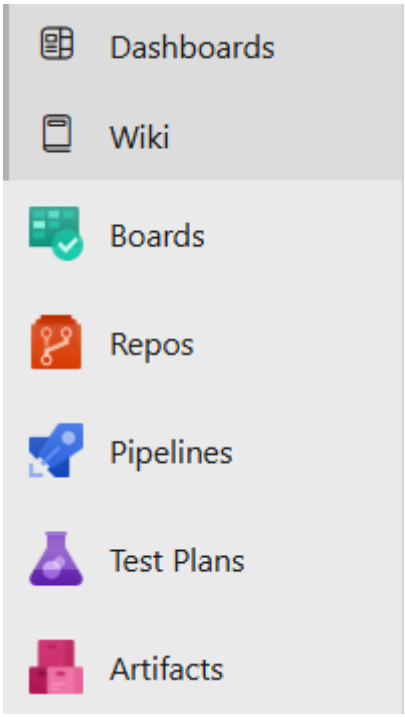
Azure DevOps offers a spectrum of service models to accommodate the unique needs of every team. The free access version helps small teams get started quickly, while the versatile subscription and pay-per-use plans support comprehensive project management.

Key characteristics:

- **End-to-end project management**: Azure DevOps stands as a cohesive suite of services designed to support the complete lifecycle of your software projects. It encompasses everything from initial planning and development, through rigorous testing, to final deployment.
- **Client/server model delivery**: Azure DevOps operates on a client/server model, offering flexibility in how you interact with its services. The web interface provides a convenient way to utilize most services and is compatible with all major browsers. Additionally, certain services like source control, build pipelines, and work tracking offer client-based management options for enhanced control.
- **Flexible and scalable service options**: Azure DevOps caters to teams of all sizes by offering a range of service options. For small teams, many services are complimentary, ensuring that you have access to robust project management tools without any initial investment. For larger teams or more advanced needs, services are accessible through a subscription model or on a pay-per-use basis.

Core services

Azure DevOps includes the following integrated services:



Azure Boards: Plan and track work using Agile tools, Kanban boards, backlogs, and dashboards. Create work items like user stories, bugs, and tasks. Use sprint planning, burndown charts, and velocity tracking. Customize workflows and work item types to match your team's process.

Example scenario: A product team planning a mobile app feature creates user stories for "user sign-in," tracks bugs found during development, and uses sprint boards to monitor progress during two-week iterations.

Azure Repos: Host unlimited private Git repositories or use Team Foundation Version Control (TFVC) for source code management. Features include branch policies, pull requests with code reviews, conflict resolution, and integration with popular IDEs and editors.

Example scenario: Development team members create feature branches for new functionality, submit pull requests for code review, and use branch policies to ensure all code is reviewed and tested before merging to the main branch.

Azure Pipelines: Build, test, and deploy applications with CI/CD pipelines that work with any language, platform, and cloud. Supports Docker containers, Kubernetes, and deployments to Azure, AWS, Google Cloud, or on-premises. Includes parallel jobs, deployment gates, and release approvals.

Example scenario: Every code commit triggers an automated pipeline that builds a .NET web application, runs unit tests, creates a Docker container, and deploys to staging environment for testing before production release.

Azure Test Plans: Plan, execute, and track testing with manual test cases, exploratory testing sessions, and automated test integration. Create test suites, track test results, capture screenshots and videos, and generate detailed test reports.

Example scenario: QA team creates test cases for user registration flow, executes manual tests on different browsers, captures screenshots of issues, and links test results to user stories for traceability.

Azure Artifacts: Create, host, and share packages like NuGet, npm, Maven, Python, and Universal packages with your team and organization. Integrate with build pipelines, manage package versions, and control access with upstream sources and retention policies.

Example scenario: Development team creates a shared authentication library, publishes it as a NuGet package to Azure Artifacts, and references it across multiple projects while controlling access to internal packages.

Typical workflow:

1. **Plan** work items in Azure Boards
2. **Code** features in Azure Repos with pull requests
3. **Build** and package with Azure Pipelines and Azure Artifacts
4. **Test** manually and automatically using Azure Test Plans
5. **Deploy** through Azure Pipelines to various environments
6. **Monitor** progress and metrics via Dashboards
7. **Iterate** based on feedback and new requirements

5.2.1 CREATE AZURE ORGANIZATION

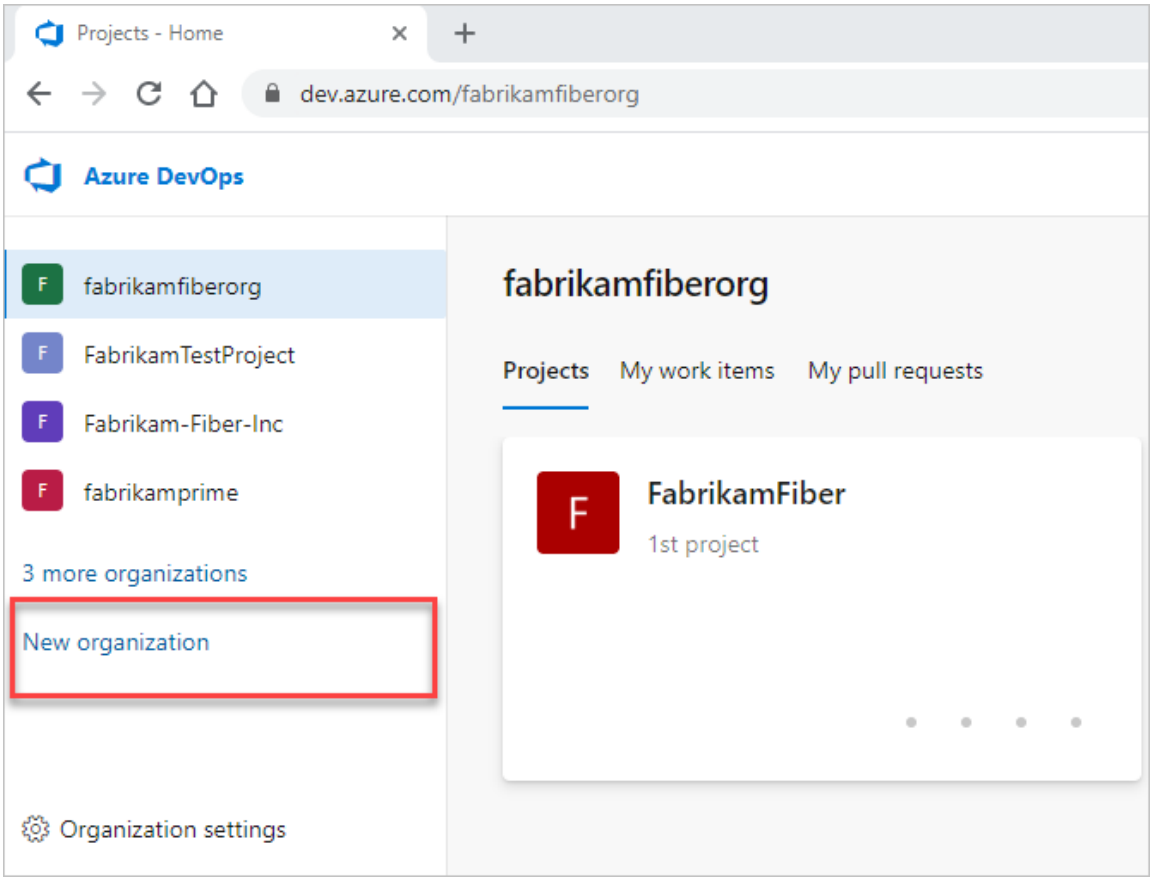
Use an organization to connect groups of related projects and help scale up your enterprise. You can use a personal Microsoft account, GitHub account, or a work or school account. Using your work or school account *automatically connects* your organization to your Microsoft Entra ID.

Prerequisites

Category	Requirements
Authentication	Determine whether you want to use only Microsoft accounts or authenticate users with Microsoft Entra ID..
Planning	Understand how to plan your organizational structure.

Create an organization

1. Sign in to Azure DevOps.
2. Select **New organization**.



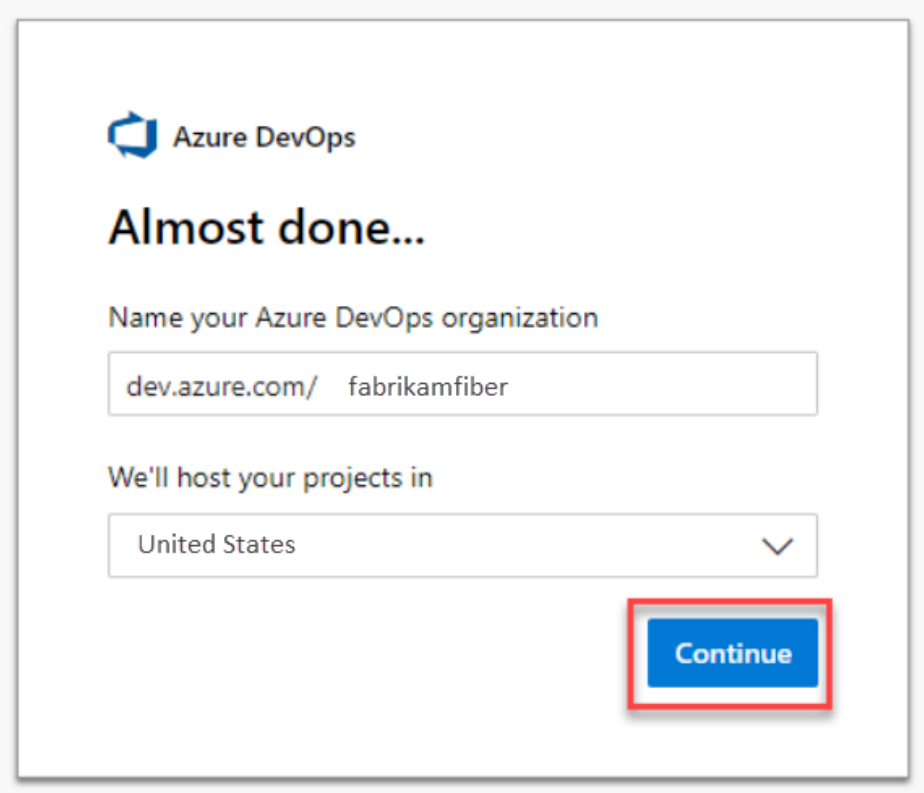
3. Enter the name for your organization, select its hosting geography, and then select **Continue**.

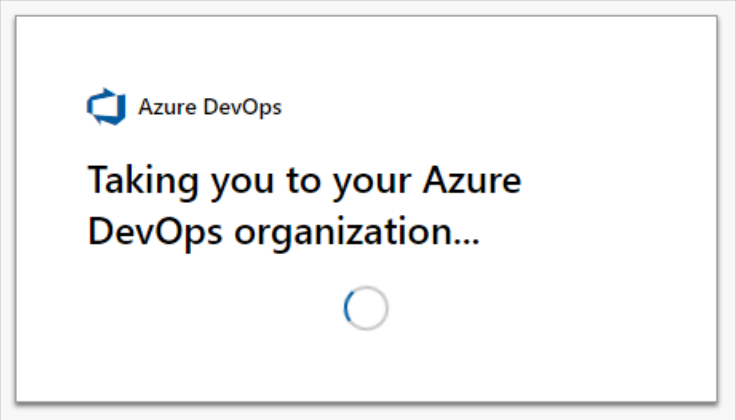
Important

Adhere to the following guidelines when you create an organization name.

- Use only letters from the English alphabet
- Start your organization name with a letter or number
- Use letters, numbers, or hyphens after the initial character
- Ensure that your organization doesn't exceed 50 Unicode characters
- End with a letter or number

If you use any of the disallowed characters, you get the following error message: VS850015: The specified name is not allowed to be used: {Organization name}.





Congratulations, you're an organization owner!

Sign in to your organization at any time, https://dev.azure.com/{yourorganization}.

With your organization, the following aspects are included in the free tier:

- First five users free (Basic license)
- **Azure Pipelines:**
  - One Microsoft-hosted CI/CD (one concurrent job, up to 30 hours per month)
  - One self-hosted CI/CD concurrent job
- **Azure Boards:** Work item tracking and boards
- **Azure Repos:** Unlimited private Git repos
- **Azure Artifacts:** Two GiB free per organization

CREATE A PROJECT

Create an Azure DevOps project to set up a source code repository and organize your work. You can tailor and manage your project to meet your business requirements. Each project isolates its data from other projects.

Prerequisites

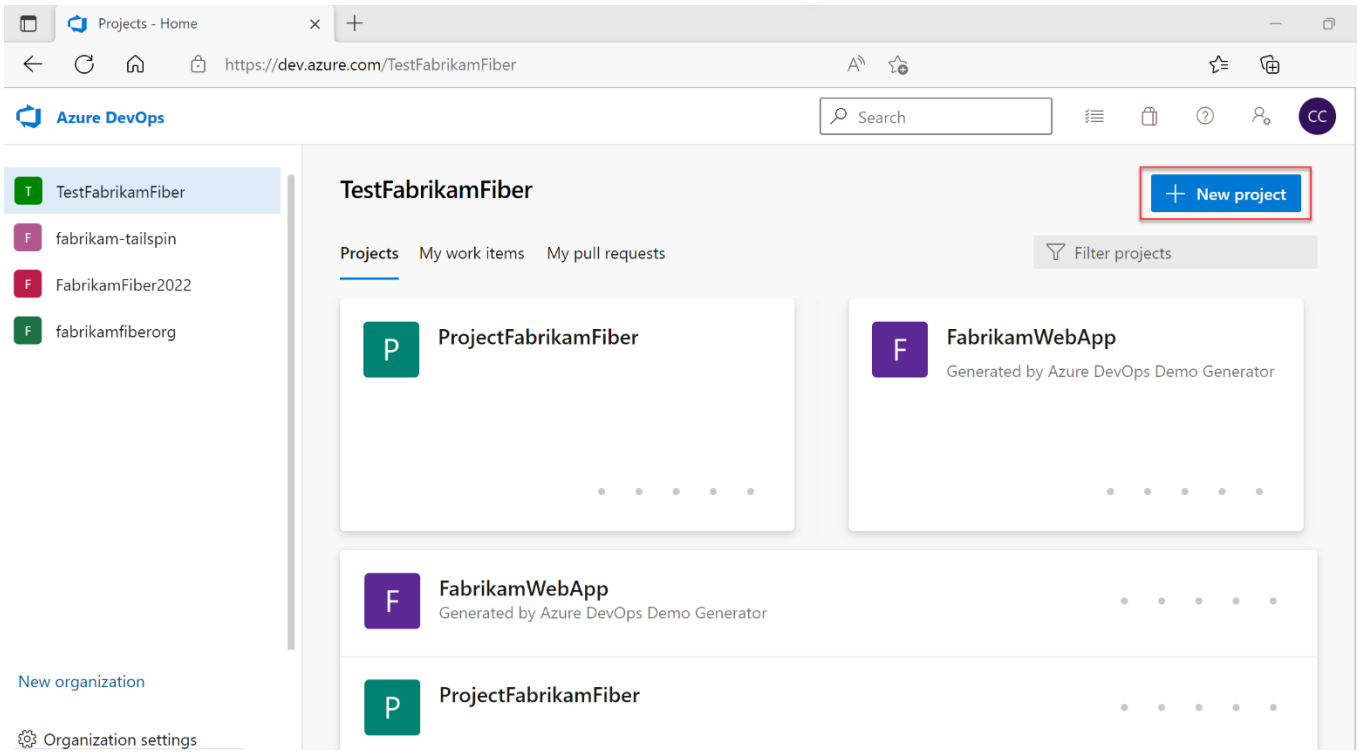
Expand table

Category	Requirements
Organization	An organization.
Permissions	Member of the Project Collection Administrators group or the collection-level "Create new projects" permission set to *Allow*. Organization owners are automatically members of this group.

(i) Create a project

You can create up to 1,000 projects within an organization in Azure DevOps.

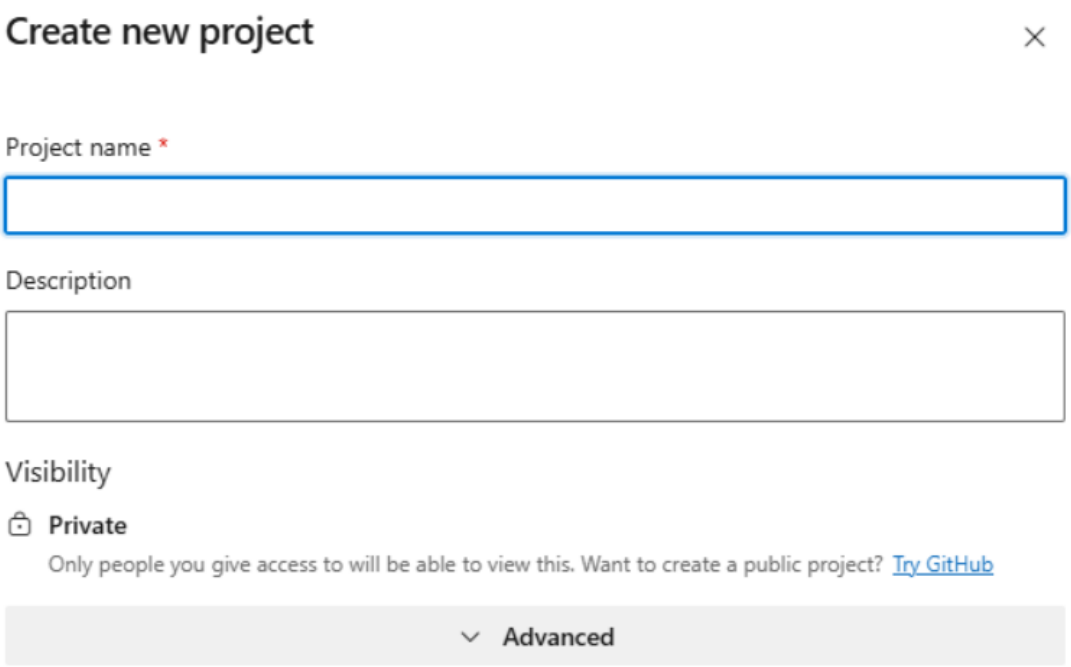
1. Sign in to your organization (https://dev.azure.com/{Your_Organization}).
2. Select **New project**.



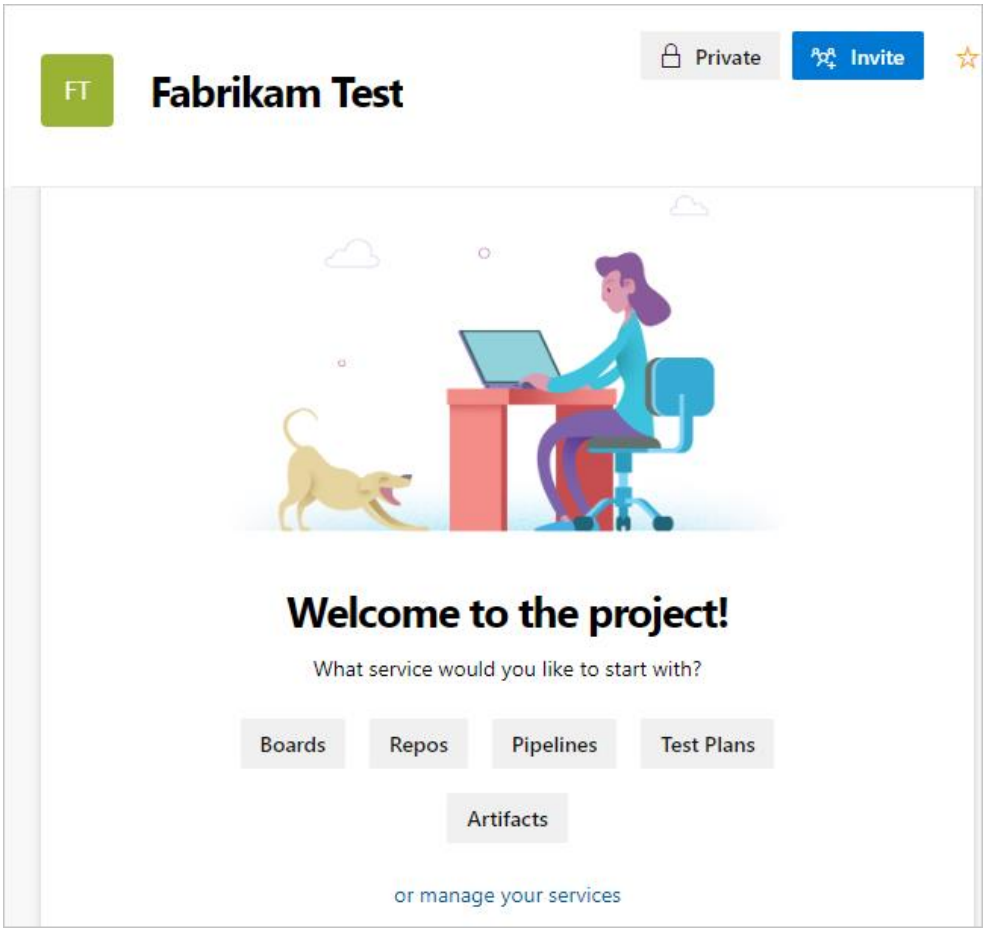
3. Enter information into the form provided.

- 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 initial source control type, and work item process.



4. Select **Create**. Azure DevOps displays the project welcome page.



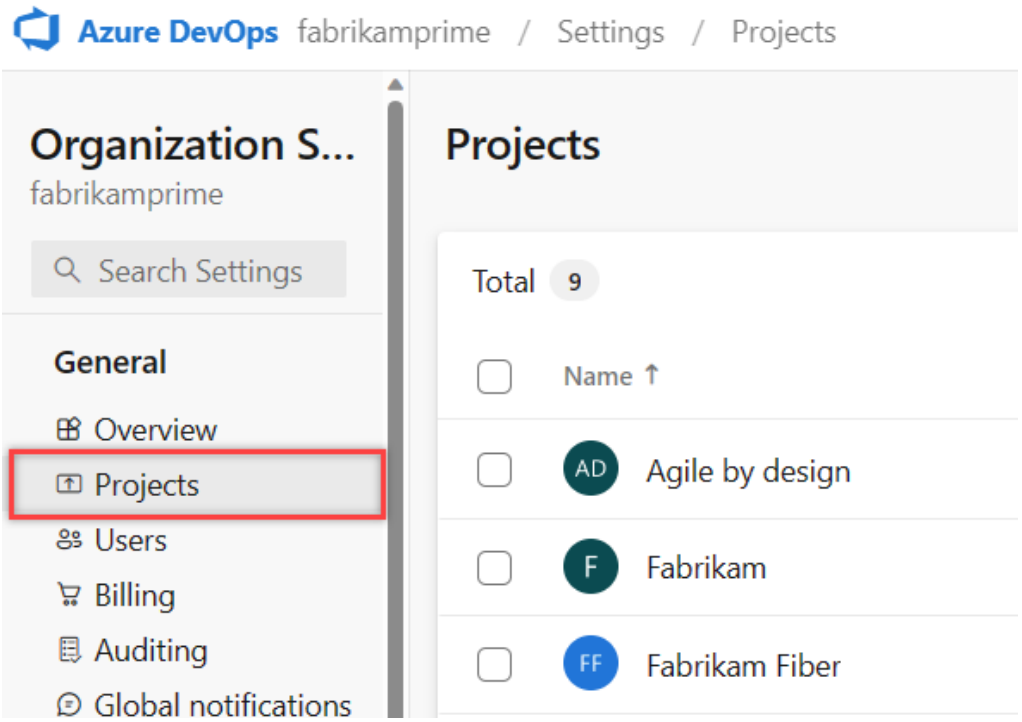
Select one of the following options to continue:

- **Invite**: Add others to your project. See Add users to a project or team. You can only invite users who are already in your organization. For more information, see Add users to a project.
- **Boards**: Add work items. See View and add work items using the Work Items page.
- **Repos**: Clone or import a repository or initialize a *README* file for your project summary page. See Clone an existing Git repo.
- **Pipelines**: Define a pipeline. See Azure Pipelines documentation.
- **Test Plans**: Define test plans and test suites. See Create test plans and test suites.
- **Artifacts**: Discover, install, and publish NuGet, npm, and Maven packages. See the Azure Artifacts overview.
- **Manage your services**: Disable the visibility of services. See Turn a service on or off.

List projects

View a list of projects from your web browser.

1. Sign in to your organization (https://dev.azure.com/{yourorganization}).
2. Select **Organization settings** and then select **Projects**.



Open your desired project.

Add a repository to your project

You can add Git (distributed) or TFVC (centralized) repositories to your project. You can create many Git repositories, but only a single TFVC repository for a project. More steps to address permissions might be required.