

# Contents

[Get started](#)

[Start using Azure DevOps](#)

[What is Azure DevOps?](#)

[Overview of services](#)

[Compare Azure DevOps hosted vs. on-premises](#)

[Get started for end users](#)

[Sign in to the web or a client](#)

[Code with Git](#)

[Set up continuous integration and delivery](#)

[Plan and track work](#)

[Add and run manual tests](#)

[Follow work and pull requests](#)

[Get started as a Stakeholder](#)

[View permissions](#)

[Get started for administrators](#)

[Sign up for Azure DevOps](#)

[Create an organization or project collection](#)

[Manage your project](#)

[Manage your organization or collection](#)

[Add users to a project or team](#)

[Manage teams and configure team tools](#)

[Change individual permissions](#)

[Grant or restrict permissions to select tasks](#)

[Security best practices](#)

[Key concepts](#)

[Plan your organizational structure](#)

[Source control](#)

[Clients and tools](#)

[Software development roles](#)

## Troubleshooting

[Troubleshoot connection](#)

[TF31002: Unable to connect](#)

[Troubleshoot access and permissions](#)

[Allowed address lists and network connections](#)

[Get support or provide feedback](#)

[Look up platform and version](#)

## Reference

[Navigate in Team Explorer](#)

[FAQs](#)

[Service limits](#)

## Resources

[Azure CLI](#)

## Integration overview

[Cross-service integration overview](#)

[GitHub integration](#)

[Deploy to Azure](#)

## Web portal navigation

[Navigation](#)

[Open a service, page, or setting](#)

[Add an artifact or team artifacts](#)

[Use breadcrumbs, selectors, and directories](#)

[Open another project or repo](#)

[Set favorites](#)

[Filter basics](#)

[Search your repo, work items, or wiki](#)

[Manage or enable features](#)

## Search

[Get started with search](#)

[Search code](#)

[Search work items](#)

## Migrate & import

[Migrate data to Azure DevOps Services](#)

- Migrate options
- Import
  - Import large collections
  - Process templates
  - Post-import
  - Troubleshooting
- FAQs, migration and process models
- Permissions & access
  - Permissions and access (Security)
  - About access levels
- Status & security
  - Service status
  - Data protection
  - Data location
  - Credential storage
- IDE Client Resources
  - Visual Studio IDE
  - Visual Studio Code
  - Visual Studio for Mac
- Resources
  - Settings, security, & usage
  - Manage projects
  - Marketplace & extensibility
- DevOps Resource Center
  - What is DevOps?
  - What is Agile?
  - What is Git?



# What is Azure DevOps?

12/13/2022 • 2 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2022](#) - [Azure DevOps Server 2019](#) | [TFS 2018](#)

Azure DevOps supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches.

You can work in the cloud using [Azure DevOps Services](#) or on-premises using Azure DevOps Server. For more information, see [Differences between Azure DevOps Services and Azure DevOps Server](#).

Azure DevOps provides integrated features that you can access through your web browser or IDE client. You can use all the services included with Azure DevOps, or choose just what you need to complement your existing workflows.

STANDALONE SERVICE	DESCRIPTION
<a href="#">Azure Boards</a>	Delivers a suite of Agile tools to support planning and tracking work, code defects, and issues using Kanban and Scrum methods. For more information about Azure Boards, see <a href="#">What is Azure Boards?</a> .
<a href="#">Azure Repos</a>	Provides Git repositories or Team Foundation Version Control (TFVC) for source control of your code. For more information about Azure Repos, see <a href="#">What is Azure Repos?</a> .
<a href="#">Azure Pipelines</a>	Provides build and release services to support continuous integration and delivery of your applications. For more information about Azure Pipelines, see <a href="#">What is Azure Pipelines?</a> .
<a href="#">Azure Test Plans</a>	Provides several tools to test your apps, including manual/exploratory testing and continuous testing. For more information about Azure Test Plans, see <a href="#">Overview of Azure Test Plans</a> .
<a href="#">Azure Artifacts</a>	Allows teams to share packages such as Maven, npm, NuGet, and more from public and private sources and integrate package sharing into your pipelines. For more information about Azure Artifacts, see <a href="#">Overview of Azure Artifacts</a> .

Azure DevOps supports adding extensions and integrating with other popular services, such as: Campfire, Slack, Trello, UserVoice, and more, and developing your own custom extensions.

## Choose Azure DevOps Services

Azure DevOps *Services* supports integration with GitHub.com and GitHub Enterprise Server repositories.

Choose Azure DevOps Services when you want the following outcomes:

- Quick set-up
- Maintenance-free operations
- Easy collaboration across domains

- Elastic scale
- Rock-solid security

Azure DevOps Services also gives you access to cloud build and deployment servers, and application insights. [Start for free](#) and create an organization. Then, either upload your code to share or source control. Begin tracking your work using Scrum, Kanban, or a combination of methods.

For more information, see the [Azure DevOps and GitHub integration overview](#).

## Choose Azure DevOps Server

Azure DevOps *Server* supports integration with GitHub Enterprise Server repositories. Choose on-premises Azure DevOps Server when:

- You need your data to stay within your network.
- Your work tracking customization requirements are met better with the on-premises XML process model over the inheritance process model. The on-premises model supports modification of XML definition files.

When you deploy Azure DevOps Server, you can also configure the following servers or integration points:

- **Build server** supports on-premises and cloud-hosted builds.
- **SQL Server and SQL Analysis Server** support SQL Server Reports and the ability to create Excel pivot charts based on the cube.

Start for free by downloading [Azure DevOps Server Express](#). Then, either upload your code to share or source control. Or, begin tracking your work using Scrum, Kanban, or a combination of methods.

For more information about managing Azure DevOps Server, see the [Administrative tasks quick reference](#).

## Next steps

[Sign up for Azure DevOps Services](#) or [Install Azure DevOps Server](#)

## Related articles

- [A tour of services](#)
- [Data protection overview](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Azure DevOps pricing](#)
- [Azure DevOps and GitHub integration overview](#)

# Overview of services

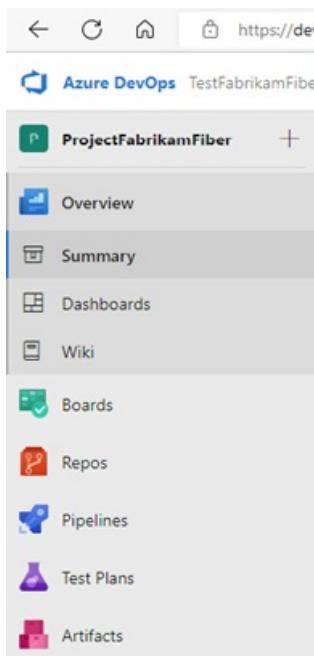
12/13/2022 • 7 minutes to read • [Edit Online](#)

## Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Azure DevOps provides an integrated set of services and tools to manage your software projects, from planning and development through testing and deployment.

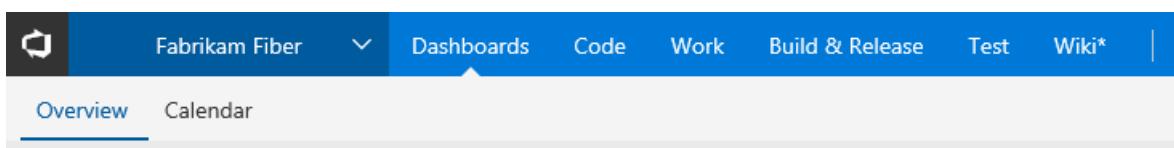
Azure DevOps delivers services through a client/server model. You can use most of the services via the web interface, which you can access from all major browsers. Some services, such as source control, build pipelines, and work tracking, can also be managed through a client.

Access Azure DevOps through the left navigational bar, as shown in the following image. For more information, see the following associated articles.



- [Dashboards](#)
- [Wiki](#)
- [Boards](#)
- [Repos](#)
- [Pipelines](#)
- [Test Plans](#)
- [Artifacts](#)

Access Azure DevOps through the top navigational bar, as shown in the following image. See the associated articles to jump to information for each major service.



- [Dashboards](#)
- [Repos](#)

- Boards
- Pipelines
- Test Plans
- Wiki

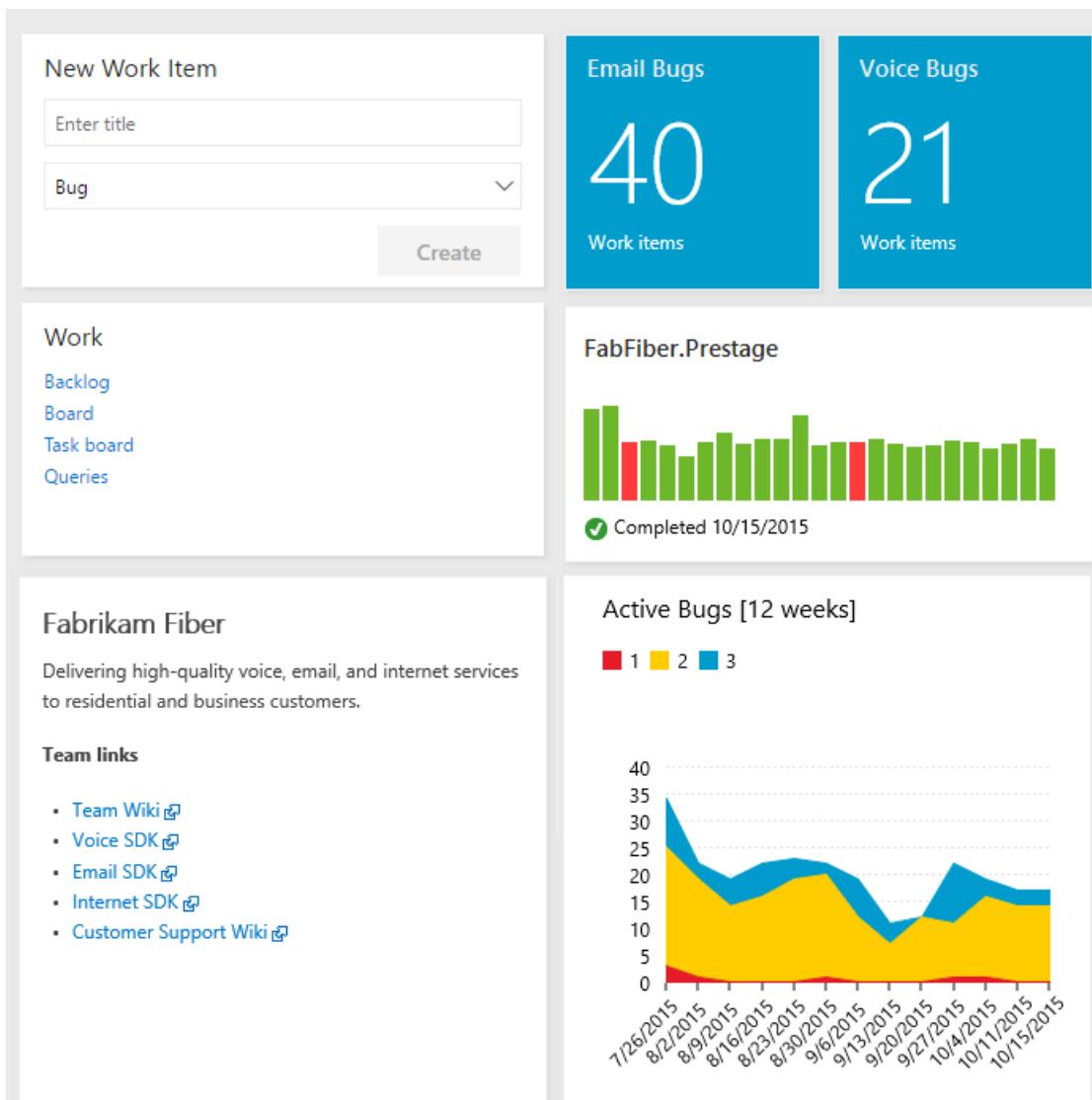
Many of our services are either free for small teams or available through a subscription model or per-use model. You can do a hybrid approach where you use an on-premises deployment to manage your code and work. Then, purchase cloud build or testing services on an as-needed basis.

For more information about client tools, see [Tools and clients that connect to Azure DevOps](#).

## Dashboards

Gain access to user-configurable dashboards from [Dashboards](#).

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' team. The left sidebar has a 'FabrikamFiber' header and links to Overview, Summary, Dashboards (which is selected), Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main area is titled 'FabrikamFiber Team Overview'. It features a 'Welcome' section with a message to get started using Azure DevOps, followed by four cards: 'Manage Work' (Add work to your board), 'Collaborate on code' (Add code to your repository), 'Continuously integrate' (Automate your builds), and 'Visualize progress' (Learn how to add charts). Below this is a 'Team Members' section with icons for CC, a person, and a plus sign. The top navigation bar shows 'Fabrikam / FabrikamFiber / Overview / Dashboards'.



You can do the following tasks in **Dashboards**:

- Add, configure, and manage dashboards
- Configure widgets that you add to dashboards
- Go to different areas of your project quickly

For more information, see [Dashboards](#).

## Repos

Source or version control systems allow developers to collaborate on code and track changes made to the code base. Source control is an essential tool for multi-developer projects.

Our systems support two types of source control: [Git](#) or [Team Foundation Version Control \(TFVC\)](#). You can check in files and organize files within folders, branches, and repositories in both systems.

### Git repos

With Git, each developer has a copy on their dev machine of the source repository, including all branch and history information. Each developer works directly with their own local repository and changes are shared between repositories as a separate step.

Developers commit each set of changes and do version control operations like history and compare without a network connection. When developers need to switch contexts, they create a private local branch, and can switch from one branch to another to pivot among different variations of the codebase. Later, they merge, publish, or dispose of the branch.

## NOTE

Git in Azure DevOps is standard Git. You can use Visual Studio with third-party Git services. You can also use third-party Git clients with Azure DevOps Server.

## TFVC

With TFVC, developers have only one version of each file on their dev machines. Historical data is maintained only on the server. Branches are path-based and created on the server.

### Access Git and TFVC

From **Repos**, you gain access to your source control Git-based or Team Foundation Version Control (TFVC) repositories to support version control of your software projects. These repositories are private.

The screenshot shows the Azure DevOps interface for managing a repository named "DotNetSample". The left sidebar provides navigation links for various project management and development tasks. The "Files" link is currently selected, indicating the focus is on version control. The main content area displays the structure of the "master" branch of the repository. It includes several files and folders, such as ".gitignore", ".vsts-ci.acr.yml", ".vsts-ci.docker.yml", ".vsts-ci.yml", "Dockerfile", "dotnetcore-sample.sln", "LICENSE", "LICENSE-CODE", and "README.md". A sidebar on the right provides a detailed view of the repository's contents, allowing users to search for specific files and view their history.

From **Code**, you gain access to your source control Git-based or TFVC repositories to support version control of your software projects. These repositories are private.

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' repository. The left sidebar has a navigation menu with 'Overview', 'Boards', 'Repos' (selected), 'Files', 'Commits', 'Pushes', 'Branches', 'Tags', and 'Pull requests'. The main area displays the 'master' branch of the 'FabrikamFiber' repository. It shows a file tree with '.gitattributes' and '.gitignore' files under the root, and a folder 'PartsUnlimited-aspnet45' containing three files: '.gitattributes', '.gitignore', and another unnamed file. There are tabs for 'Contents' (selected), 'History', and 'New'. A search bar at the top right says 'Type to find a file or folder...'. The status bar at the bottom indicates 'Last ch'.

From Azure Repos for Git, you can do the following tasks:

- Review, download, and edit files, and review the change history for a file
- Review and manage commits that have been pushed
- Review, create, approve, comment on, and complete pull requests
- Add and manage Git tags

## Boards

Software development projects require ways to easily share information and track the status of work, tasks, issues, or code defects. In the past, you might have used Microsoft Excel, Microsoft Project, a bug tracking system, or a combination of tools. Now, many teams have adopted Agile methods and practices to support planning and development.

From **Boards**, you gain access to Agile tools to support planning and tracking work.

FabrikamFiber Team

Stories backlog

New Active 3/5

New item

Technician can report busy/late on Windows Phone (3)

Technician can see service tickets on Windows Phone (0/2)

Add an information form

Welcome back page

Secure sign in

Unassigned (1)

Add Test | ↗

Test for secure sign in

From **Work**, you gain access to Agile tools to support planning and tracking work.

Backlogs Queries Plans

Stories

Backlog Board Forecast Off Parents Hide In progress items Show Mapping On

New | + | Create query | Column options |

Type	User Story	X	
Title		Add	
Order	State	Story Points	Title
1	New	5	Add an information form
2	New	3	Welcome back page
+ 3	New	8	Interim save on long forms
4	Active	5	> Secure Sign-in
5	Active	5	Canadian addresses don't display

You can do the following tasks with boards.

- Add and update work items
- Define work item queries, and create status and trend charts based on those queries
- Manage your product backlog

- Plan sprints by using sprint backlogs
- Review sprint tasks and update tasks through the task boards
- Visualize the workflow and update the status by using Kanban boards
- Manage portfolios by grouping stories under features and grouping features under epics
- Use task boards during daily Scrum meetings to review work that's completed, remaining, or blocked

Our systems provide several types of work items that you use to track features, requirements, user stories, tasks, bugs, and issues. Each work item is associated with a work item type and a set of fields that can be updated, as progress is made.

For planning purposes, you have access to several types of backlogs and boards to support the main Agile methods—Scrum, Kanban, or Scrumban.

Project managers and developers share information by tracking work items on the backlogs and boards. Useful charts and dashboards complete the picture and help teams monitor progress and trends.

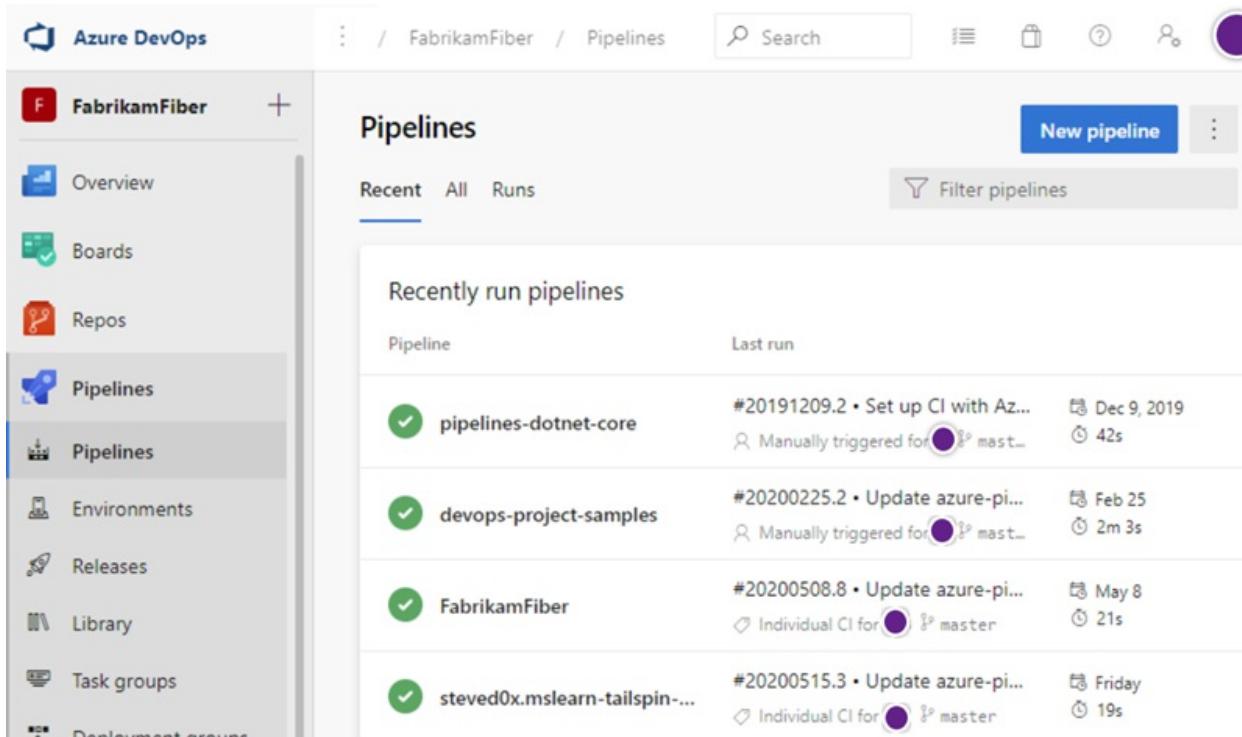
See [Backlogs, boards, and plans](#) for an overview of each.

## Pipelines

The rapid and reliable release of software comes from automating as many processes as possible. Our systems support build, test, and release automation.

- You can define builds to automatically run whenever a team member checks in code changes.
- Your build pipelines can include instructions to run tests after the build runs.
- Release pipelines support managing deployment of your software builds to staging or production environments.

Azure Pipelines provides an integrated set of features to support building and deploying your applications.



The screenshot shows the Azure DevOps Pipelines interface for the 'FabrikamFiber' project. The left sidebar has 'Pipelines' selected. The main area displays a table of recently run pipelines:

Pipeline	Last run
pipelines-dotnet-core	#20191209.2 • Set up CI with Az... Manually triggered for master Dec 9, 2019 42s
devops-project-samples	#20200225.2 • Update azure-pi... Manually triggered for master Feb 25 2m 3s
FabrikamFiber	#20200508.8 • Update azure-pi... Individual CI for master May 8 21s
steved0x.mslearn-tailspin...	#20200515.3 • Update azure-pi... Individual CI for master Friday 19s

Azure Pipelines provides an integrated set of features to support building and deploying your applications.

The screenshot shows the Microsoft DevOps interface for managing build definitions. At the top, there are tabs for Dashboards, Code, Work, Build & Release, Test, and more. Below that, a sub-menu for 'Builds' is selected, along with options for Releases, Library, Task Groups, and Deployment Groups\*. A search bar allows you to find specific build IDs or numbers. Buttons for '+ New' and '+ Import' are available. The main area displays four build definitions:

Definition	Status	Triggered by	7-day pass rate
Content.VS Build : #Content.VS Build_20160609.1	✓ passing	Updated the overview s... 80496e4 in 99 users/...	0% →
Content.VS.PR : #Content.VS.PR_20161019.14	✓ passing	Merge pull request 152... 2be71b1 in 152638	0% →
MSDN.GatedCheck.ALM-master : #20170313.2	✓ passing	Merge pull request 194... 8f7955d in 194899	0% →
MSDN.GatedCheck.VS-master : #20160725.1	✓ passing	Merge pull request 126... 2d56c79 in 126293	0% →

Use pipelines to implement continuous integration and continuous delivery.

- **Build automation:** Define the steps to take during build and the triggers that start a build.
- **Release management:** Manage simultaneous releases. You can also do the following tasks:
  - Configure release pipelines that represent your environments from development to production
  - Run automation to deploy your app to each environment
  - Add approvers to confirm that the app has been successfully deployed in an environment
  - Create your release manually or automatically from a build
  - Track your releases as they're deployed to various environments

For more information, see [Continuous integration on any platform](#).

## Test Plans

Test Plans supports creating and managing manual, exploratory, and continuous tests.

The screenshot shows the Microsoft Test Plans interface. At the top, it says 'Test Plans > FabrikamFiber'. The main area displays a test suite named 'FabrikamFiber (Suite ID: 367)'. The suite contains two active tests:

Outcome	Order	ID	Title
Active	1	368	Fabrikam Test
Active	2	369	Test sign in flow

Test supports creating and managing manual tests.

The screenshot shows the Azure DevOps interface for 'Fabrikam Fiber'. The top navigation bar includes 'Dashboards', 'Code', 'Work', 'Build & Release', 'Test', and 'Wiki\*'. Below this, a secondary navigation bar has 'Test Plans' selected, along with 'Parameters', 'Configurations', 'Runs', 'Machines', and 'Load test'. On the left, a sidebar for 'Fabrikam Fiber: Fabrikam Fiber Team\_Sto...' shows a tree structure with 'Fabrikam Fiber Team\_Stories\_Fabrikam Fiber' expanded, and '379 : Phone sign in (2)' selected. The main area displays a 'Test suite: 379 : Phone sign in (Suite ID: 477)'. A grid titled 'Tests' lists two entries: 'Active' (ID 474, Configuration Windows 8) and 'Active' (ID 478, Configuration Windows 8). The grid has columns for 'Outcome', 'Order', 'ID ↑', 'Title', and 'Configuration'.

With test features, you gain access to the following features:

- Customization of workflows with test plan, test suite, and test case work items
- End-to-end traceability from requirements to test cases and bugs with requirement-based test suites
- Criteria-based test selection with query-based test suites
- Excel-like interface with the grid for easy creation of test cases
- Reusable test steps and test data with shared steps and shared parameters
- Sharable test plans, test suites, and test cases for reviewing with Stakeholders
- Browser-based test execution on any platform
- Real-time charts for tracking test activity

For more information, see [Azure Test Plans documentation](#).

## Collaboration services

Azure DevOps also provides the following collaboration services.

- [Team dashboards](#)
- [Project wiki](#)
- [Discussion within work item forms](#)
- Linking of [work items](#), [commits](#), [pull requests](#), and other artifacts to support traceability
- [Alerts and change notifications](#) managed per user, team, project, or organization
- Ability to [request](#), [provide](#), and manage feedback
- [Analytics service](#), [analytics views](#), and [Power BI reporting](#)
- [Dashboards](#)
- [Project wiki](#)
- [Discussion within work item forms](#)
- Linking of [work items](#), [commits](#), [pull requests](#), and other artifacts to support traceability
- [Alerts and change notifications](#) managed per user, team, project, or collection
- Ability to [request](#), [provide](#), and manage feedback
- [SQL Server Reporting](#)

## Service hooks

With service hooks, you can complete tasks on other services when events happen within your project hosted on Azure DevOps. For example, you can send a push notification to your team's mobile devices when a build fails. You can also use service hooks in custom apps and services as a more efficient way to drive activities in your projects.

The following services are available as the target of service hooks. For more information about other apps and services that integrate with Azure DevOps, visit the [Visual Studio Marketplace](#).

For the latest set of supported services, see [Integrate with service hooks](#).

## Azure cloud-hosted services

Azure provides cloud-hosted services to support application development and deployment. You can make use of these services solely or in combination with Azure DevOps.

To browse the directory of integrated services, features, and bundled suites, see [Azure products](#).

For continuous delivery to Azure from Azure DevOps, see [Automatically build and deploy to Azure web apps or cloud services](#).

## Administrative services

There are features and tasks associated with administering a collaborative software development environment. You can complete most of these tasks through the web portal. For more information, see [About user, team, project, and organization-level settings](#).

## Related articles

- [Azure DevOps Services vs. Azure DevOps Server](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Azure DevOps pricing](#)

# Compare Azure DevOps Services with Azure DevOps Server

12/13/2022 • 9 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The **cloud offering**, Azure DevOps Services, provides a scalable, reliable, and globally available hosted service. It's backed by a 99.9% SLA, monitored by our 24/7 operations team, and available in local data centers around the world.

The **on-premises offering**, Azure DevOps Server, is built on a SQL Server back end. Customers usually choose the on-premises version when they need their data to stay within their network. Or, when they want access to SQL Server reporting services that integrate with Azure DevOps Server data and tools.

Both offerings provide the same [essential features and services](#), but Azure DevOps Services offers the following added benefits:

- Simplified server management
- Immediate access to the latest and greatest features
- Improved connectivity with remote sites
- A transition from capital expenditures (servers and the like) to operational expenditures (subscriptions)

To determine which offering—cloud or on-premises—meets your needs, consider the following key differences.

## Key differences between Azure DevOps Services and Azure DevOps Server

When you're choosing which platform you want, or if you're considering a move from on-premises to the cloud, consider the following areas:

- [Data scope and scalability](#)
- [Authentication](#)
- [Users and groups](#)
- [User access management](#)
- [Security and data protection](#)

### Differences in specific feature areas

Although Azure DevOps Services is a hosted version of Azure DevOps Server, there are some differences between features. Some Azure DevOps Server features aren't supported in Azure DevOps Services. For example, Azure DevOps Services doesn't support integration with SQL Server Analysis Services to support reporting.

### Differences in support

- [Process customization](#)
- [Reporting](#)

If you're using Azure DevOps Server and considering a move to Azure DevOps Services, understand your [migration options](#).

## Scope and scale data

As your business grows, you may need to scale up your Azure DevOps instance.

## Azure DevOps Services

Azure DevOps Services offer two options for scoping and scaling data: organizations and projects.

Organizations in Azure DevOps Services get their own URLs (for example, <https://dev.azure.com/fabrikamfiber>), and they always have exactly one project collection. Organizations can have many projects within a collection.

We recommend that you create organizations in Azure DevOps Services wherever you would create collections in Azure DevOps Server. The following scenarios apply:

- You can purchase Azure DevOps Services users per organization - Paid users can access only the organization in which the payment is made. If you have users who need access to many organizations, Visual Studio subscriptions can be an attractive option. Visual Studio subscribers can be added to any number of organizations at no charge. We're also considering other ways to make access available to many organizations that are grouped into a single organization.
- You currently have to administer organizations one at a time. This process can be cumbersome when you have many organizations.

For more information, see [Plan your organizational structure in Azure DevOps](#).

## Azure DevOps Server

Azure DevOps Server offers the following three options for scoping and scaling data: deployments, project collections, and projects. In the simplest case, deployments are just servers.

Deployments can be more complicated, however, which could include:

- Two-server deployment where SQL is split out on a separate machine
- High-availability farms with lots of servers

Project collections serve as containers for security and administration, and physical database boundaries. They're also used to group related projects.

Finally, projects are used to encapsulate the assets of individual software projects, including source code, work items, and so on. For more information, see [Plan your organizational structure in Azure DevOps](#).

# Authentication

## Azure DevOps Services

With Azure DevOps Services, you connect over the public internet (for example, <https://contoso.visualstudio.com>). You either authenticate with [Microsoft account](#) credentials or with [Azure AD](#) credentials, depending on your organization setup. You can also set up Azure AD to require features such as multi-factor-authentication, IP address restrictions, and so on.

We recommend that you configure your organizations to use Azure AD rather than Microsoft accounts. This method provides a better experience in many scenarios and more options for enhanced security.

For more information, see [About accessing Azure DevOps Services with Azure AD](#).

## Azure DevOps Server

With Azure DevOps Server, you connect to an intranet server (for example, <https://tfs.corp.contoso.com:8080/tfs>). You authenticate with Windows Authentication and your Active Directory (AD) domain credentials. This process is transparent and you never see any kind of sign-in experience.

# Manage users and groups

## Azure DevOps Services

In Azure DevOps Services, you can use a similar mechanism to [provide access to groups of users](#). You can add Azure AD groups to Azure DevOps Services groups. If you use Microsoft Accounts instead of Azure AD, you have to [add users](#) one at a time.

## Azure DevOps Server

In Azure DevOps Server, you provide users access to deployments by adding Active Directory (AD) groups to various Azure DevOps groups (for example, the Contributors group for an individual project). The AD group memberships are kept in sync. As users are added and removed in AD, they also gain and lose access to Azure DevOps Server.

# Manage user access

In both Azure DevOps Services and Azure DevOps Server, you manage access to features by assigning users to an [access level](#). All users must be assigned to a single access level. In both the cloud and on-premises offerings, you can give free access to work item features to an unlimited number of Stakeholders. Also, an unlimited number of Visual Studio subscribers can have access to all Basic features at no extra charge. You pay only for other users who need access.

## Azure DevOps Services

In Azure DevOps Services, you must [assign an access level](#) to each user in your organization. Azure DevOps Services validates Visual Studio subscribers as they sign in. You can assign Basic access for free to five users without Visual Studio subscriptions.

To give Basic access or higher to more users, [set up billing](#) for your organization and [pay for more users](#). Otherwise, all other users get Stakeholder access.

Azure AD groups give access to groups of users. Access levels are automatically assigned at first sign-in. For organizations that are configured to use Microsoft accounts for signing in, you must assign access levels to each user explicitly.

## Azure DevOps Server

In Azure DevOps Server, all use is on the honor system. To set access levels for users based on their licenses, specify their [access levels](#) on the administration page. For example, assign unlicensed users Stakeholder access only.

Users with an Azure DevOps Server Client Access License (CAL) can have Basic access. Visual Studio subscribers can have either Basic or Advanced access, depending on their subscriptions. Azure DevOps Server doesn't attempt to verify these licenses or enforce compliance.

# Security and data protection

Many entities want to know more about data protection when they consider moving to the cloud. We're committed to ensuring that Azure DevOps Services projects stay safe and secure. We have technical features and business processes in place to deliver on this commitment. You can also take steps to secure your data. Learn more in our [Data Protection overview](#).

# Process customization

You can customize the work-tracking experience in different ways, depending on the supported process model:

## Azure DevOps Services

Azure DevOps Services uses the **Inheritance** process model, which supports WYSIWYG customization.

## Azure DevOps Server

With Azure DevOps Server, you can choose the **Inheritance** process model or the **On-premises XML** process model, which supports customization through import or export of XML definition files for work-tracking objects. Azure DevOps Server 2018 and earlier versions only has access to the **On-premises XML** process model. Although the **On-premises XML** process model option is powerful, it can cause various issues. The main issue is that processes for existing projects aren't automatically updated.

To help you avoid these issues in Azure DevOps Services, custom process templates and the **witadmin.exe** tool have always been disabled. This approach has enabled us to automatically update all projects with each Azure DevOps Services upgrade. Meanwhile, the product team is working hard to make customizing processes possible in ways that we can support easily and continuously. We recently introduced the first of these changes and more changes are on the way.

With the new process-customization capability, you can make changes directly within the web user interface (UI). If you want to customize your processes programmatically, you can do so through REST endpoints. When you customize projects this way, they're automatically updated when we release new versions of their base processes with Azure DevOps Services upgrades.

For more information, see [Customize your work-tracking experience](#).

## Analytics and reporting

Azure DevOps Services and Azure DevOps Server offer the following tools that give you insight into the progress and quality of your software projects:

### Azure DevOps Server 2019 to Azure DevOps Services

- **Dashboards** and lightweight **charts** that are available in both the cloud and on-premises platforms. These tools are easy to set up and use.
- **The Analytics service** and **Analytics widgets**. The Analytics service is optimized for fast read-access and server-based aggregations.
- **Microsoft Power BI integration**, which supports getting Analytics data into Power BI reports and provides a combination of simplicity and power.
- **OData support**, which allows you to directly query the Analytics service from a supported browser, and then use the returned JSON data as you want. You can generate queries that span many projects or your entire organization. To learn more about the Analytics service, see our [Reporting roadmap](#).

### Azure DevOps Server 2018

- **Dashboards** and lightweight **charts** that are available in both the cloud and on-premises platforms. These tools are easy to set up and use.
- **SQL Server Reporting Services (SSRS) reports** are available when Azure DevOps Server is configured with SQL Server Analysis Services.

## Visual Studio Team Services is now Azure DevOps Services

Many of the featured services in VSTS are now offered as standalone services in both Azure DevOps Services and Azure DevOps Server 2019 and up. You can get services separately or all together as Azure DevOps Services. If you're an Azure DevOps subscriber, you have access to all of the services already.

VSTS FEATURE NAME	AZURE DEVOPS SERVICE NAME	DESCRIPTION
Build & release	Azure Pipelines	Continuous integration and continuous delivery (CI/CD) that works with any language, platform, and cloud.
Code	Azure Repos	Unlimited cloud-hosted private Git and Team Foundation Version Control (TFVC) repositories for your project.
Work	Azure Boards	Work tracking with Kanban boards, backlogs, team dashboards, and custom reporting.
Test	Azure Test Plans	All-in-one planned and exploratory testing solution.
Packages (extension)	Azure Artifacts	Maven, npm, Python, Universal Package, and NuGet package feeds from public and private sources.

Azure DevOps Services and Azure DevOps Server 2019 and up use the new navigation user interface, with a vertical sidebar to go to the main service areas: **Boards**, **Repos**, **Pipelines**, **Artifacts**, **Test Plans**, and more. For more information, see [Web portal navigation in Azure DevOps](#).

#### NOTE

You can disable select services from the user interface. For more information, see [Turn a service on or off](#).

You can still use `visualstudio.com` to access Azure DevOps Services. We've moved to the new `dev.azure.com` domain name as the primary URL for new organizations. That URL is

`https://dev.azure.com/{your organization}/{your project}`. If you want to change your URL to be based on `dev.azure.com` as the primary, an organization administrator can do so from the organization settings page.

## Related articles

- [Essential services](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Pricing for Azure DevOps Services](#)
- [Pricing for Azure DevOps Server](#)

# Connect to a project in Azure DevOps

12/13/2022 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Learn how to connect to a project to share code, build apps, track work, and collaborate with team members. You can use any of the following clients:

- [Web portal](#)
- [Visual Studio or Team Explorer](#)
- [Eclipse/Team Explorer Everywhere](#)
- [Android Studio with the Azure DevOps Services Plugin for Android Studio](#)
- [IntelliJ with the Azure DevOps Services Plugin for IntelliJ](#)
- [Visual Studio Code](#)

A project defines a process and data storage in which you manage your software projects from planning to deployment. When you connect to a project, you connect to an organization or project collection. One or more projects may be defined within a collection. There must be at least one project. For more information, see [About projects and scaling your organization](#).

## Prerequisites

- If you don't have a project yet, [create one](#).
- If you need to add a team, see [Add teams](#). If you don't have access to the project, [get invited to the team](#).
- From each of these clients, you can switch context to a different project and connect as a different user. If you work remotely, configure your client to [connect to an Azure DevOps Proxy Server](#).
- To get started with a code base, [set up Git](#) or [set up TFVC](#).

## Connect from the web portal

1. If you're not a member of a security group, ask your Project Administrator to add you.
2. Open a browser and enter a URL that uses the following form:

```
https://dev.azure.com/OrganizationName/ProjectName
```

```
http://ServerName/DefaultCollection/ProjectName
```

For example, to connect to the server named **FabrikamPrime**, type:  
**http://FabrikamPrime/DefaultCollection**.

```
http://ServerName:8080/tfs/DefaultCollection/ProjectName
```

For example, to connect to the server named **FabrikamPrime**, type:  
**http://FabrikamPrime:8080/tfs/DefaultCollection**.

The default Port is 8080. If you don't use default values, specify the port number and directory for your server.

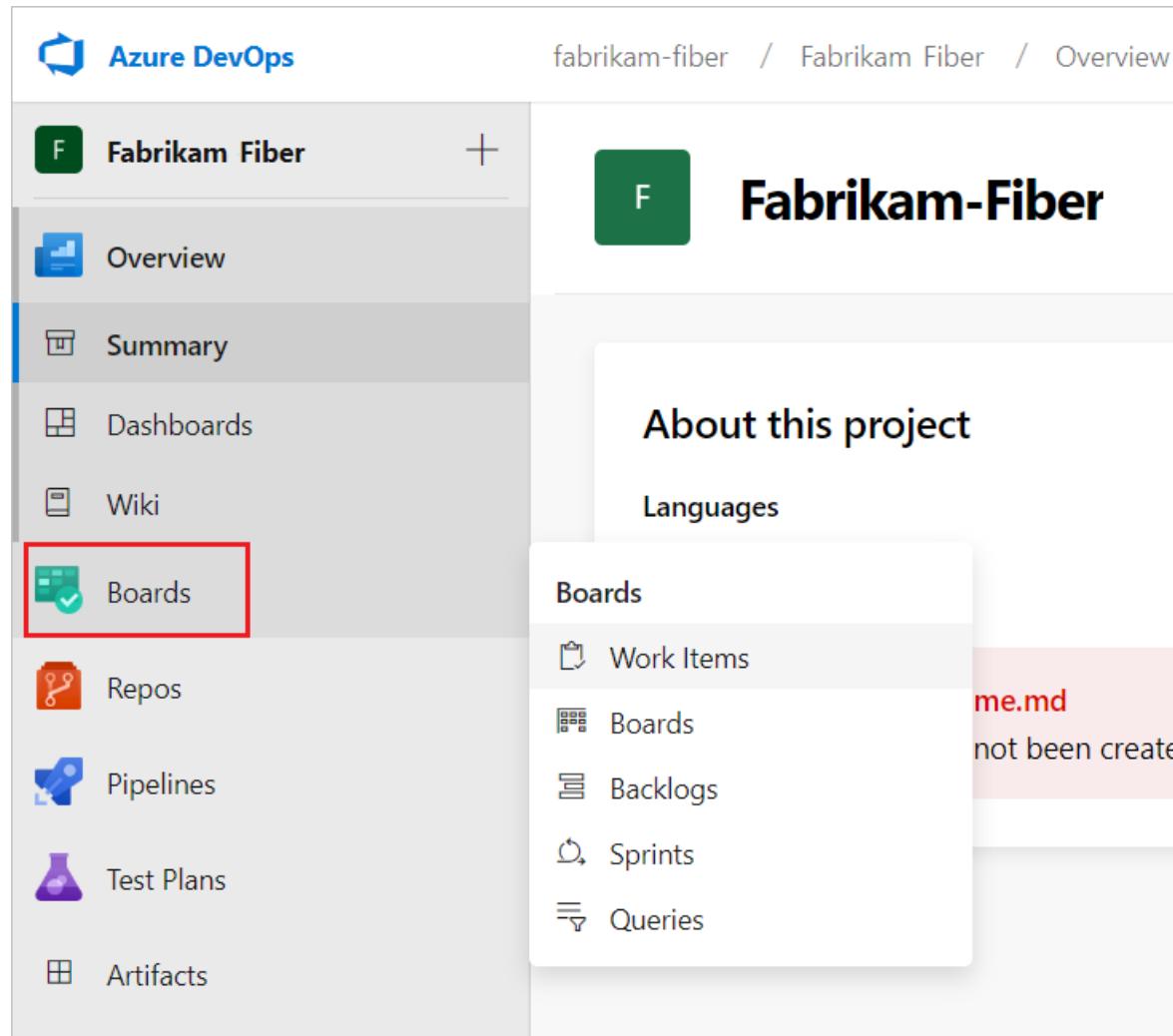
- When you access the server for the first time, a Windows Identity dialog box appears. Enter your credentials and choose OK.

**TIP**

If you select **Remember me**, you won't have to enter your credentials the next time you connect.

- Choose your project, team, or page of interest.

From the project summary page, hover over a service and then choose the page you want. To choose another project, choose **Azure DevOps**. 



The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. On the left, there's a sidebar with various navigation links: Overview, Summary, Dashboards, Wiki, Boards (which is highlighted with a red box), Repos, Pipelines, Test Plans, and Artifacts. The main content area is titled 'Fabrikam-Fiber' and displays the 'About this project' section. Below it, there's a 'Languages' section and a 'Boards' dropdown menu. The 'Boards' menu contains five items: Work Items, Boards, Backlogs, Sprints, and Queries. The 'Boards' item is currently selected, as indicated by its bolded text and the fact that it's the only item with a red background.

From the project summary page, hover over a service and then choose the page you want. To choose another project, choose the  Azure DevOps logo.

Fabrikam Fiber star  
Web, voice, and phone apps

Add tags

New work item >

Fabrikam Fiber / README.md

minor modification to test development section in mobile form

Update this README.md file.

A README.md file is intended to quickly orient readers to what your project can do.  
Learn more about Markdown.

[page 1](#)  
[page 2](#)  
[page 3 - verifying this works as advertised](#)

To learn more about each page and the tasks you can do, see [Web portal navigation](#).

### Sign in with different credentials

1. Open your profile menu and choose **Sign out**.

Search

Jamal Hartnett  
fabrikamfiber4@hotmail.com

My profile

Security

Usage

Notification settings

Preview features

Theme

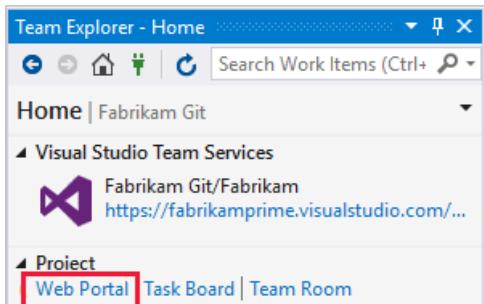
Help >

Sign out

2. Choose **Sign in** and enter your credentials.

### Open the web portal from Team Explorer

Open the web portal from the home page.



## Connect from Visual Studio or Team Explorer

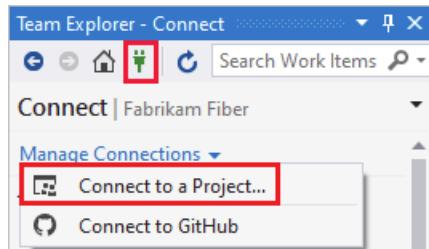
If you haven't already, [download and install a version of Visual Studio](#).

If you're not a member of an Azure DevOps security group, [get added to one](#). Check with a team member. You'll need the names of the server, project collection, and project to connect to.

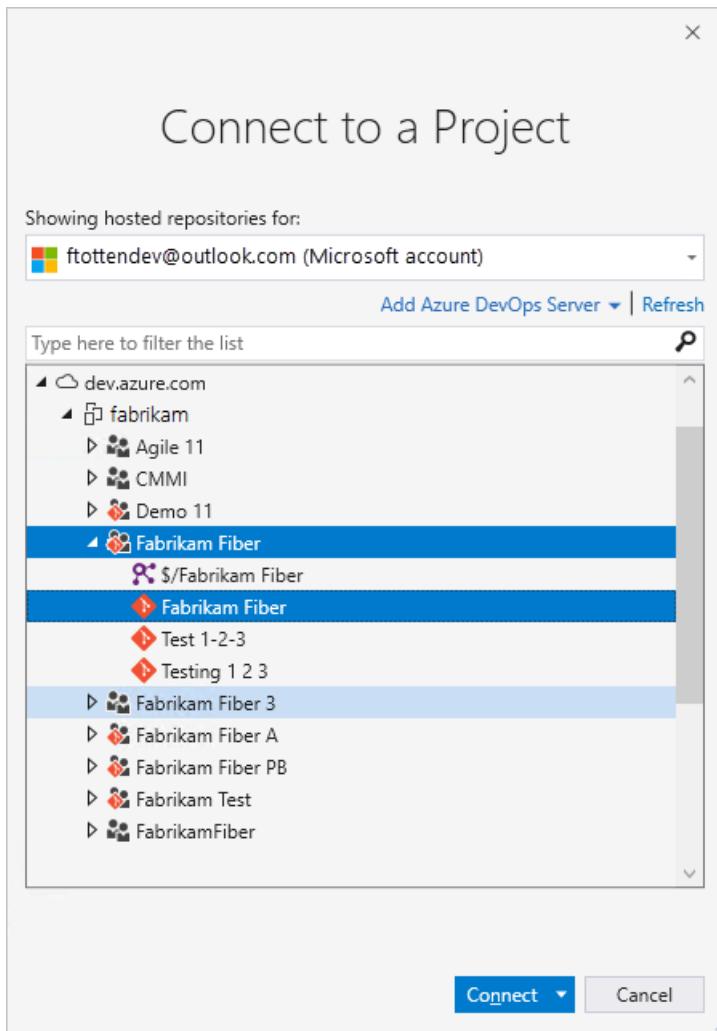
- [Visual Studio 2019](#)
- [Visual Studio 2017](#)
- [Visual Studio 2015](#)

### Visual Studio 2019

1. Select the **Manage Connections** button in Team Explorer to open the **Connect** page. Choose **Connect to a Project** to select a project to connect to.



**Connect to a Project** shows the projects you can connect to, along with the repos in those projects.



2. Select **Add Azure DevOps Server** to connect to a project in Azure DevOps Services. Enter the URL to your server and select Add.



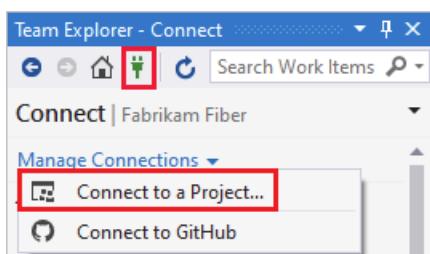
3. Select a project from the list and select Connect.

#### Change sign-in credentials

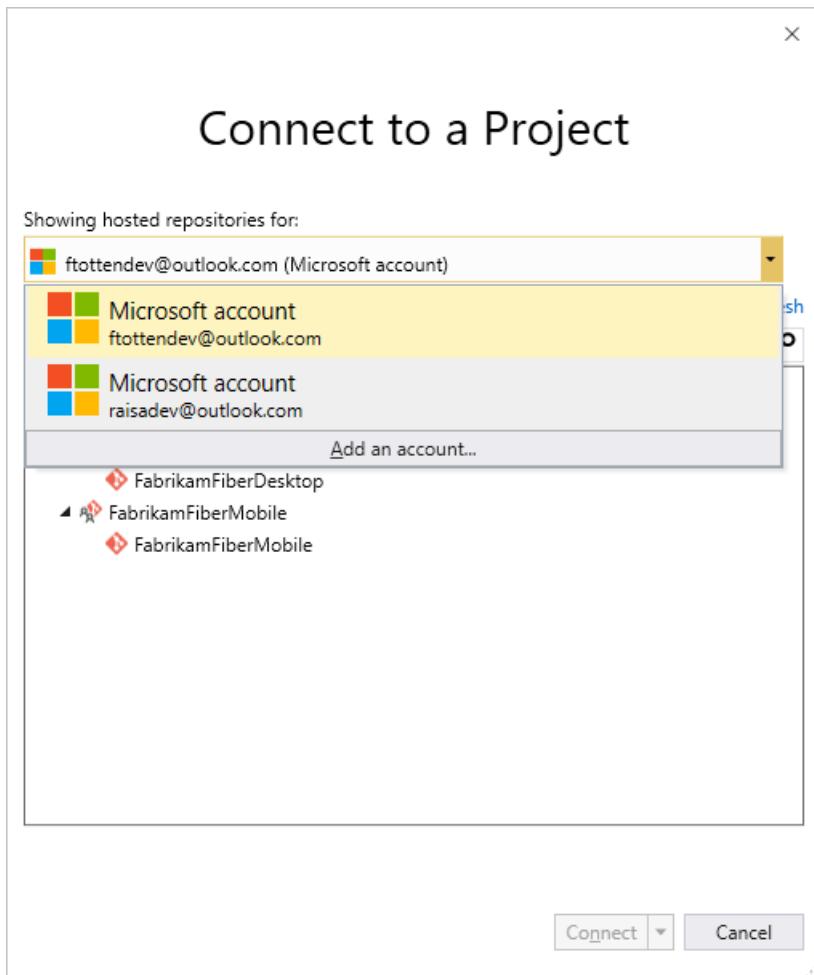
- [Visual Studio 2019](#)
- [Visual Studio 2017](#)
- [Visual Studio 2015](#)

#### Visual Studio 2019

1. From **Connect**, choose the **Connect to a Project** link to sign in with different credentials.



2. Select a different user or select **Add an account** to access a project using different credentials.

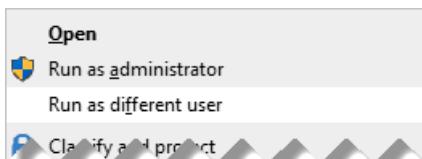


3. Sign in using an account that is associated with an Azure DevOps project, either a valid Microsoft account or GitHub account.

#### Use different Visual Studio credentials

You can run Visual Studio with credentials different from your current Windows user account. Find `devenv.exe` under the *Program Files (86)* folder for your version of Visual Studio.

Select Shift and right-click `devenv.exe`, then select **Run as different user**.



#### User accounts and licensing for Visual Studio

To connect to a project, you need your user account added to the project. The **Organization owner** for Azure DevOps Services or a member of the **Project Administrators** group usually adds user accounts. To learn more, see [Add organization users and manage access](#) or [Add or remove users or groups, manage security groups](#).

Azure DevOps Services provides access to the first five account users free. After that, you need to [pay for more users](#).

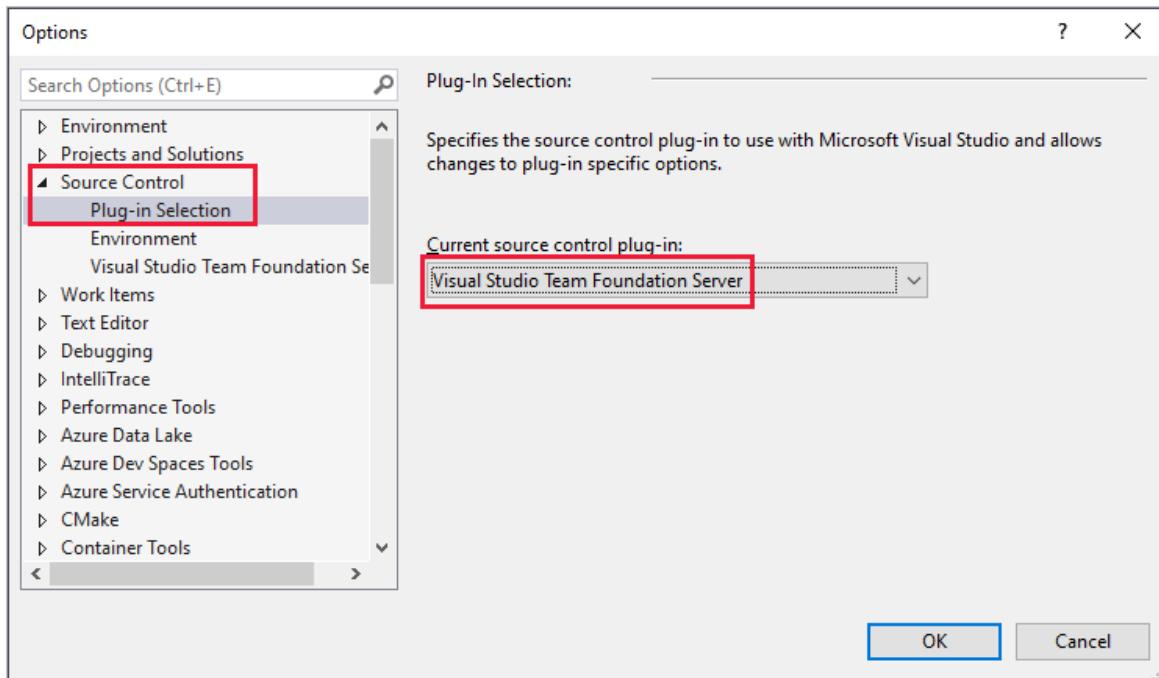
For on-premises TFS, each user account must have a TFS client access license (CAL). All Visual Studio subscriptions and paid Azure DevOps Services users include a TFS CAL. Find out more about licensing from the [Team Foundation Server pricing page](#).

You can also provide access to Stakeholders in your organization who have limited access to select features as described in [Work as a Stakeholder](#).

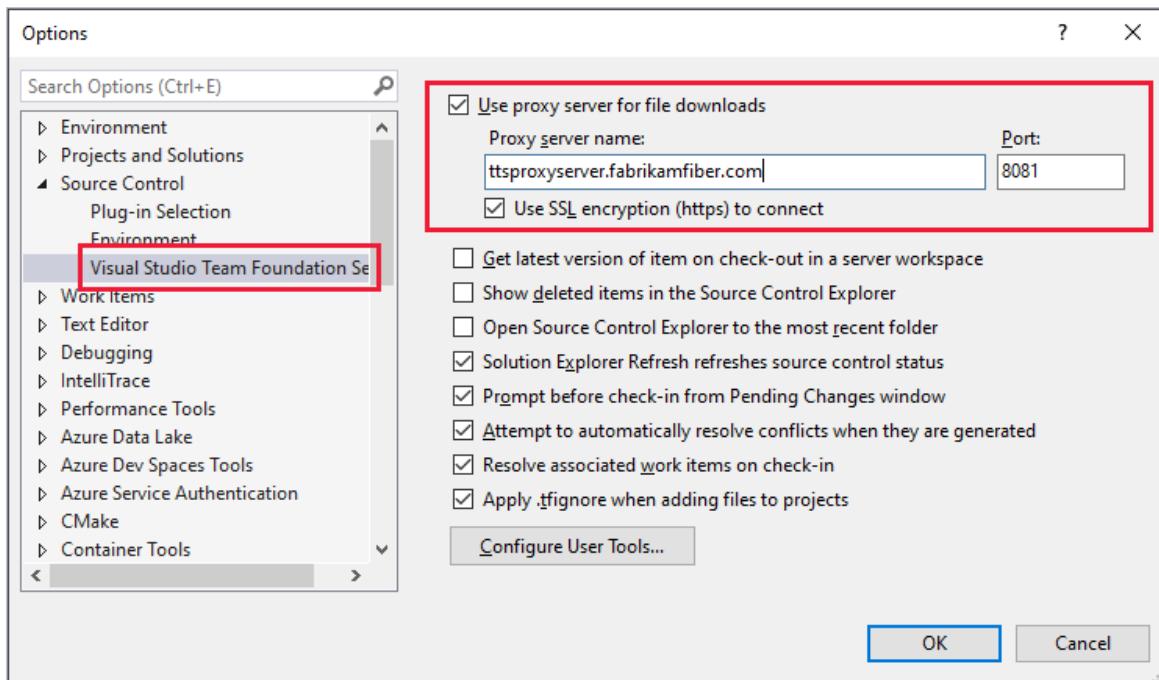
# Configure Visual Studio to connect to Azure DevOps Proxy Server

If your remote team uses a [Azure DevOps Proxy Server](#) to cache files, you can configure Visual Studio to connect through that proxy server and download files under Team Foundation version control.

1. First, make sure that you've connected to Azure DevOps Server as described [in the previous section](#).
2. From the Visual Studio Tools menu, select Options, then select Source Control > Plug-in Selection. Select Visual Studio Team Foundation Server.



3. For Visual Studio Team Foundation Server, enter the name and port number for the Azure DevOps Proxy Server. Select Use SSL encryption ([https](https://)) to connect.



Make sure you specify the port number that your administrator assigned to TFS Proxy.

To associate a file type with a compare or merge tool, see [Associate a file type with a file-comparison tool](#) or [Associate a file type with a merge tool](#).

**What other clients support connection to Azure DevOps?**

Besides connecting through a web browser, Visual Studio, Eclipse, Excel, and Project you can connect to a project from these clients:

- [Visual Studio Code](#)
- [Visual Studio Community](#)
- [Eclipse: Team Explorer Everywhere](#)
- [Azure Test Plans](#) (formerly Test Manager)
- [Microsoft Feedback Client](#)

### Requirements and client compatibility

Some tasks or features aren't available when you connect to a later version of Azure DevOps Server than your client supports. For more information, see [client compatibility](#).

### Determine your platform version

See [Look up your Azure DevOps platform and version](#).

## Next steps

Learn more about how to:

- [Work in web portal](#)
- [Work in Team Explorer](#)
- [Work in Office Excel or Project](#)
- [Troubleshoot connection](#)

If all you need is a code repository and bug tracking solution, then start with the [Get Started with Azure Repos](#) and [Manage bugs](#).

To start planning and tracking work, see [Get started with Agile tools to plan and track work](#).

# Code with Git

12/13/2022 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Learn how to share your code with others when you use a Git repository.

## Prerequisites

You must have an [organization](#) and [project](#) in Azure DevOps. When you create a project, Azure DevOps automatically creates an empty repository in Repos.

## 1. Install Git command-line tools

Install one of the following Git command-line tools:

- To install Git for Windows, including Git Credential Manager, see [Install the Git Credential Manager](#).
- To install on macOS or Linux, check out the [Installing Git](#) chapter in the open-source *Pro Git* book. For macOS and Linux, we recommend that you [configure SSH authentication](#).

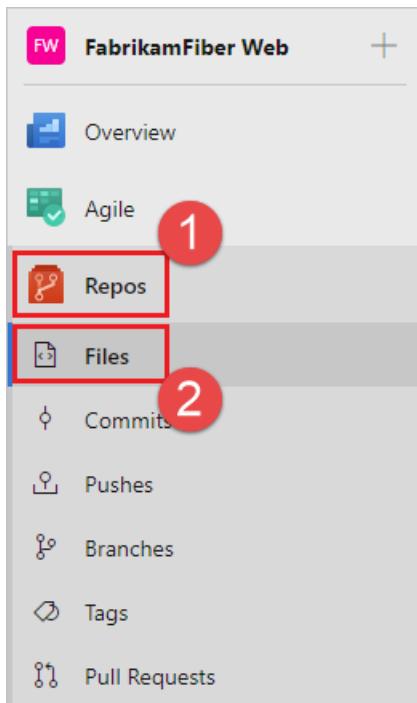
## 2. Clone the repo to your computer

To work with a Git repo, you clone it to your computer, which creates a complete local copy of the repo for you to work with. Your code might be in one of several places.

1. Complete the following step that's applicable to your scenario:

- If **You don't have any code yet**, first [Create a new Git repo in your project](#), and then complete the next step.
- If **the code is in another Git repo**, such as a GitHub repo or a different Azure Repo instance, [import it into a new or existing empty Git repo](#), and then complete the next step.
- If **the code is on your local computer and not yet in version control**, either [create a new Git repo in your project](#) or add your code to an existing repository.

2. From your web browser, open the team project for your organization and select **Repos > Files**.



3. Select **Clone** in the upper-right corner of the **Code** window and copy the URL.

FabrikamFiber01-01 is empty. Add some code!

Clone to your computer

HTTPS   SSH   [https://FabrikamFiber01@dev.azure.com/FabrikamFiber01/FabrikamFiber01/\\_git/FabrikamFiber01-01](https://FabrikamFiber01@dev.azure.com/FabrikamFiber01/FabrikamFiber01/_git/FabrikamFiber01-01) Copy OR [Clone in VS Code](#) ▼

Generate Git Credentials

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

4. Open the Git command window (Git Bash on Git for Windows). Go to the folder where you want the code from the repo stored on your computer, and run `git clone`, followed by the path copied from **Clone URL** in the previous step. See the following example:

```
git clone https://FabrikamFiber01@dev.azure.com/FabrikamFiber01/FabrikamFiber01-01/_git/FabrikamFiber01-01
```

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

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

```
cd fabrikam-web
```

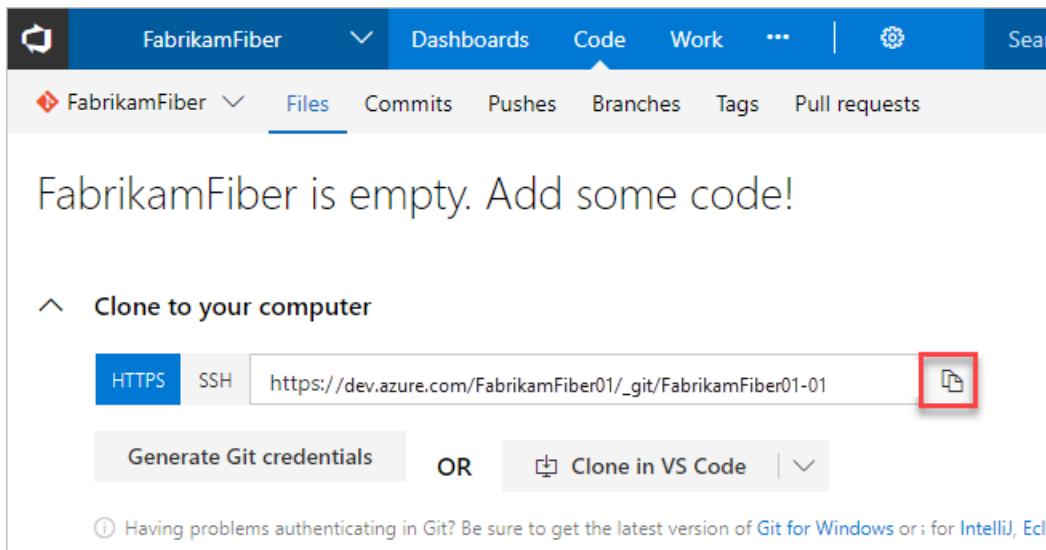
Keep this command window open to work in a branch.

1. Complete the following step that's applicable to your scenario:

- If You don't have any code yet, first [Create a new Git repo in your project](#), and then complete the next step.
- If the code is in another Git repo, such as a GitHub repo or a different Azure Repo instance, [import it into a new or existing empty Git repo](#), and then complete the next step.
- If the code is on your local computer and not yet in version control, either [create a new Git repo in your project](#) or add your code to an existing repository.

2. From your web browser, open the project for your organization, and select **Code**.

3. Select **Clone** in the upper-right corner of the Code window, and copy the URL.



4. Open the Git command window (Git Bash on Git for Windows). Go to the folder where you want the code from the repo stored on your computer, and run `git clone`, followed by the path copied from **Clone URL** in the previous step. See the following example:

```
git clone https://contoso-ltd.visualstudio.com/MyFirstProject/_git/contoso-demo
```

Git downloads a copy of the code in a new folder for you to work with. The download includes all **commits** and **branches** from the repo.

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

```
cd contoso-demo
```

Keep the command window open to work in a branch.

### 3. Work in a branch

Git **branches** isolate your changes from other work being done in the project. We recommend using the [Git workflow](#), which uses a new branch for every feature or fix that you work on. For our examples, we use the branch, `users/jamal/feature1`.

1. Create a branch with the `branch` command.

```
git branch users/jamal/feature1
```

This command creates a reference in Git for the new branch. It also creates a pointer back to the parent commit so Git can keep a history of changes as you add commits to the branch.

#### TIP

If you're working with a previously cloned repository, ensure that you've checked out the right branch (`git checkout main`) and that it's up to date (`git pull origin main`) before you create your new branch.

2. Use `checkout` to switch to that branch.

```
git checkout users/jamal/feature1
```

Git changes the files on your computer to match the latest commit on the checked-out branch.

**TIP**

When you create a branch from the command line, the branch is based on the currently checked-out branch.

When you clone the repository, the default branch (typically `main`) gets checked out. Because you cloned, your local copy of `main` has the latest changes.

```
git checkout main  
git pull origin main  
git branch users/jamal/feature1  
git checkout users/jamal/feature1
```

You can replace the first three commands in the previous example with the following command, which creates a new branch named `users/jamal/feature1` based on the latest `main` branch.

```
git pull origin main:users/jamal/feature1
```

Switch back to the Git Bash window that you used in the previous section. Run the following commands to create and check out a new branch based on the main branch.

```
git pull origin main:users/jamal/feature1  
git checkout feature1
```

## 4. Work with the code

In the following steps, we make a change to the files on your computer, commit the changes locally, and then push the commit to the repo stored on the server.

1. Browse to the folder on your computer where you cloned the repo, open the `README.md` file in your editor of choice, and make some changes. Then, **Save** and close the file.
2. In the Git command window, go to the `contoso-demo` directory by entering the following command:

```
cd contoso-demo
```

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

```
git add .  
git commit -m "My first commit"
```

The `git add .` command stages any new or changed files, and `git commit -m` creates a commit with the specified commit message.

**TIP**

Check what branch you're working on before you commit, so that you don't commit changes to the wrong branch. Git always adds new commits to the current local branch.

4. Push your changes to the Git repo on the server. Enter the following command into the Git command

window:

```
git push origin users/jamal/feature1
```

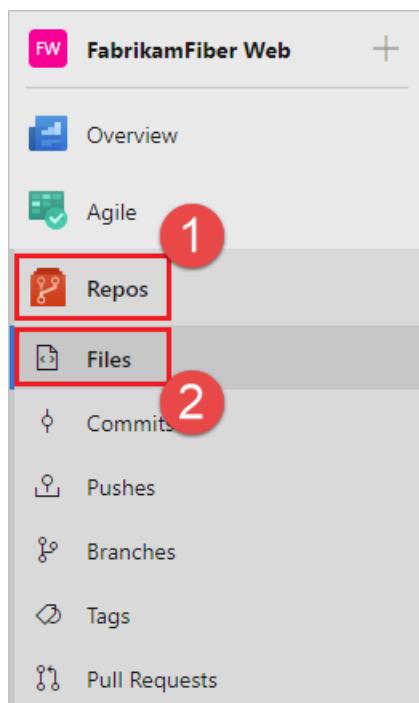
Your code is now shared to the remote repository, in a branch named `users/jamal/feature1`. To merge the code from your working branch into the `main` branch, use a pull request.

## 5. Merge your changes with a pull request

Pull requests combine the review and merge of your code into a single collaborative process. After you're done fixing a bug or new feature in a branch, create a new pull request. Add the members of the team to the pull request so they can review and vote on your changes. Use pull requests to review works in progress and get early feedback on changes. There's no commitment to merge the changes because you can abandon the pull request at any time.

The following example shows the basic steps of creating and completing a pull request.

1. Open the team project for your organization in your web browser and select **Repos > Files**. If you kept your browser open after getting the clone URL, you can just switch back to it.



2. Select **Create a pull request** in the upper-right corner of the **Files** window. If you don't see a message like **You updated users/jamal/feature1 just now**, refresh your browser.

A screenshot of the 'Files' window in the Microsoft Team Project. At the top, there is a search bar and a message: '(i) You updated 🌐 users/jamal/feature1 just now — Create a pull request'. Below the message are buttons for 'Set up build', 'Fork', and 'Clone'. The main area shows a list of files with columns for Name, Last change, and Commits. The files listed are: AzureEndpoint.png, gear.png, and README.md, all last changed on 5/2/2018 by ae9e9911 with initial commits.

New pull requests are configured to merge your branch into the default branch, which in this example is `main`. The title and description are pre-populated with your commit message.

The screenshot shows the 'New Pull Request' dialog. At the top, it displays the source branch as 'users/jamal/feature1' and the target branch as 'main'. Below this, the 'Title' field contains 'My first commit'. A 'Description' section follows, also containing 'My first commit'. A note 'Markdown supported.' is present. Below the description is a rich text editor toolbar with icons for bold, italic, link, and other styling options. The main body of the dialog contains the text 'My first commit'. Underneath, there are sections for 'Reviewers' (with a search bar) and 'Work Items' (with a search bar). At the bottom right is a large blue 'Create' button.

You can [add reviewers](#) and [link work items](#) to your pull request.

You can review the files included in the pull request at the bottom of the **New Pull Request** window.

The screenshot shows the pull request details page. At the top right is a blue 'Create' button. Below it, there are tabs for 'Files (1)' and 'Commits (1)', with 'Files (1)' being active. The main area shows a single file change: 'Showing 1 file change: 1 edit'. The file 'README.md' has been modified. The diff view shows the following code changes:

```
M README.md +1 -1 /README.md
...
13 13
14 14 -----
15 15
16 - Please follow below exercises inorder to deploy your application, :
16 + Please follow below exercises inorder to deploy your application:
17 17
18 18 ## Exercise 1: Endpoint Creation
19 19
...
```

3. Select **Create**.

View the details of your pull request from the **Overview** tab. You can also view the changed files, updates, and commits in your pull request from the other tabs.

4. Select **Complete** to begin the process of completing the pull request.

The screenshot shows the 'My first commit' pull request details page. At the top, there are tabs for 'Overview', 'Files', 'Updates', and 'Commits'. The 'Overview' tab is selected. In the top right, there are buttons for 'Approve' and 'Complete', with 'Complete' being highlighted by a red box. Below the tabs, there's a 'Description' section containing 'My first commit'. A comment input field says 'Add a comment...'. Below it, it shows 'Created by Jamal Hartnett just now'. To the right, there are sections for 'Work Items' (no related work items), 'Reviewers' (no reviewers), and 'Labels' (add label). The entire interface has a light blue background.

5. Select **Complete merge** to complete the pull request and merge your code into the `main` branch.

The screenshot shows the 'Complete pull request' dialog box. It displays a summary of the merged pull request: 'Merged PR 5: My first commit' and 'My first commit'. Below this, there are three checkboxes:

- Complete linked work items after merging [\(i\)](#)
- Delete users/jamal/feature1 after merging
- Squash changes when merging [Learn more](#)

At the bottom, there are 'Complete merge' and 'Cancel' buttons, with 'Complete merge' being highlighted by a red box.

#### NOTE

This example shows the basic steps of creating and completing a pull request. For more information, see [Create, view, and manage pull requests](#).

1. Open the team project for your organization from your web browser and select the **Code** page.
2. Select **Clone** in the upper-right corner of the **Code** page and copy the **Clone URL**.

FabrikamFiber01-01 is empty. Add some code!

Clone to your computer

HTTPS

SSH

[https://FabrikamFiber01@dev.azure.com/FabrikamFiber01/FabrikamFiber01/\\_git/FabrikamFiber01-01](https://FabrikamFiber01@dev.azure.com/FabrikamFiber01/FabrikamFiber01/_git/FabrikamFiber01-01)



OR

[Clone in VS Code](#)

[Generate Git Credentials](#)

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

3. Open the Git command window, for example Git Bash on Git for Windows, and browse to the folder where the repo is stored on your computer.

Run `git clone` 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/contoso-ltd/MyFirstProject/_git/contoso-demo
```

Git downloads a copy of the code into a new folder for you to work with. The download includes all [commits](#) and [branches](#) from the repo.

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

```
cd fabrikam-web
```

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

Your changes are now merged into the `main` branch, and your `users/jamal/feature1` branch is deleted on the remote repository.

## View history

1. Switch back to the web portal and select **History** from the **Code** page to view your new commit.

The screenshot shows the Azure DevOps Code History interface. At the top, there's a navigation bar with a redacted repository name, a dropdown for branches (set to master), and a search bar. Below the navigation is a tabs section with 'Files' (selected) and 'History'. Underneath is a grid with two columns: 'Graph' and 'Commit'. The 'Commit' column displays a single commit entry:

- Added README.md**
- Author: Jamal Hartnett
- Date: Just now

2. Switch to the **Files** tab, and select the README file to view your changes.

Checked in changeset 1529: Updated README

Contents History Compare Annotate

1 My first edit to README file.

1. Switch back to the web portal, and select **History** from the **Code** tab to view your new commit. Two commits appear: the first commit, where the README and .gitignore were added upon repo creation, and the commit you just made.

MyFirstProject Dashboards Code **History** Work ...

contoso-demo Files History Branches Tags Pull Requests

master contoso-demo / Type to find a file or folder...

Commits Branch Updates

Simple history (...) Author From date To date

Graph Commit Message

4b38f92b My first commit

fefb3a74 Added README.md, .gitignore (VisualStudio) files

2. Switch to the **Files** tab, and select the README file to view your changes.

MyFirstProject Dashboards Code Work ...

contoso-demo Files History Branches Tags Pull Requests

master contoso-demo / README.md

contoso-demo .gitignore README.md

Contents Preview History Compare Blame

Introduction

This is my first edit.

TODO: Give a short introduction of your project. Let the user know what the project does, who created it, etc.

Getting Started

TODO: Guide users through getting your code up and running. This can include instructions for setting up dependencies, running tests, or building the project.

## Clean up

Switch back to your Git Bash command prompt and run the following command to delete your local copy of the branch.

```
git checkout main
git pull origin main
git branch -d users/jamal/feature1
```

This action completes the following tasks:

- The `git checkout main` command switches you to the `main` branch.
- The `git pull origin main` command pulls down the latest version of the code in the main branch, including your changes and the fact that `users/jamal/feature1` was merged.
- The `git branch -d users/jamal/feature1` command deletes your local copy of that branch.

## Next steps

[Set up continuous integration & delivery](#)

## Related articles

- [Key concepts for new users to Azure Pipelines](#)
- [What is Azure Repos?](#)
- [Learn more about working with a Git repo](#)
- [What is source control?](#)

# Create your first pipeline

12/13/2022 • 26 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This is a step-by-step guide to using Azure Pipelines to build a sample application. This guide uses YAML pipelines configured with the [YAML pipeline editor](#). If you'd like to use Classic pipelines instead, see [Define your Classic pipeline](#).

## Prerequisites - Azure DevOps

Make sure you have the following items:

- A GitHub account where you can create a repository. [Create one for free](#).
- An Azure DevOps organization. [Create one for free](#). If your team already has one, then make sure you're an administrator of the Azure DevOps project that you want to use.
- An ability to run pipelines on Microsoft-hosted agents. You can either purchase a [parallel job](#) or you can request a free tier.

## Create your first pipeline

- [Java](#)
- [.NET](#)
- [Python](#)
- [JavaScript](#)
- [Azure CLI \(Java\)](#)

### Get the Java sample code

To get started, fork the following repository into your GitHub account.

```
https://github.com/MicrosoftDocs/pipelines-java
```

### Create your first Java pipeline

1. Sign-in to your Azure DevOps organization and go to your project.
2. Go to **Pipelines**, and then select **New pipeline**.
3. Do the steps of the wizard by first selecting **GitHub** as the location of your source code.
4. You might be redirected to GitHub to sign in. If so, enter your GitHub credentials.
5. When you see the list of repositories, select your repository.
6. You might be redirected to GitHub to install the Azure Pipelines app. If so, select **Approve & install**.
7. Azure Pipelines will analyze your repository and recommend the **Maven** pipeline template.
8. When your new pipeline appears, take a look at the YAML to see what it does. When you're ready, select **Save and run**.

9. You're prompted to commit a new `azure-pipelines.yml` file to your repository. After you're happy with the message, select **Save and run** again.

If you want to watch your pipeline in action, select the build job.

You just created and ran a pipeline that we automatically created for you, because your code appeared to be a good match for the [Maven](#) template.

You now have a working YAML pipeline (`azure-pipelines.yml`) in your repository that's ready for you to customize!

10. When you're ready to make changes to your pipeline, select it in the **Pipelines** page, and then **Edit** the `azure-pipelines.yml` file.

Learn more about [working with Java](#) in your pipeline.

## Add a status badge to your repository

Many developers like to show that they're keeping their code quality high by displaying a status badge in their repo.



To copy the status badge to your clipboard:

1. In Azure Pipelines, go to the **Pipelines** page to view the list of pipelines. Select the pipeline you created in the previous section.
2. Select , and then select **Status badge**.
3. Select **Status badge**.
4. Copy the sample Markdown from the Sample markdown section.

Now with the badge Markdown in your clipboard, take the following steps in GitHub:

1. Go to the list of files and select `Readme.md`. Select the pencil icon to edit.
2. Paste the status badge Markdown at the beginning of the file.
3. Commit the change to the `main` branch.
4. Notice that the status badge appears in the description of your repository.

To configure anonymous access to badges for private projects:

1. Navigate to **Project Settings**
2. Open the **Settings** tab under **Pipelines**
3. Toggle the **Disable anonymous access to badges** slider under **General**

### NOTE

Even in a private project, anonymous badge access is enabled by default. With anonymous badge access enabled, users outside your organization might be able to query information such as project names, branch names, job names, and build status through the badge status API.

Because you just changed the `Readme.md` file in this repository, Azure Pipelines automatically builds your code, according to the configuration in the `azure-pipelines.yml` file at the root of your repository. Back in Azure Pipelines, observe that a new run appears. Each time you make an edit, Azure Pipelines starts a new run.

#### NOTE

In Microsoft Team Foundation Server (TFS) 2018 and previous versions, build and release *pipelines* are called *definitions*, *runs* are called *builds*, *service connections* are called *service endpoints*, *stages* are called *environments*, and *jobs* are called *phases*.

We'll show you how to use the classic editor in Azure DevOps Server 2019 to create a build and release that prints "Hello world".

We'll show you how to use the classic editor in TFS to create a build and a release that prints "Hello world".

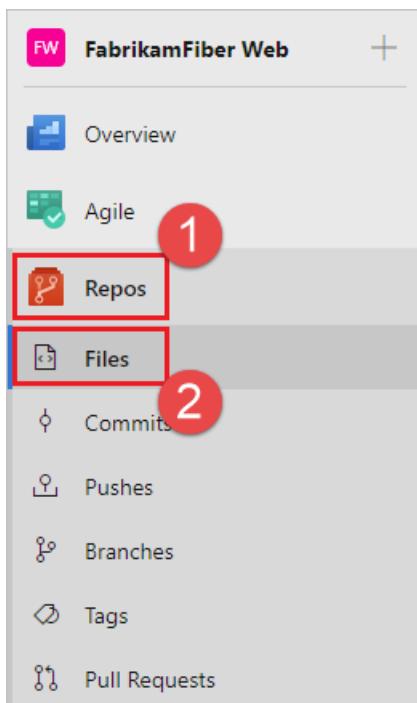
## Prerequisites

- A [self-hosted Windows agent](#).

## Initialize your repository

If you already have a repository in your project, you can skip to the next step: [Skip to adding a script to your repo](#)

1. Go to [Azure Repos](#). (The Code hub in the previous navigation)



2. If your project is empty, you will be greeted with a screen to help you add code to your repository. Choose the bottom choice to **initialize** your repo with a `readme` file:

## FabrikamFiber Pipelines is empty. Add some code!

### Clone to your computer

HTTPS SSH [https://regius.visualstudio.com/FabrikamFiber%20Pipelines/\\_git/Fabrika...](https://regius.visualstudio.com/FabrikamFiber%20Pipelines/_git/Fabrika...) OR [Clone in VS Code](#) | [▼](#)

Generate Git credentials

ⓘ 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](#).

### or push an existing repository from command line

HTTPS SSH

```
git remote add origin https://regius.visualstudio.com/FabrikamFiber%20Pipelines/_git/FabrikamFiber%20Pipelines  
git push -u origin --all
```

### or import a repository

Import

#### or initialize with a README or gitignore

Add a README

Add a .gitignore: None ▾

Initialize

1. Navigate to your repository by clicking **Code** in the top navigation.

2. If your project is empty, you will be greeted with a screen to help you add code to your repository. Choose the bottom choice to **initialize** your repo with a `readme` file:

## FabrikamFiber Pipelines is empty. Add some code!

### Clone to your computer

HTTPS SSH [https://regius.visualstudio.com/FabrikamFiber%20Pipelines/\\_git/Fabrika...](https://regius.visualstudio.com/FabrikamFiber%20Pipelines/_git/Fabrika...) OR [Clone in VS Code](#) | [▼](#)

Generate Git credentials

ⓘ 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](#).

### or push an existing repository from command line

HTTPS SSH

```
git remote add origin https://regius.visualstudio.com/FabrikamFiber%20Pipelines/_git/FabrikamFiber%20Pipelines  
git push -u origin --all
```

### or import a repository

Import

#### or initialize with a README or gitignore

Add a README

Add a .gitignore: None ▾

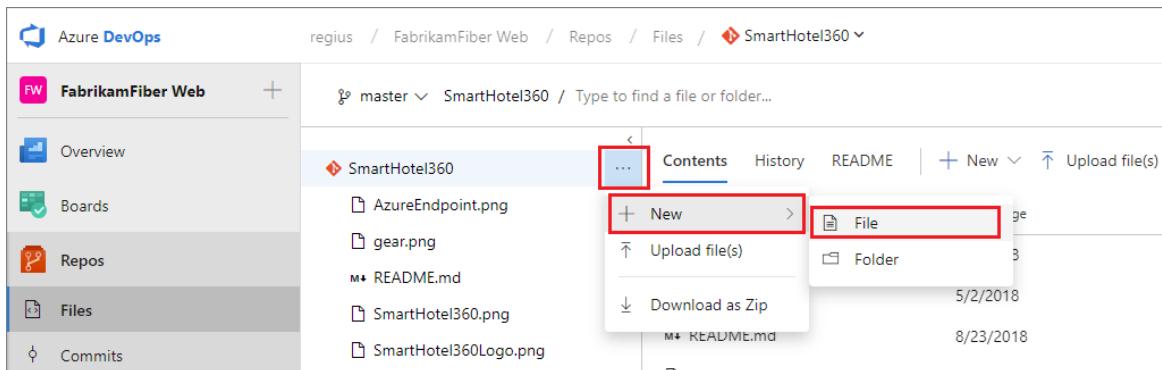
Initialize

## Add a script to your repository

Create a PowerShell script that prints `Hello world`.

1. Go to Azure Repos.

2. Add a file.



3. In the dialog box, name your new file and create it.

```
HelloWorld.ps1
```

4. Copy and paste this script.

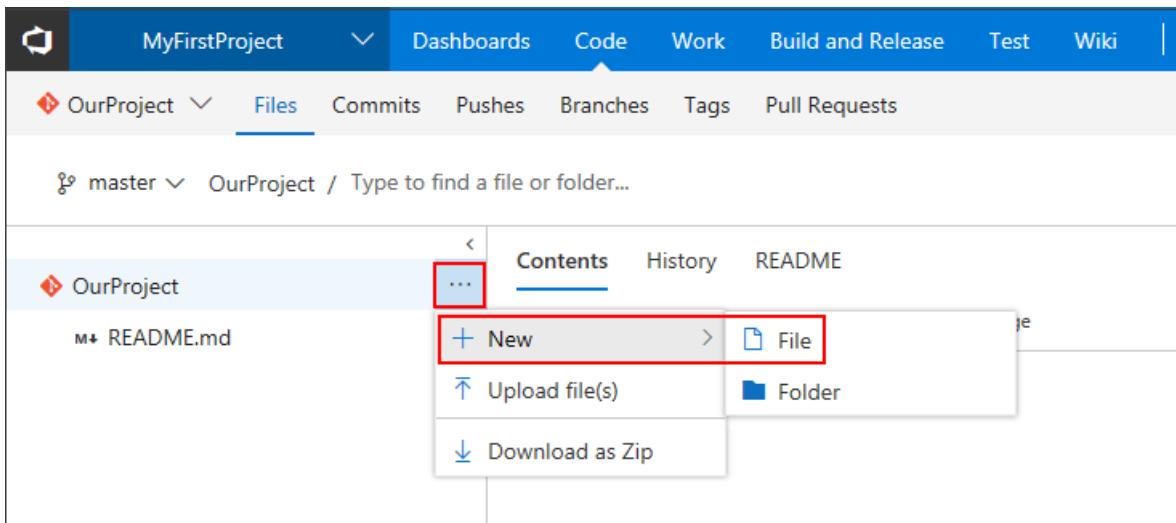
```
Write-Host "Hello world"
```

5. Commit (save) the file.

1. Go to the Code hub.

2. Add a file.

- [TFS 2018.2](#)
- [TFS 2018 RTM](#)



1. In the dialog box, name your new file and create it.

```
HelloWorld.ps1
```

2. Copy and paste this script.

```
Write-Host "Hello world"
```

### 3. Commit (save) the file.

In this tutorial, our focus is on CI/CD, so we're keeping the code part simple. We're working in an Azure Repos Git repository directly in your web browser.

When you're ready to begin building and deploying a real app, you can use a wide range of version control clients and services with Azure Pipelines CI builds. [Learn more](#).

## Create a build pipeline

Create a build pipeline that prints "Hello world."

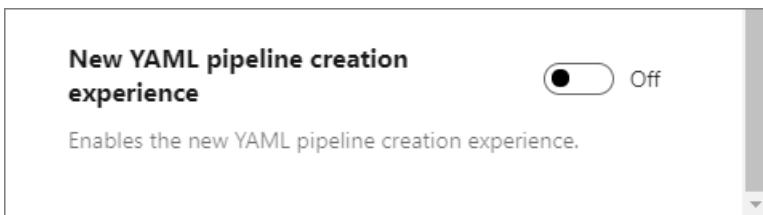
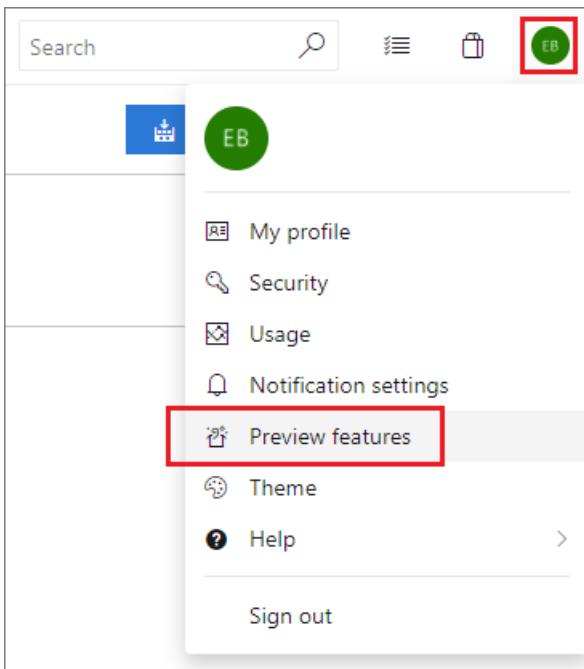
1. Select **Azure Pipelines**, it should automatically take you to the **Builds** page.

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber Web' project. The left sidebar has a red box around the 'Pipelines' and 'Builds' items, with 'Builds' currently selected. The main area displays a search bar and a list of pipelines. One pipeline, 'PublicWebCI', is shown with a status of 'master · 8 hours ago'. To the right, there are columns for 'Pub', 'Hist', 'Com', and three user icons.

2. Create a new pipeline.

The screenshot shows the same Azure DevOps interface as the previous one, but the 'Builds' section is no longer visible. Instead, the 'New' button in the top right of the main area is highlighted with a red box. A tooltip over the button says 'New build pipeline'. The rest of the interface remains the same, including the sidebar and the list of pipelines on the right.

For new Azure DevOps users, this will automatically take you to the *YAML pipeline creation experience*. To get to the classic editor and complete this guide, you must turn off the **preview feature** for the *New YAML pipeline creation experience*.



3. Make sure that the **source**, **project**, **repository**, and default **branch** match the location in which you created the script.
4. Start with an **Empty job**.
5. On the left side, select **Pipeline** and specify whatever **Name** you want to use. For the **Agent pool**, select **Hosted VS2017**.
6. On the left side, select the plus sign ( + ) to add a task to **Job 1**. On the right side, select the **Utility** category, select the **PowerShell** task from the list, and then choose **Add**.

A screenshot of the Azure DevOps Pipeline editor. The top navigation bar shows 'Tasks', 'Variables', 'Triggers', 'Options', 'Retention', 'History', 'Save &amp; queue', and 'Discard'. The main area is titled 'Pipeline' with the sub-section 'Build pipeline'. On the left, there's a 'Get sources' task for 'SmartHotel360' branch 'master'. On the right, there's an 'Agent job 1' section with a button labeled 'Add a task to Agent job 1' highlighted with a red box. The 'Run on agent' checkbox is checked.

7. On the left side, select your new **PowerShell** script task.
8. For the **Script Path** argument, select the button to browse your repository and select the script you created.

The screenshot shows the 'Pipeline' tab in the Azure DevOps interface. A 'PowerShell' task is selected, with its configuration pane open. The 'Display name' is set to 'PowerShell Script'. The 'Type' is set to 'File Path', and the 'Script Path' is set to 'HelloWorld.ps1'. A red box highlights the 'HelloWorld.ps1' input field.

9. Select **Save & queue**, and then select **Save**.

10. Select **Build and Release**, and then choose **Builds**.

The screenshot shows the 'Builds' page under the 'Build and Release' section. It lists files in the repository: 'HelloWorld.ps1' and 'README.md'. The 'HelloWorld.ps1' file was last changed 9 minutes ago, and the 'README.md' file was last changed 17 minutes ago.

11. Create a new pipeline.

The screenshot shows the 'Build Definitions' page under the 'Builds' section. It lists build definitions: 'Mine', 'Definitions', and 'Queued'. A red box highlights the '+ New' button at the top right.

12. Start with an **empty pipeline**

13. Select **Pipeline** and specify whatever **Name** you want to use. For the **Agent pool**, select **Default**.

14. On the left side, select **+ Add Task** to add a task to the job, and then on the right side select the **Utility** category, select the **PowerShell** task, and then choose **Add**.

The screenshot shows the 'Process' pipeline configuration. It includes a 'Get sources' task and a placeholder for adding a new task. A red box highlights the '+ Add Task' button.

15. On the left side, select your new **PowerShell** script task.

16. For the **Script Path** argument, select the button to browse your repository and select the script you

created.

The screenshot shows the 'Tasks' tab of a build pipeline named 'Build process'. It contains a 'Get sources' task and a selected 'PowerShell Script' task. The PowerShell task has a 'Display name' of 'PowerShell Script', a 'Type' of 'File Path', and a 'Script Path' of 'HelloWorld.ps1'. A red box highlights the 'HelloWorld.ps1' input field.

17. Select **Save & queue**, and then select **Save**.

A build pipeline is the entity through which you define your automated build pipeline. In the build pipeline, you compose a set of tasks, each of which perform a step in your build. The task catalog provides a rich set of tasks for you to get started. You can also add PowerShell or shell scripts to your build pipeline.

## Publish an artifact from your build

A typical build produces an artifact that can then be deployed to various stages in a release. Here to demonstrate the capability in a simple way, we'll simply publish the script as the artifact.

1. On the **Tasks** tab, select the plus sign ( + ) to add a task to **Job 1**.
2. Select the **Utility** category, select the **Publish Build Artifacts** task, and then select **Add**.

The screenshot shows the 'Tasks' tab of a build pipeline. It includes a 'Get sources' task and a selected 'Publish Artifact: drop' task. The 'Publish Artifact: drop' task has a 'Display name' of 'Publish Artifact: drop', a 'Path to publish' of 'HelloWorld.ps1', an 'Artifact name' of 'drop', and an 'Artifact publish location' of 'Visual Studio Team Services/TFS'. A large red box highlights the entire configuration area of the task card.

**Path to publish:** Select the **...** button to browse and select the script you created.

**Artifact name:** Enter **drop**.

**Artifact publish location:** Select **Azure Artifacts/TFS**.

1. On the **Tasks** tab, select **Add Task**.
2. Select the **Utility** category, select the **Publish Build Artifacts** task, and then select **Add**.

The screenshot shows the Azure Pipelines interface for managing a build pipeline. The 'Tasks' tab is active. On the left, a list of tasks is shown: 'Get sources' (OurProject, master), 'PowerShell Script' (PowerShell), and 'Publish Artifact: drop' (highlighted with a red border). The 'Publish Artifact: drop' task has a checkmark and three vertical dots. Below the tasks, there's a '+ Add Task' button. To the right, the 'Publish Build Artifacts' task configuration is displayed. It includes fields for 'Display name' (Publish Artifact: drop), 'Path to Publish' (HelloWorld.ps1, with a browse button ...), 'Artifact Name' (drop), and 'Artifact Type' (Server). A 'Link settings' button is also present.

**Path to Publish:** Select the **...** button to browse and select the script you created.

**Artifact Name:** Enter **drop**.

**Artifact Type:** Select **Server**.

Artifacts are the files that you want your build to produce. Artifacts can be nearly anything your team needs to test or deploy your app. For example, you've got a .DLL and .EXE executable files and .PDB symbols file of a C# or C++ .NET Windows app.

To enable you to produce artifacts, we provide tools such as copying with pattern matching, and a staging directory in which you can gather your artifacts before publishing them. See [Artifacts in Azure Pipelines](#).

## Enable continuous integration (CI)

1. Select the **Triggers** tab.
2. Enable **Continuous integration**.

A continuous integration trigger on a build pipeline indicates that the system should automatically queue a new build whenever a code change is committed. You can make the trigger more general or more specific, and also schedule your build (for example, on a nightly basis). See [Build triggers](#).

## Save and queue the build

Save and queue a build manually and test your build pipeline.

1. Select **Save & queue**, and then select **Save & queue**.
2. On the dialog box, select **Save & queue** once more.

This queues a new build on the Microsoft-hosted agent.

3. You see a link to the new build on the top of the page.

The screenshot shows the Azure Pipelines interface for the 'FabrikamFiber Web' project. A green notification bar at the top right indicates that 'Build #116 has been queued.' The pipeline name 'FabrikamFiber Web-CI' is visible above the queue status.

Choose the link to watch the new build as it happens. Once the agent is allocated, you'll start seeing the live logs of the build. Notice that the PowerShell script is run as part of the build, and that "Hello world" is printed to the console.

The screenshot shows the build summary for 'Agent job 1 Job'. The 'Logs' tab is selected, displaying the PowerShell script output:

```
PowerShell Script
1 2018-08-27T17:30:29.8976103Z ##[section]Starting: PowerShell Script
2 2018-08-27T17:30:29.8988064Z =====
3 2018-08-27T17:30:29.8988307Z Task : PowerShell
4 2018-08-27T17:30:29.8988513Z Description : Run a PowerShell script on Windows, macOS, or Linux.
5 2018-08-27T17:30:29.8988736Z Version : 2.136.0
6 2018-08-27T17:30:29.8988923Z Author : Microsoft Corporation
7 2018-08-27T17:30:29.8989137Z Help : [More Information](https://go.microsoft.com/fwlink/?LinkID=857000)
8 2018-08-27T17:30:32.6103320Z Generating script.
9 2018-08-27T17:30:32.8474601Z Formatted command: . 'D:\a\1\s\HelloWorld.ps1'
10 2018-08-27T17:30:32.8474601Z ##[command]"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
11 2018-08-27T17:30:33.0990683Z Hello world
12 2018-08-27T17:30:33.2699756Z ##[section]Finishing: PowerShell Script
13
14
```

The 'Artifacts' tab is also visible, showing a list of successful steps:

- ✓ Prepare job · succeeded
- ✓ Initialize Agent · succeeded
- ✓ Initialize Job · succeeded
- ✓ Checkout · succeeded
- ✓ PowerShell Script · succeeded
- ✓ Publish Artifact: drop · succeeded
- ✓ Post-job: Checkout · succeeded
- ✓ Report build status · succeeded

4. Go to the build summary. On the Artifacts tab of the build, notice that the script is published as an artifact.

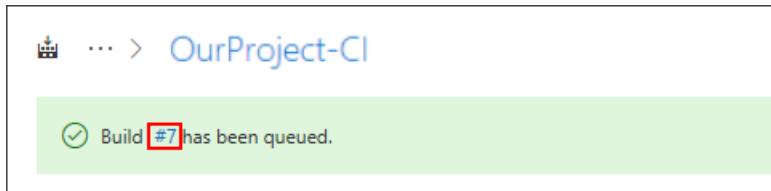
The screenshot shows the build summary for 'FabrikamFiber Web-CI 162'. The 'Summary' tab is selected. The 'Artifacts' section shows one artifact named 'drop' under 'File container'.

1. Select Save & queue, and then select Save & queue.

2. On the dialog box, select **Save & queue** once more.

This queues a new build on the Microsoft-hosted agent.

3. You see a link to the new build on the top of the page.



Choose the link to watch the new build as it happens. Once the agent is allocated, you'll start seeing the live logs of the build. Notice that the PowerShell script is run as part of the build, and that "Hello world" is printed to the console.

- [TFS 2018.2](#)
- [TFS 2018 RTM](#)

A screenshot of the TFS 2018.2 build summary page. The top navigation bar includes 'Builds', 'Releases', 'Packages', 'Library', 'Task Groups', 'Deployment Groups\*', and 'Explorer'. The 'Builds' tab is selected. On the left, a tree view shows 'Build 1722' expanded, with 'Build' selected. Under 'Build', several tasks are listed with green checkmarks: 'Initialize Agent', 'Initialize Job', 'Get Sources', 'PowerShell Script', 'Publish Artifact: drop', 'Post Job Cleanup', 'Finalize build', and 'Report build status'. To the right, the build details are shown. The title is 'Hello world / Build 1722 / Build'. A toggle switch is set to 'Build not retained'. Below the title are buttons for 'Edit build definition', 'Queue new build...', 'Download all logs as zip', and 'Release'. A large green banner at the top says 'Build succeeded'. Below it, a bar chart indicates the build status. The log output shows the PowerShell script execution: 'git checkout -b &lt;new-branch-name&gt;', 'HEAD is now at 0ab86c0... Updated HelloWorld.ps1', 'Finishing: Get Sources', 'Starting: PowerShell Script', and the final output 'Hello world'. Navigation tabs for 'Console', 'Logs', 'Code coverage\*', and 'Tests' are visible.

4. Go to the build summary.

A screenshot of the TFS 2018.2 build summary page, similar to the previous one but with a different URL. The top navigation bar and the tree view on the left are identical. The build details on the right show the same information: title 'Hello world / Build 1722 / Build', 'Build not retained' switch, and the 'Build succeeded' banner. The log output shows the PowerShell script execution. The URL in the browser's address bar is 'Hello world / **Build 1722** / Build', where 'Build 1722' is highlighted with a red box.

5. On the **Artifacts** tab of the build, notice that the script is published as an artifact.

Build succeeded

Build 1722  
Ran for 31 seconds (Hosted), completed 15.4 minutes ago

Summary Timeline Artifacts\* Code coverage\* Tests

Name ↑

drop Download Explore

Artifacts Explorer

drop

HelloWorld.ps1

You can view a summary of all the builds or drill into the logs for each build at any time by navigating to the **Builds** tab in **Azure Pipelines**. For each build, you can also view a list of commits that were built and the work items associated with each commit. You can also run tests in each build and analyze the test failures.

## Add some variables and commit a change to your script

We'll pass some build variables to the script to make our pipeline a bit more interesting. Then we'll commit a change to a script and watch the CI pipeline run automatically to validate the change.

1. Edit your build pipeline.
2. On the **Tasks** tab, select the PowerShell script task.
3. Add these arguments.

regius / FabrikamFiber Web / Pipelines

FabrikamFiber Web-Cl

Tasks Variables Triggers Options Retention History Save & queue Discard Summary Queue ...

Pipeline Build pipeline

Get sources SmartHotel360 master

Agent job 1 Run on agent

PowerShell Script PowerShell

Publish Artifact: drop Publish Build Artifacts

PowerShell Version 2.\*

Display name \* PowerShell Script

Type File Path

Script Path \* HelloWorld.ps1

Arguments -greeter "\$(Build.RequestedFor)" -trigger "\$(Build.Reason)"

- TFS 2018.2
- TFS 2018 RTM

The screenshot shows the Azure DevOps build pipeline editor. On the left, there's a list of tasks: 'Process Build process', 'Get sources' (with 'OurProject' and 'master' selected), 'Phase 1' (with 'Run on agent'), and 'PowerShell Script' (selected, indicated by a blue background). Below these are 'Publish Artifact: drop' and 'Publish Build Artifacts'. On the right, the 'PowerShell Script' task is being configured. It has a 'Version' dropdown set to '1.\*'. The 'Display name' is 'PowerShell Script'. The 'Type' is 'File Path', with 'HelloWorld.ps1' specified. The 'Script Path' is also 'HelloWorld.ps1'. The 'Arguments' field contains the command: '-greeter "\$(Build.RequestedFor)" -trigger "\$(Build.Reason)"'. This argument line is highlighted with a red rectangle.

## Arguments

```
-greeter "$(Build.RequestedFor)" -trigger "$(Build.Reason)"
```

Finally, save the build pipeline.

Next you'll add the arguments to your script.

1. Go to your **Files** in **Azure Repos** (the **Code** hub in the previous navigation and TFS).
2. Select the **HelloWorld.ps1** file, and then **Edit** the file.
3. Change the script as follows:

```
Param(
[string]$greeter,
[string]$trigger
)
Write-Host "Hello world" from $greeter
Write-Host Trigger: $trigger
```

4. **Commit** (save) the script.

Now you can see the results of your changes. Go to **Azure Pipelines** and select **Queued**. Notice under the **Queued or running** section that a build is automatically triggered by the change that you committed.

Now you can see the results of your changes. Go to the **Build and Release** page and select **Queued**. Notice under the **Queued or running** section that a build is automatically triggered by the change that you committed.

1. Select the new build that was created and view its log.
2. Notice that the person who changed the code has their name printed in the greeting message. You also see printed that this was a CI build.

```

1 2018-08-30T17:33:29.1723775Z ##[section]Starting: PowerShell Script
2 2018-08-30T17:33:29.1729508Z =====
3 2018-08-30T17:33:29.1729715Z Task : PowerShell
4 2018-08-30T17:33:29.1729878Z Description : Run a PowerShell script on Windows, macOS, or Linux.
5 2018-08-30T17:33:29.1730038Z Version : 2.136.0
6 2018-08-30T17:33:29.1730265Z Author : Microsoft Corporation
7 2018-08-30T17:33:29.1730373Z Help : [More Information](https://go.microsoft.com/fwlink/?LinkId=902752)
8 2018-08-30T17:33:29.1730567Z =====
9 2018-08-30T17:33:30.9668773Z Generating script.
10 2018-08-30T17:33:30.9756003Z Formatted command: . 'D:\a\1\s\HelloWorld.ps1' -greeter "Elijah Batkoski"
11 2018-08-30T17:33:31.0882200Z ##[command]"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -Command ". 'D:\a\1\s\HelloWorld.ps1' -greeter "Elijah Batkoski"
12 2018-08-30T17:33:31.3101480Z Hello world from Elijah Batkoski
13 2018-08-30T17:33:31.3107687Z Trigger: Manual
14 2018-08-30T17:33:31.4515161Z ##[section]Finishing: PowerShell Script
15

```

**Hello world / Build 1723 / Build / PowerShell Script**

**Build succeeded**

**Logs**

```

1 2017-04-10T20:55:12.0502205Z ##[section]Starting: PowerShell Script
2 2017-04-10T20:55:12.0592196Z =====
3 2017-04-10T20:55:12.0602014Z Task : PowerShell
4 2017-04-10T20:55:12.0602014Z Description : Run a PowerShell script
5 2017-04-10T20:55:12.0602014Z Version : 1.2.3
6 2017-04-10T20:55:12.0602014Z Author : Microsoft Corporation
7 2017-04-10T20:55:12.0602014Z Help : [More Information](https://go.microsoft.com/fwlink/?LinkId=902752)
8 2017-04-10T20:55:12.0602014Z =====
9 2017-04-10T20:55:12.1292010Z ##[command]. 'd:\a\1\s\HelloWorld.ps1' -g
10 2017-04-10T20:55:12.8952061Z Hello world from Raisa Pokrovskaya
11 2017-04-10T20:55:12.8952061Z Trigger: IndividualCI
12 2017-04-10T20:55:12.9002073Z ##[section]Finishing: PowerShell Script

```

We just introduced the concept of build variables in these steps. We printed the value of a variable that is automatically predefined and initialized by the system. You can also define custom variables and use them either in arguments to your tasks, or as environment variables within your scripts. To learn more about variables, see [Build variables](#).

## You've got a build pipeline. What's next?

You've created a build pipeline that automatically builds and validates whatever code is checked in by your team. At this point, you can continue to the next section to learn about release pipelines. Or, if you prefer, you can [skip ahead](#) to create a build pipeline for your app.

## Create a release pipeline

Define the process for running the script in two stages.

1. Go to the **Pipelines** tab, and then select **Releases**.
2. Select the action to create a **New pipeline**. If a release pipeline is already created, select the plus sign (+) and then select **Create a release pipeline**.
3. Select the action to start with an **Empty job**.
4. Name the stage **QA**.
5. In the Artifacts panel, select + **Add** and specify a **Source (Build pipeline)**. Select **Add**.
6. Select the **Lightning bolt** to trigger continuous deployment and then enable the **Continuous deployment trigger** on the right.

Pipeline Tasks Variables Retention Options History

**Artifacts** | + Add

 \_FabrikamFiber Web-Cl

Schedule not set

**Stages** | + Add ▾

 QA  
1 job, 0 task

**Continuous deployment trigger**  
Build: \_FabrikamFiber Web-Cl

 Enabled  
Creates a release every time a new build is available.

Build branch filters ⓘ  
No filters added.  
+ Add | ▾

**Pull request trigger**  
Build: \_FabrikamFiber Web-Cl

 Disabled

7. Select the **Tasks** tab and select your **QA** stage.
8. Select the plus sign (+) for the job to add a task to the job.
9. On the **Add tasks** dialog box, select **Utility**, locate the **PowerShell** task, and then select its **Add** button.
10. On the left side, select your new **PowerShell** script task.
11. For the **Script Path** argument, select the \*\*\* button to browse your artifacts and select the script you created.
12. Add these **Arguments**:

```
-greeter "$(Release.RequestedFor)" -trigger "$(Build.DefinitionName)"
```

13. On the **Pipeline** tab, select the **QA** stage and select **Clone**.

Pipeline Tasks Variables Retention Options History

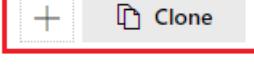
**Artifacts** | + Add

 \_FabrikamFiber Web-Cl

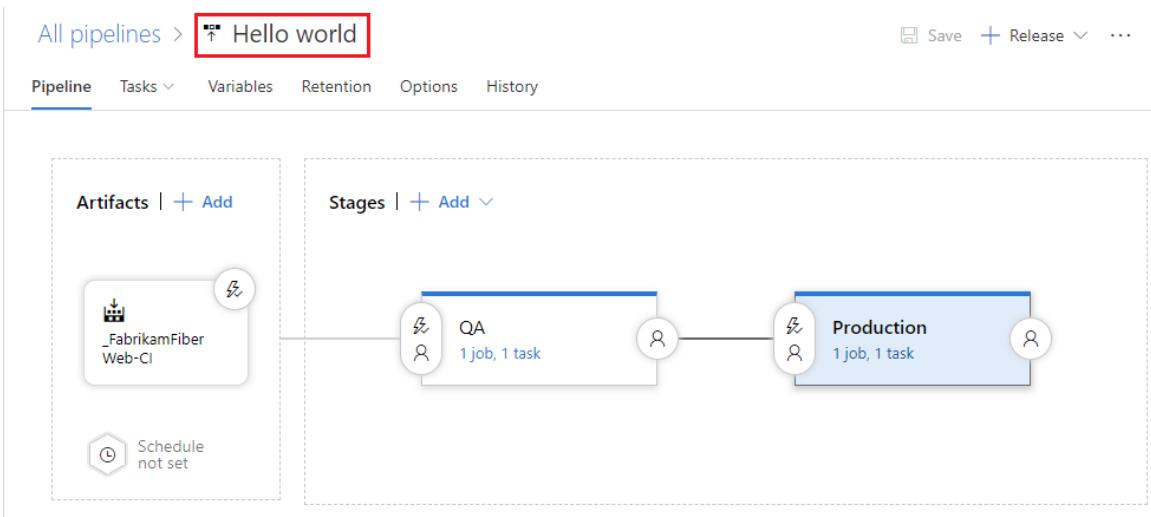
Schedule not set

**Stages** | + Add ▾

 QA  
1 job, 1 task

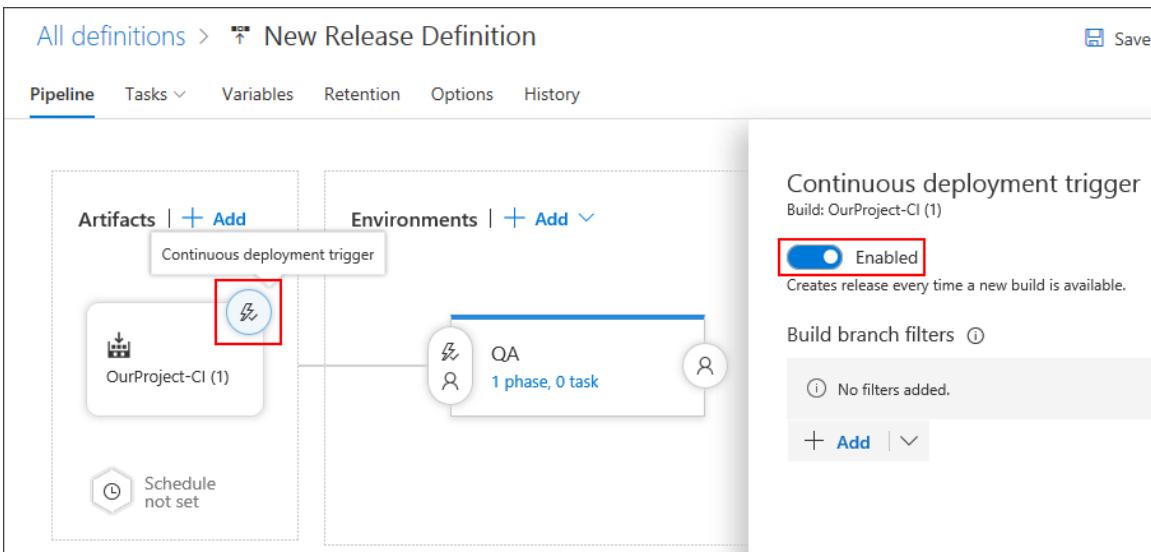


14. Rename the cloned stage **Production**.
15. Rename the release pipeline **Hello world**.



16. Save the release pipeline.

1. Go to the **Build and Release** tab, and then select **Releases**.
2. Select the action to create a **New pipeline**. If a release pipeline is already created, select the plus sign (+) and then select **Create a release definition**.
3. Select the action to start with an **Empty definition**.
4. Name the stage **QA**.
5. In the Artifacts panel, select + **Add** and specify a **Source (Build pipeline)**. Select **Add**.
6. Select the **Lightning bolt** to trigger continuous deployment and then enable the **Continuous deployment trigger** on the right.
  - [TFS 2018.2](#)
  - [TFS 2018 RTM](#)



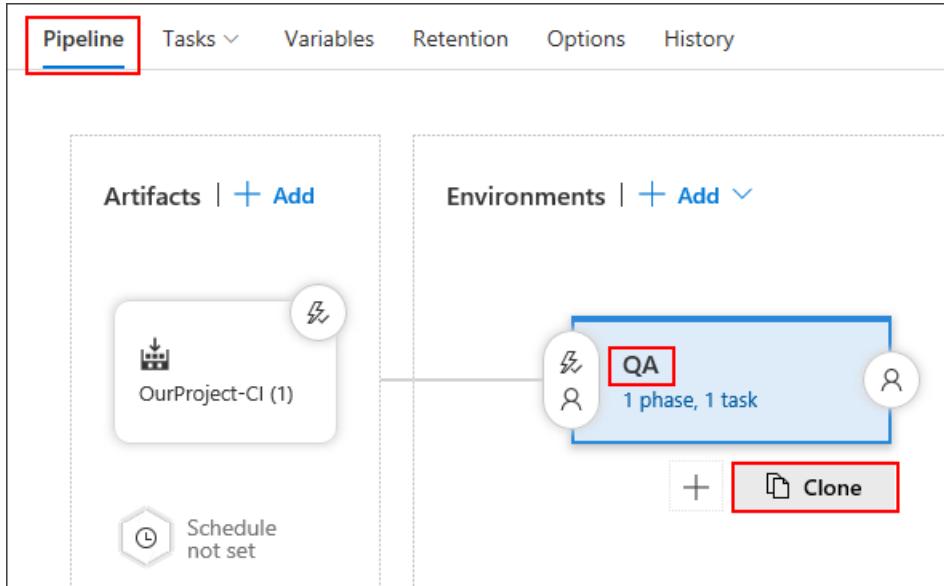
7. Select the **Tasks** tab and select your **QA** stage.
8. Select the plus sign (+) for the job to add a task to the job.
9. On the **Add tasks** dialog box, select **Utility**, locate the **PowerShell** task, and then select its **Add** button.
10. On the left side, select your new **PowerShell** script task.
11. For the **Script Path** argument, select the \*\*\* button to browse your artifacts and select the script you

created.

12. Add these Arguments:

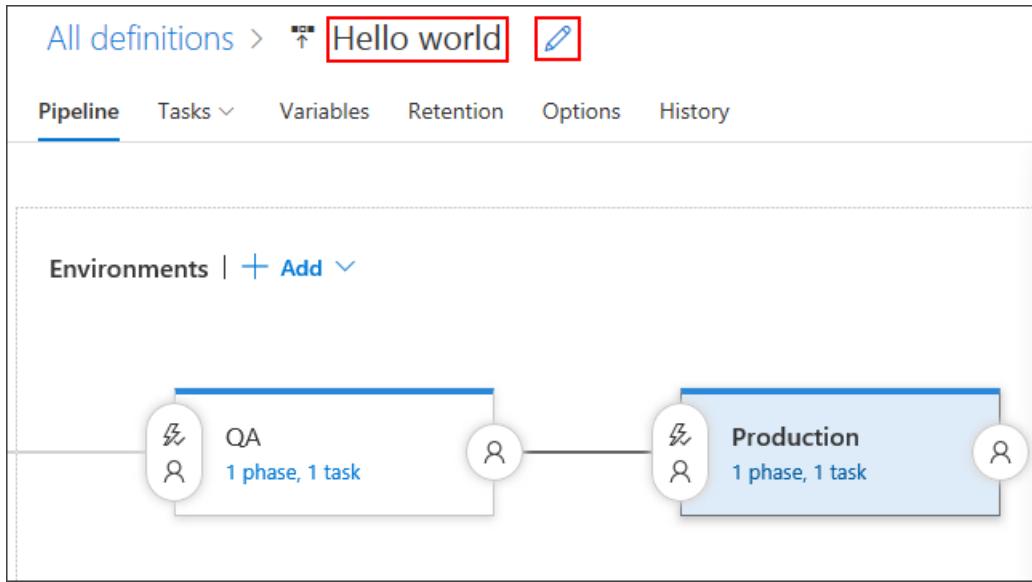
```
-greeter "$(Release.RequestedFor)" -trigger "$(Build.DefinitionName)"
```

13. On the Pipeline tab, select the QA stage and select Clone.



14. Rename the cloned stage Production.

15. Rename the release pipeline Hello world.



16. Save the release pipeline.

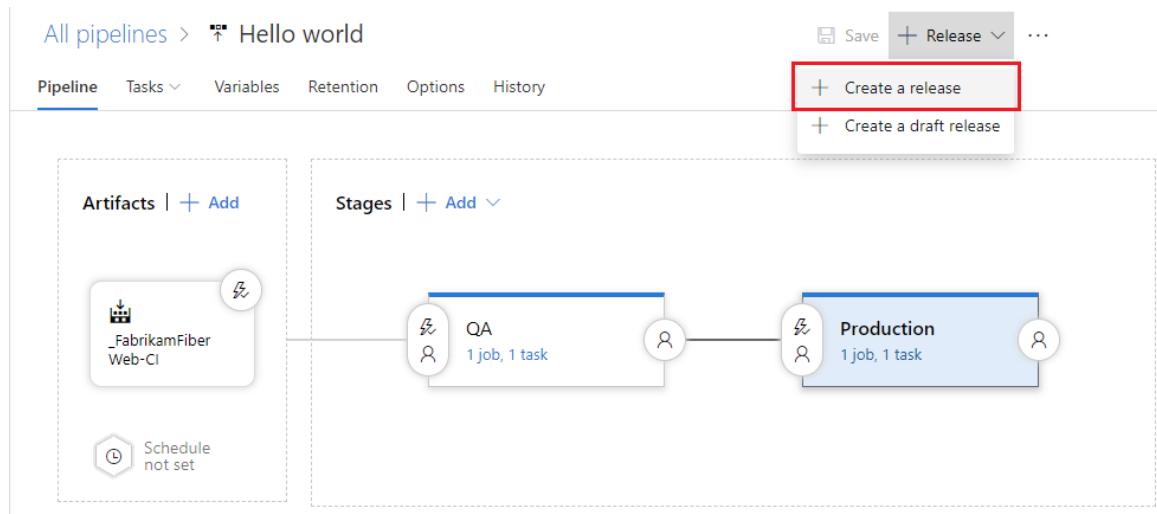
A release pipeline is a collection of stages to which the application build artifacts are deployed. It also defines the actual deployment pipeline for each stage, as well as how the artifacts are promoted from one stage to another.

Also, notice that we used some variables in our script arguments. In this case, we used [release variables](#) instead of the build variables we used for the build pipeline.

## Deploy a release

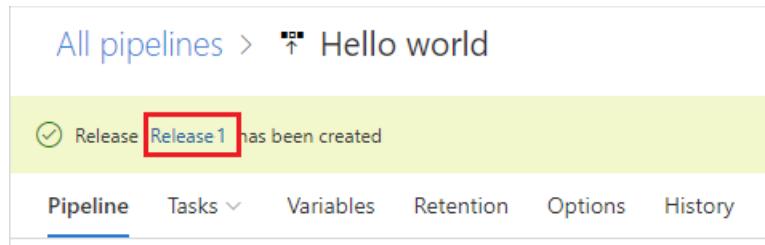
Run the script in each stage.

1. Create a new release.

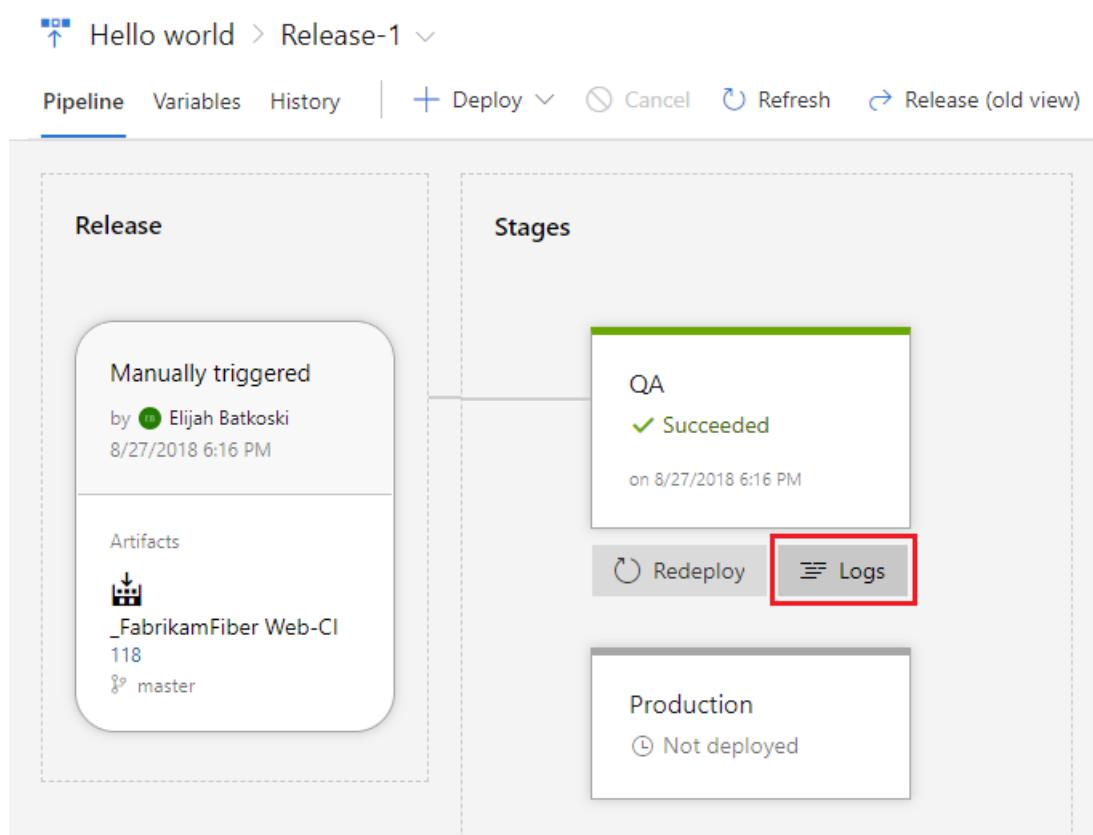


When **Create new release** appears, select **Create**.

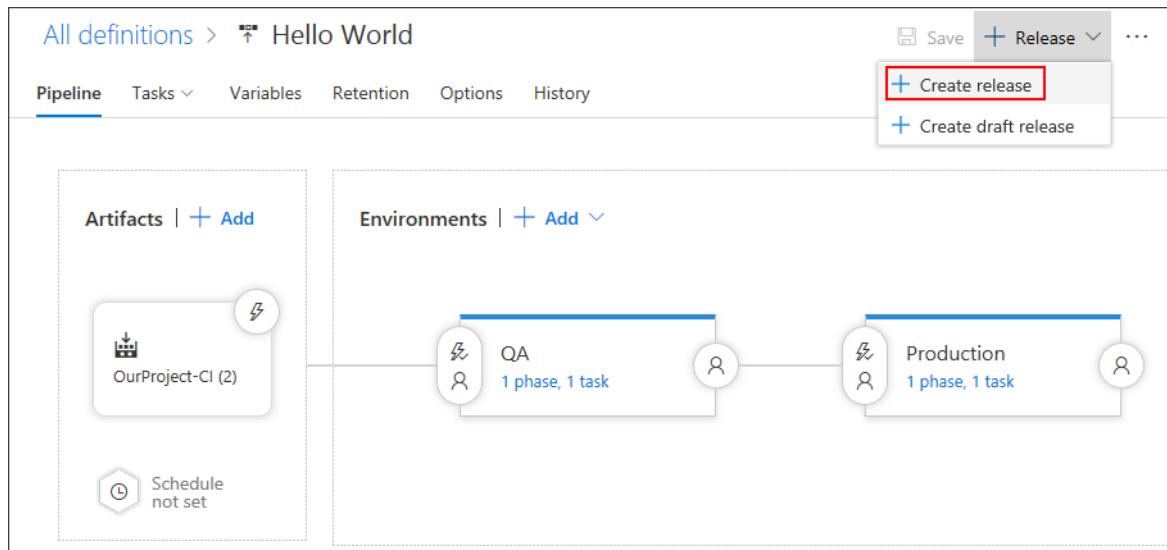
2. Open the release that you created.



3. View the logs to get real-time data about the release.



4. Create a new release.



When **Create new release** appears, select **Create** (TFS 2018.2) or **Queue** (TFS 2018 RTM).

5. Open the release that you created.

The screenshot shows the 'All definitions' page with a green banner indicating that 'Release-2' has been created. Below the banner, the pipeline tab is selected.

6. View the logs to get real-time data about the release.

The screenshot shows the 'Logs' tab for 'Release-2'. It lists three deployment steps: 'QA', 'Production', and 'Production'. The 'QA' step is expanded, showing a 'Pre-deployment approval' task with a checkmark and an 'Agent phase' task. The 'Logs' pane on the right displays the command-line output for the 'Agent phase' task, which includes starting the job, initializing, preparing the release directory, and listing environment variables available.

You can track the progress of each release to see if it has been deployed to all the stages. You can track the commits that are part of each release, the associated work items, and the results of any test runs that you've added to the release pipeline.

## Change your code and watch it automatically deploy to production

We'll make one more change to the script. This time it will automatically build and then get deployed all the way to the production stage.

1. Go to the **Code hub**, **Files** tab, edit the **HelloWorld.ps1** file, and change it as follows:

```

Param(
[string]$greeter,
[string]$trigger
)
Write-Host "Hello world" from $greeter
Write-Host Trigger: $trigger
Write-Host "Now that you've got CI/CD, you can automatically deploy your app every time your team checks in code."

```

2. **Commit** (save) the script.

3. Select the **Builds** tab to see the build queued and run.

4. After the build is completed, select the **Releases** tab, open the new release, and then go to the **Logs**.

Your new code automatically is deployed in the **QA** stage, and then in the **Production** stage.

The screenshot shows the logs for a PowerShell script execution. The output includes the command to run the script, the task description, author information, and the final message: "Hello world from Elijah Batkoski". The last line of the log, which contains the CI/CD deployment message, is highlighted with a red box.

```

1 2018-08-27T18:31:42.7222014Z ##[section]Starting: PowerShell Script
2 2018-08-27T18:31:42.7228660Z =====
3 2018-08-27T18:31:42.7228871Z Task : PowerShell
4 2018-08-27T18:31:42.7229057Z Description : Run a PowerShell script on Windows, macOS, or Linux.
5 2018-08-27T18:31:42.7229253Z Version : 2.136.0
6 2018-08-27T18:31:42.7229424Z Author : Microsoft Corporation
7 2018-08-27T18:31:42.7229609Z Help : [More Information](https://go.microsoft.com/fwlink/?LinkId=613736)
8 2018-08-27T18:31:42.7229827Z =====
9 2018-08-27T18:31:45.5150962Z Generating script.
10 2018-08-27T18:31:45.5196754Z Formatted command: . 'D:\a\r1\a\FabrikamFiber Web-CI\drop\HelloWorld.ps1' -greeter "Elijah Batkoski"
11 2018-08-27T18:31:45.6901750Z ##[command]"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -NoLogo -NoProfile -NonInteractive
12 2018-08-27T18:31:45.9197804Z Hello world from Elijah Batkoski
13 2018-08-27T18:31:45.9203491Z Trigger: FabrikamFiber Web-CI
14 2018-08-27T18:31:45.9208413Z Now that you've got CI/CD, you can automatically deploy your app every time your team checks in code.
15 2018-08-27T18:31:46.0566737Z ##[section]Finishing: PowerShell Script
16 |

```

The screenshot shows the logs for a release pipeline step. The step is named "PowerShell Script" and is part of a "Production" environment. The log output is identical to the one shown in the previous screenshot, including the deployment message, which is highlighted with a red box.

Step	Action	Log Output
> <b>QA</b>	...	Agent queue: Hosted VS2017   Agent: Hosted Agent
<b>Production</b>	...	1 2018-04-25T14:53:54.9742188Z ##[section]Starting: PowerShell Script
<b>Pre-deployment approval</b>	...	2 2018-04-25T14:53:54.9746441Z =====
<b>Agent phase</b>	...	3 2018-04-25T14:53:54.9746591Z Task : PowerShell
<b>Initialize Agent</b>	...	4 2018-04-25T14:53:54.9746697Z Description : Run a PowerShell scri
<b>Initialize Job</b>	...	5 2018-04-25T14:53:54.9746796Z Version : 1.2.3
<b>Download artifact - OurProject-CI (2)</b>	...	6 2018-04-25T14:53:54.9746897Z Author : Microsoft Corporation
<b>PowerShell Script</b>	...	7 2018-04-25T14:53:54.9747808Z Help : [More Information](ht
<b>Post-deployment approval</b>	...	8 2018-04-25T14:53:54.9747921Z =====
		9 2018-04-25T14:53:55.0138848Z ##[command]. 'D:\a\r1\a\OurProject-C
		10 2018-04-25T14:53:57.2485315Z Hello world from Raisa Pokrovskaya
		11 2018-04-25T14:53:57.2486035Z Trigger: Hello world
		12 2018-04-25T14:53:57.2486230Z Now that you've got CI/CD, you can a
		13 2018-04-25T14:53:57.3058801Z ##[section]Finishing: PowerShell Scr
		14

In many cases, you probably would want to edit the release pipeline so that the production deployment happens only after some testing and approvals are in place. See [Approvals and gates overview](#).

## Next steps

You've just learned how to create your first pipeline in Azure. Learn more about configuring pipelines in the language of your choice:

- [.NET Core](#)

- [Go](#)
- [Java](#)
- [Node.js](#)
- [Python](#)
- [Containers](#)

Or, you can proceed to [customize the pipeline](#) you just created.

To run your pipeline in a container, see [Container jobs](#).

For details about building GitHub repositories, see [Build GitHub repositories](#).

To learn how to publish your Pipeline Artifacts, see [Publish Pipeline Artifacts](#).

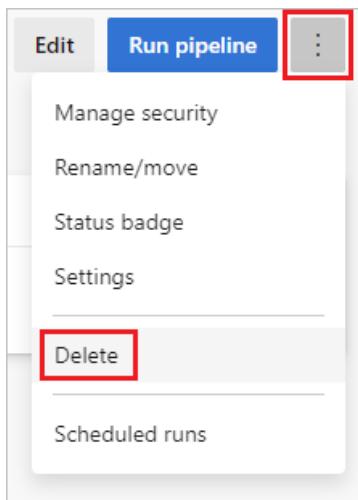
To find out what else you can do in YAML pipelines, see [YAML schema reference](#).

## Clean up

If you created any test pipelines, they are easy to delete when you are done with them.

- [Browser](#)
- [Azure DevOps CLI](#)

To delete a pipeline, navigate to the summary page for that pipeline, and choose **Delete** from the ... menu at the top-right of the page. Type the name of the pipeline to confirm, and choose **Delete**.



You've learned the basics of creating and running a pipeline. Now you're ready to configure your build pipeline for the programming language you're using. Go ahead and create a new build pipeline, and this time, use one of the following templates.

LANGUAGE	TEMPLATE TO USE
.NET	ASP.NET
.NET Core	ASP.NET Core
C++	.NET Desktop
Go	Go
Java	Gradle

LANGUAGE	TEMPLATE TO USE
JavaScript	Node.js
Xcode	Xcode

## FAQ

### Where can I read articles about DevOps and CI/CD?

[What is Continuous Integration?](#)

[What is Continuous Delivery?](#)

[What is DevOps?](#)

### What version control system can I use?

When you're ready to get going with CI/CD for your app, you can use the version control system of your choice:

- Clients
  - [Visual Studio Code for Windows, macOS, and Linux](#)
  - [Visual Studio with Git for Windows or Visual Studio for Mac](#)
  - [Eclipse](#)
  - [Xcode](#)
  - [IntelliJ](#)
  - [Command line](#)
- Services
  - [Azure Pipelines](#)
  - Git service providers such as GitHub and Bitbucket Cloud
  - Subversion
- Clients
  - [Visual Studio Code for Windows, macOS, and Linux](#)
  - [Visual Studio with Git for Windows or Visual Studio for Mac](#)
  - [Visual Studio with TFVC](#)
  - [Eclipse](#)
  - [Xcode](#)
  - [IntelliJ](#)
  - [Command line](#)
- Services
  - [Azure Pipelines](#)
  - Git service providers such as GitHub and Bitbucket Cloud
  - Subversion

### How do I replicate a pipeline?

If your pipeline has a pattern that you want to replicate in other pipelines, clone it, export it, or save it as a template.

Azure DevOps Pipelines interface. On the left, the sidebar shows 'FabrikamFiber Web' selected. In the center, the 'FabrikamFiber Web-Cl' pipeline is listed under the 'master' branch. A context menu is open on the right, with the '...' button highlighted by a red box. Other options in the menu include: Security, Rename/move, Pause builds, Add to my favorites, Clone (highlighted), Save as a template (highlighted), Export (highlighted), Status badge, and Delete.

Azure DevOps Build Definitions interface. The 'Build Definitions' tab is selected. Under the 'All Definitions' tab, the 'HelloWorld-Cl' pipeline is selected. A context menu is open on the right, with the '...' button highlighted by a red box. Other options in the menu include: Queue new build..., Move definition, View definition summary, Edit..., Add to my favorites, Add to team favorites, Clone... (highlighted), Export (highlighted), Rename..., Save as a template... (highlighted), Delete definition, Security..., and Add to dashboard.

After you clone a pipeline, you can make changes and then save it.

After you export a pipeline, you can import it from the All pipelines tab.

After you create a template, your team members can use it to follow the pattern in new pipelines.

#### TIP

If you're using the New Build Editor, then your custom templates are shown at the bottom of the list.

## How do I work with drafts?

If you're editing a build pipeline and you want to test some changes that are not yet ready for production, you can save it as a draft.

The screenshot shows the 'FabrikamFiber Web-CITest' pipeline editor. At the top, there are tabs for Tasks, Variables, Triggers, Options, Retention, and History. Below these, the pipeline structure is shown with a 'Get sources' task. On the right, there's a 'Save & queue' dropdown menu with three options: 'Save & queue', 'Save', and 'Save as draft'. The 'Save as draft' option is highlighted with a red box. To the right of the menu, there's a 'Name \*' field containing 'FabrikamFiber W'.

The screenshot shows the 'Artifacts' pipeline editor. It has tabs for Builds, Releases, Packages, Library, Task Groups, and Deployment Groups\*. Below these, the pipeline structure is shown with a 'Process' task and a 'Get sources' task. On the right, there's a 'Save & queue' dropdown menu with three options: 'Save & queue', 'Save', and 'Save as draft'. The 'Save as draft' option is highlighted with a red box. To the right of the menu, there's a 'Name \*' field containing 'Artifacts'.

You can edit and test your draft as needed.

The screenshot shows the pipeline library. On the left, there's a search bar and a 'New' button. Below it, a list of pipelines includes 'All build pipelines': 'FabrikamFiber Web-Cl' (disabled), 'FabrikamFiber Web-CITest' (selected and highlighted with a red box), and 'PublicWebCl'. On the right, details for the selected pipeline 'FabrikamFiber Web-CITest' are shown, including tabs for History, Analytics, Edit (highlighted with a red box), Queue, and ...

When you're ready, you can publish the draft to merge the changes into your build pipeline.

The screenshot shows the 'FabrikamFiber Web-CITest' pipeline editor again. At the top, there are tabs for Tasks, Variables, Options, History, and a 'Save draft & queue' button. Below these, the pipeline structure is shown with a 'Get sources' task. On the right, there's a 'Publish draft' button highlighted with a red box. To the right of the pipeline editor, there's another screenshot of the 'Artifacts' editor, which also has a 'Save draft & queue' button and a 'Publish draft' button highlighted with a red box.

Or, if you decide to discard the draft, you can delete it from the All Pipeline tab shown above.

### How can I delete a pipeline?

To delete a pipeline, navigate to the summary page for that pipeline, and choose **Delete** from the ... menu in the top-right of the page. Type the name of the pipeline to confirm, and choose **Delete**.

### What else can I do when I queue a build?

You can queue builds [automatically](#) or manually.

When you manually queue a build, you can, for a single run of the build:

- Specify the [pool](#) into which the build goes.
- Add and modify some [variables](#).
- Add [demands](#).
- In a Git repository
  - Build a [branch](#) or a [tag](#).
  - Build a [commit](#).
- In a TFVC repository
  - Specify the source version as a [label](#) or [changeset](#).
  - Run a private build of a [shelveset](#). (You can use this option on either a [Microsoft-hosted agent](#) or a [self-hosted agent](#).)

You can queue builds [automatically](#) or manually.

When you manually queue a build, you can, for a single run of the build:

- Specify the [pool](#) into which the build goes.
- Add and modify some [variables](#).
- Add [demands](#).
- In a Git repository
  - Build a [branch](#) or a [tag](#).
  - Build a [commit](#).

### Where can I learn more about pipeline settings?

To learn more about build pipeline settings, see:

- [Getting sources](#)
- [Tasks](#)
- [Variables](#)
- [Triggers](#)
- [Options](#)
- [Retention](#)
- [History](#)

To learn more about pipeline settings, see:

- [Getting sources](#)

- [Tasks](#)
- [Variables](#)
- [Triggers](#)
- [Retention](#)
- [History](#)

## How do I programmatically create a build pipeline?

[REST API Reference: Create a build pipeline](#)

### NOTE

You can also manage builds and build pipelines from the command line or scripts using the [Azure Pipelines CLI](#).

# Plan and track work in Azure Boards

12/13/2022 • 21 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You track your work by creating work items. This article walks you through creating issues and tasks using a Kanban board. You can learn the Basic process or the Agile process for creating these items.

Choose one of the following four system processes—**Agile**, **Basic**, **Scrum**, or **Capability Maturity Model Integration (CMMI)**—for guidance depending on what process was selected for your project. For an overview of each of these processes, see [Choose a process](#).

## NOTE

The Basic process is available when you add a project to Azure DevOps Services or [Azure DevOps Server 2019 Update 1](#). For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

The Agile process provides several work item types—for example, user stories, tasks, bugs, features, and epics among others—to plan and track work. We recommend you start by adding user stories. If you need to group them into a hierarchy, you can define features. To track other details of work, you can add tasks to a user story.

WORK ITEM TYPES

BACKLOG HIERARCHY

WORK ITEM TYPES	BACKLOG HIERARCHY																																				
<p><b>Portfolio backlog</b></p> <pre> graph TD     Portfolio[Portfolio backlog] --&gt; Epic[Epic]     Portfolio --&gt; Feature1[Feature]     Feature1 --&gt; UserStory1[User Story]     UserStory1 --&gt; Task1[Task]     Product[Product backlog] --&gt; Feature1     Product --&gt; UserStory1     UserStory1 --&gt; Task1     Issues[Issue tracking] --&gt; Issue1[Issue]     subgraph Configurable [Configurable]         Bug[Bug]         Task2[Task]         Bug -.-&gt; Task2     end   </pre> <p><b>Product backlog</b></p> <p><b>Issue tracking</b></p> <p><b>Configurable</b></p>	<table> <thead> <tr> <th>State</th> <th>Title</th> </tr> </thead> <tbody> <tr> <td>● New</td> <td>Web site updates</td> </tr> <tr> <td>● New</td> <td>Web pages</td> </tr> <tr> <td>● New</td> <td>Cancel order form</td> </tr> <tr> <td>● Active</td> <td>Hello World web site</td> </tr> <tr> <td>● Active</td> <td>Change background color</td> </tr> <tr> <td>● New</td> <td>Change page layout</td> </tr> <tr> <td>● New</td> <td>Develop about page</td> </tr> <tr> <td>● New</td> <td>Slow response on form</td> </tr> <tr> <td>● Active</td> <td>Secure Sign-in</td> </tr> <tr> <td>● New</td> <td>Improve User Experience</td> </tr> <tr> <td>● Active</td> <td>Emoticon feedback enabled in client</td> </tr> <tr> <td>● New</td> <td>Service status</td> </tr> <tr> <td>● New</td> <td>Service support</td> </tr> <tr> <td>● New</td> <td>Lookup service outages</td> </tr> <tr> <td>● Resolved</td> <td>Canadian addresses don't display</td> </tr> <tr> <td>● Active</td> <td>Voicemail hang issue</td> </tr> <tr> <td>● Active</td> <td>Check issues with permissions</td> </tr> </tbody> </table>	State	Title	● New	Web site updates	● New	Web pages	● New	Cancel order form	● Active	Hello World web site	● Active	Change background color	● New	Change page layout	● New	Develop about page	● New	Slow response on form	● Active	Secure Sign-in	● New	Improve User Experience	● Active	Emoticon feedback enabled in client	● New	Service status	● New	Service support	● New	Lookup service outages	● Resolved	Canadian addresses don't display	● Active	Voicemail hang issue	● Active	Check issues with permissions
State	Title																																				
● New	Web site updates																																				
● New	Web pages																																				
● New	Cancel order form																																				
● Active	Hello World web site																																				
● Active	Change background color																																				
● New	Change page layout																																				
● New	Develop about page																																				
● New	Slow response on form																																				
● Active	Secure Sign-in																																				
● New	Improve User Experience																																				
● Active	Emoticon feedback enabled in client																																				
● New	Service status																																				
● New	Service support																																				
● New	Lookup service outages																																				
● Resolved	Canadian addresses don't display																																				
● Active	Voicemail hang issue																																				
● Active	Check issues with permissions																																				

Within each work item form, you can describe the work to be done, assign work to project contributors, track status, and collaborate with others through the Discussion section.

Here we show how to add user stories and child tasks from the web portal and add details to those work items.

## Prerequisites

- After you connect to a project, you can add work items. If you don't have a project yet, [create one in Azure DevOps](#).
- To add work items to a board, and use all other board features, you must be granted **Basic** access and have been added as a member of the Contributors or Project Administrators group.
- If you have been granted **Stakeholder** access for a private project and have been added as a member of the Contributors or Project Administrators group, you can view boards, open and modify work items, and add child tasks to a checklist. However, you can't reorder or reparent a backlog item using drag-and-drop, nor update a field on a card.
- If you have been granted **Stakeholder** access for a public project, and have been added as a member of the Contributors or Project Administrators group, you have full access to all Boards features.
- After you connect to a project, you can add work items. If you don't have a project yet, [create one in Azure DevOps](#).
- To add work items to a board, and use all other board features, you must be granted **Basic** access and have been added as a member of the Contributors or Project Administrators group.
- If you have been granted **Stakeholder** access and have been added as a member of the Contributors or Project Administrators group, you can view boards, open and modify work items, and add child tasks to a checklist. However, you can't reorder or reparent a backlog item using drag-and-drop, nor update a field on a card.

#### NOTE

The ability for Stakeholders to drag-and-drop cards to different columns requires installation of Azure DevOps Server 2020.1 update. To learn more, see [Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards](#).

- After you connect to a project, you can add work items. If you don't have a project yet, [create one in Azure DevOps](#).
- To add work items to a board, and use all other board features, you must be granted **Basic** access and have been added as a member of the Contributors or Project Administrators group.
- If you have been granted **Stakeholder** access for a private project and have been added as a member of the Contributors or Project Administrators group, you can view boards, open and modify work items, and add child tasks to a checklist. However, you can't update the status of a backlog item or reorder or reparent a backlog item using drag-and-drop, nor update a field on a card.
- If you have been granted **Stakeholder** access for a public project, and have been added as a member of the Contributors or Project Administrators group, you have full access to all Boards features.

For details, see [Default permissions and access for Azure Boards](#)

#### NOTE

The images shown in this article correspond to the latest version of Azure Boards. While they may differ from those shown in earlier, on-premises versions of Azure DevOps, they are similar in the functions described unless otherwise noted.

## Open your Kanban board

A Kanban board is provisioned with the addition of each project and each team. You can only create or add Kanban boards to a project by adding another team. To learn more, see [About teams and Agile tools](#).

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

The User Stories Kanban board is the best tool for quickly adding user stories and child tasks. To open, choose **Boards > Boards**.

The screenshot shows the Azure DevOps interface with the following elements:

- Header:** Azure DevOps logo and the project name "fabrikam / Fabrikam Fiber" (highlighted with a red box and circled with a red number 1).
- Sidebar:** Navigation menu with "Fabrikam" selected (highlighted with a red box and circled with a red number 2). Other options include "Overview", "Boards", "Work Items", "Backlogs", and "Sprints".
- Team Selection:** "Fabrikam Fiber Team" dropdown (highlighted with a red box and circled with a red number 3).
- Board Header:** "View as backlog" and "Stories" dropdown.
- Columns:** "Backlog", "Active", and "Resolved".
- Card Actions:** "New item" button and search icon.
- Card Data:** Three cards listed:
  - Secure sign-in:** Raisa Pokrovskaya, Priority 1, 0/4 completed.
  - Hello World web site:** Jamal Hartnett, Priority 2, 2/4 completed.
  - Cancel form:** Christie Churchill, Priority 3, 3/4 completed.

The Features Kanban board is the best tool for quickly adding features and user stories that are children of those features. To open the Features board from the Stories board, choose **Features** from the board selector.

Fabrikam Fiber Team ▾ ★ 8

View as backlog

Backlog < Active 8/5 Resolved

+ New item

Secure Sign-in Raisa Pokrovskaya 5

Hello World web site Jamal Hartnett 5 0/4

Cancel order form Christie Church

GPS locator

Epics

Features

Stories

## Add work items to your board

Work items you add to your board are automatically assigned the default **Area Path** and **Iteration Path** assigned to the team. To learn more, see [Configure team settings](#).

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

1. From the Stories board, choose **New item** and start adding those stories you want to track.

New < Active 0/10 Resolved 0/10 Closed <

+ New item

Change initial view

2. Enter return and the system assigns a work item ID to the user story.

New < Active 0/10 Resolved 0/10 Closed <

+ New item

1 Change initial view State New

3. To track the work you want to manage, add as many user stories that you need.

## Add details to a board item

Choose the issue or user story title to open it. Change one or more field values, add a description, or make a note in the **Discussion** section. You can also choose the **Attachments** tab and drag-and-drop a file to share the file with others.

- [Agile process](#)
- [Basic process](#)

- Scrum process
- CMMI process

For example, here we assign the story to Raisa Pokrovskaya and we add a discussion note, at-mentioning Raisa.

Choose Save & Close when done.

## Field descriptions

### Field

### Usage

### Title

Enter a description of 255 characters or less. You can always modify the title later.

### Assigned To

Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu lists only team members or contributors to the project.

#### NOTE

You can only assign work to a single user. If you need to assign work to more than one user, add a work item for each user and distinguish the work to be done by title and description. The Assigned To field only accepts user accounts that have been [added to a project or team](#).

### State

When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current status.

---

#### Reason

Use the default first. Update it when you change state as need. Each State is associated with a default reason.

---

#### Area (Path)

Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting. To change the dropdown list of areas, see [Define area paths and assign to a team](#).

---

#### Iteration (Path)

Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later during a planning meeting. To change the drop-down list of iterations, see [Define iteration paths and configure team iterations](#).

---

#### Description

Provide enough detail to create shared understanding of scope and support estimation efforts. Focus on the user, what they want to accomplish, and why. Don't describe how to develop the product. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

---

#### Acceptance Criteria

Provide the criteria to be met before the work item can be closed. Define what "Done" means by describing the criteria for the team to use to verify whether the backlog item or bug fix is fully implemented. Before work begins, describe the [criteria for customer acceptance](#) as clearly as possible. Have conversations between the team and customers to determine the acceptance criteria. These criteria help ensure a common understanding within the team to meet customers' expectations. Also, this information provides the basis for acceptance testing.

---

#### Priority

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **1:** Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible.
  - **2:** Product cannot ship without the successful resolution of the work item, but it does not need to be addressed immediately.
  - **3:** Resolution of the work item is optional based on resources, time, and risk.
  - **4:** Resolution of the work item is not required.
- 

#### Value Area

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **Architectural:** Technical services to implement business features that deliver solution .
  - **Business:** Services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default).
- 

#### Effort, Story Points, Size

Provide a relative estimate of the amount of work required to complete an issue. Most Agile methods recommend that you set estimates for backlog items based on relative size of work. Such methods include

powers of 2 (1, 2, 4, 8) and the Fibonacci sequence (1, 2, 3, 5, 8, etc.). Use any numeric unit of measurement your team prefers.

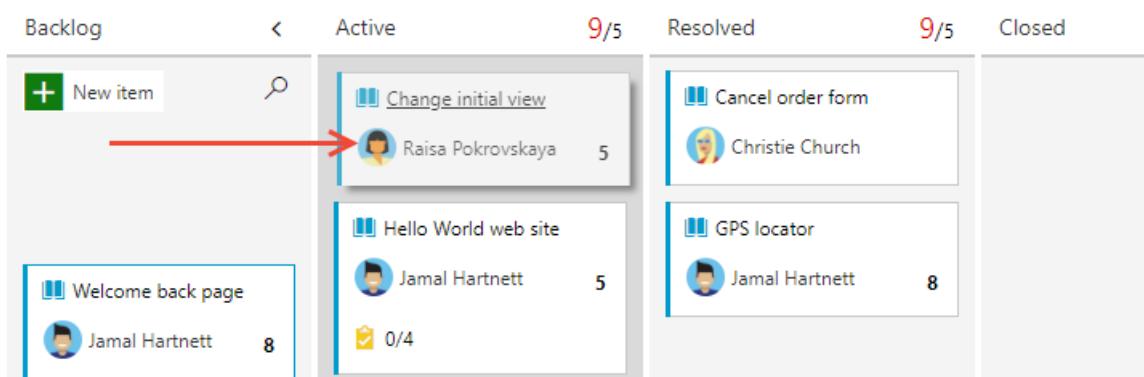
The estimates you set are used to calculate [team velocity](#) and [forecast sprints](#).

## Update work status

The State field tracks the status of a work item. With the Kanban board, you can quickly update the status of backlog items by dragging and dropping them to a different column. This feature requires that you have Basic access or higher.

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

As work starts, drag the user story card from the **Backlog** column to the **Active** column. Once work is ready for review, move to the **Resolved** column. After it's reviewed and accepted, move to the **Closed** column.



You can add or rename columns as needed, see [Customize your board](#).

**TIP**

You can add or rename columns as needed, see [Customize your board](#).

## Add tasks

Task checklists provide a quick and easy way to track elements of work that are important to support completing a backlog item. Also, you can assign individual tasks to different team members.

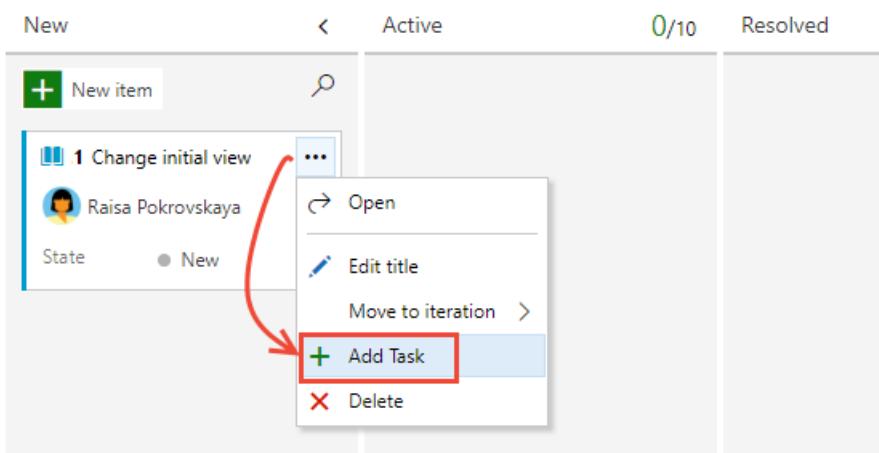
**TIP**

Tasks that you create from the Kanban board are automatically assigned the **Area Path** and **Iteration Path** of their parent work item.

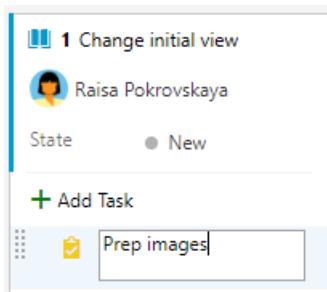
Tasks that you create from the Kanban board show up on your sprint taskboard. Also, tasks that you create from the [sprint backlog](#) or [taskboard](#) show up within tasks checklists on the Kanban board.

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

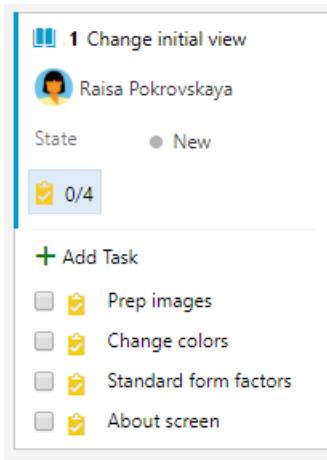
1. To start adding tasks, choose the actions icon for the story and select the Add Task option.



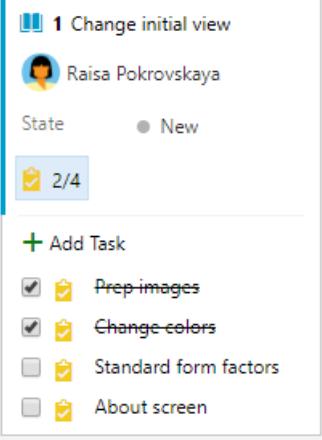
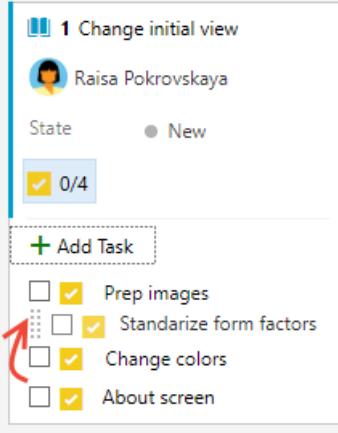
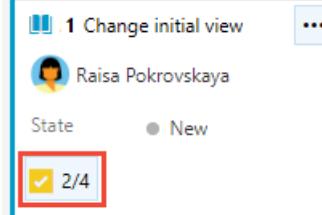
Enter a title for the task and type Enter when done.



2. If you have many tasks to add, keep typing your task titles and type Enter.



3. You can mark a task as done, expand or collapse the task checklist, or reorder and reparent tasks.

MARK A TASK AS DONE	REORDER AND REPARENT TASKS	EXPAND OR COLLAPSE THE CHECKLIST
To mark a task as complete, check the task checkbox. The task State changes to <b>Done</b> .  <p>The screenshot shows a task card for '1 Change initial view' assigned to Raisa Pokrovskaya. The state is 'New'. The checklist shows one item checked: 'Prep images'. Below the checklist is a list of four tasks: 'Prep images', 'Change colors', 'Standard form factors', and 'About screen', each with a checkbox.</p>	To reorder a task, drag it within the checklist. To reparent a the task, drag it to another issue on the board.  <p>The screenshot shows the same task card as above, but the checklist is collapsed, indicated by a dashed border. A red arrow points from the collapsed checklist back up to the main task card area.</p>	To expand or collapse a task checklist, simply choose the task annotation.  <p>The screenshot shows the task card with the checklist expanded. The first item, 'Prep images', is checked. A red box highlights the '2/4' completion status at the bottom of the card.</p>

## Add details to a task

If you have details you want to add about a task, choose the title, to open it. Change one or more field values, add a description, or make a note in the **Discussion** section. Choose **Save & Close** when done.

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

Here we assign the task to Christie Church.

The screenshot shows a task card in the Azure DevOps interface. At the top, it says "TASK 2\*" and has a "Save & Close" button. Below that, there are fields for "State" (New), "Area" (Fabrikam Fiber), "Reason" (New), and "Iteration" (Fabrikam Fiber). A "Details" tab is selected, showing a description: "Prep new images for use on web site." Below the description is a rich text editor toolbar. To the right of the description are sections for "Planning" (Priority 2, Activity Design), "Effort (Hours)" (Original Estimate 8, Remaining 8, Completed 0), "Development", and "Related Work".

## Field descriptions

In addition to the fields you can define for a backlog item—user story, issue, product backlog item, or requirement—you can specify the following fields for a task to support capacity and time tracking.

### NOTE

There are no inherent time units associated with this field even though the taskboard always shows "h" for hours in relationship to Remaining Work. You can specify work in any unit of measurement your team chooses.

## Field

### Usage

### Activity

The type of activity that's required to do a task. To learn more about how this field is used, see [Capacity planning](#).

Allowed values are:

- Deployment
- Design
- Development
- Documentation
- Requirements
- Testing

### Discipline (CMMI process)

The type of activity that's required to do a task. To learn more about how this field is used, see [Capacity planning](#). Allowed values are:

- Analysis
- Development
- Test
- User Education
- User Experience

### Original Estimate

The amount of estimated work required to complete a task. Typically, this field doesn't change after it is assigned.

### Remaining Work

The amount of work that remains to finish a task. You can specify work in hours or in days. As work progresses, update this field. It's used to calculate [capacity charts](#) and the [sprint burndown chart](#).

If you divide a task into subtasks, specify Remaining Work for the subtasks only.

### Completed Work

The amount of work spent implementing a task. Enter a value for this field when you complete the task.

### Task Type (CMMI only)

Select the kind of task to implement from the allowed values:

- Corrective Action
- Mitigation Action
- Planned

## Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.

### Discussion



Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.



Jamal Hartnett commented just now

@Christie Church - Assigning this to you



Christie Church commented less than a minute ago

I've updated the storyboard per our discussions yesterday.



Helena Petersen commented 9 minutes ago

@Christie Church, @Jamal Hartnett - Let's do an A/B test on the colors used in the form.



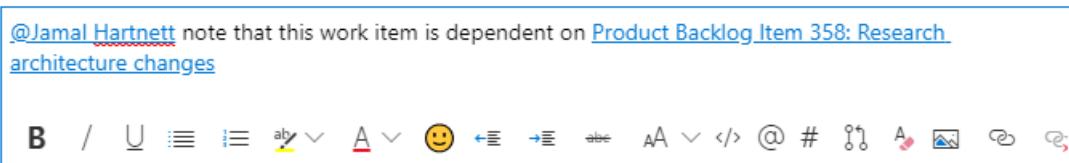
Jamal Hartnett commented 21 hours ago

Make sure the standards guidelines are written in a similar manner to those done for account setup.

---

The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.

## Discussion



@Jamal Hartnett note that this work item is dependent on [Product Backlog Item 358: Research architecture changes](#)

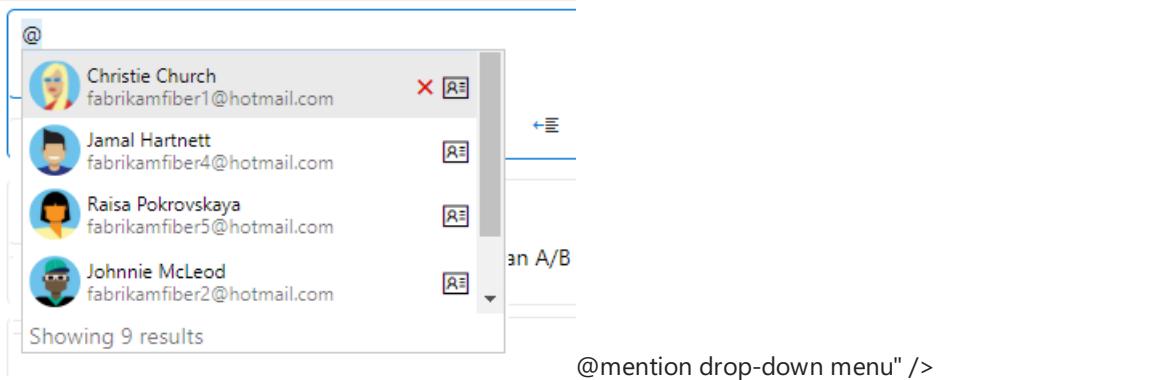
### NOTE

There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the [History field](#). The full content of the text entered into the Discussion text box is added to the History field.

### Mention someone, a group, work item, or pull request

Choose one of these icons —@, #, or !! — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type @, #, or !.

#### Discussion

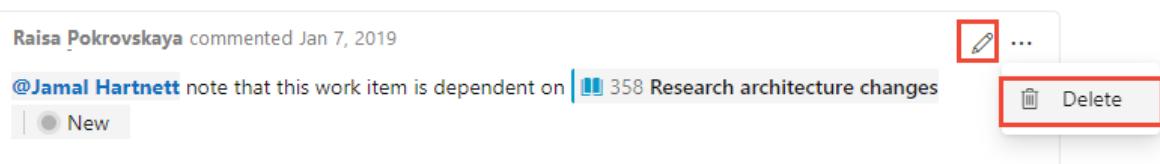


@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing @ and the group name, such as a team or security group.

### Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose  **Edit** or choose the  actions icon and then choose **Delete**.



### NOTE

Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the [@mention control](#) to notify another team member about the discussion. Simply type @ and their name. To reference a work item, use the [#ID control](#). Type # and a list of work items that you've recently

referenced will appear from which you can select.

To reference a work item, use the #ID control. Type # and a list of work items that you've recently referenced will appear from which you can select.

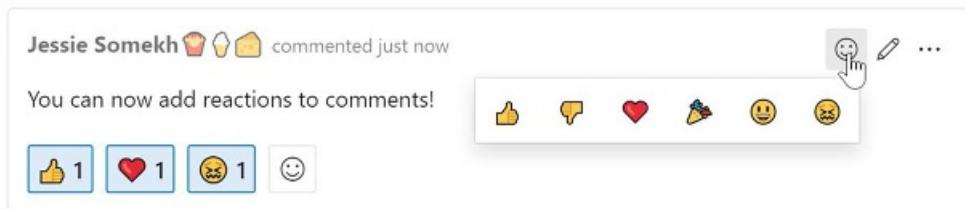
You can't edit or delete comments once you've entered them.

#### IMPORTANT

For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

### Add a reaction to a comment

Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



## Next step

[Customize your board](#)

## Related articles

- [Azure Boards FAQs](#)
- [Index to field descriptions](#)
- [Add tags to issues or tasks](#)

# Add, run, update inline tests

12/13/2022 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Learn how to add, run, update, and expand and collapse inline tests in Azure DevOps.

To start manual testing, add the test to the user story or bug that you want to test. From the Kanban board, you can define inline tests or a set of manual tests for a backlog item. You also can run these tests and update their status. If you're new to working with the Kanban board, see the [Kanban quickstart](#).

Tests you create from the Kanban board are automatically linked to the user story or backlog item.

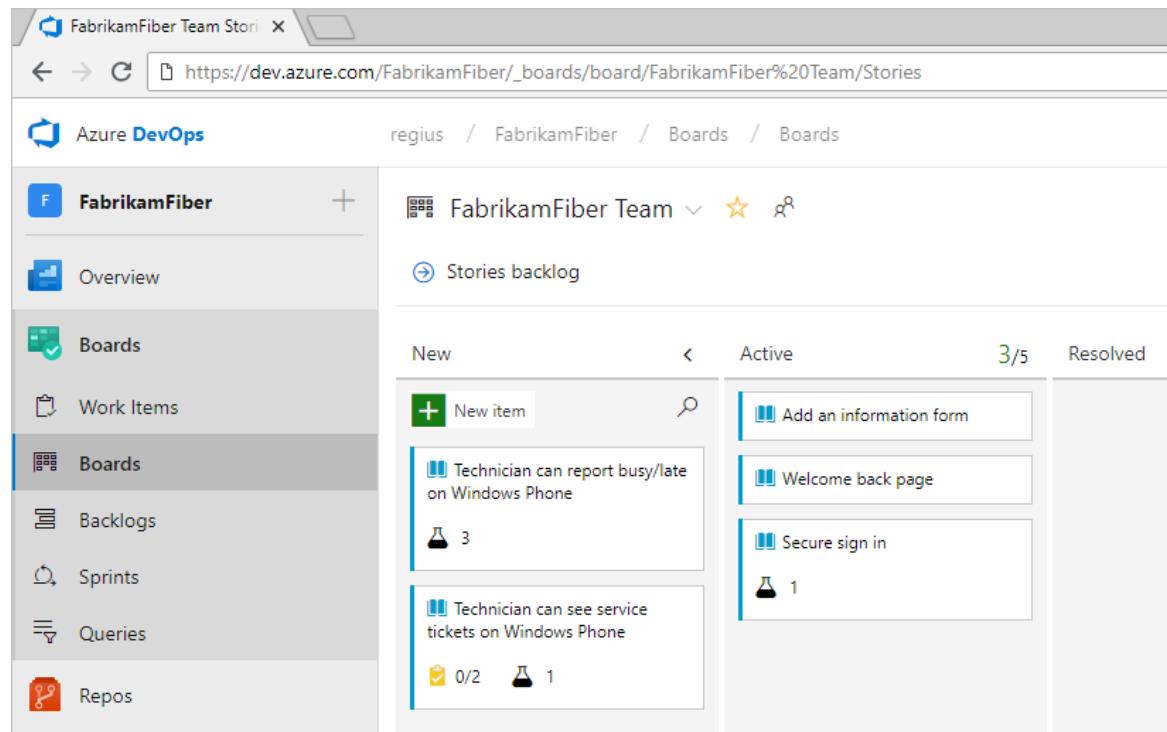
## Open your Kanban board

- From your web browser, open the project for your organization and select **Azure Boards**. If you don't have a project, [create one now](#). If you haven't been added as a team member, [get invited now](#).

The URL follows this pattern: `https://dev.azure.com/fabrikamfiber/_boards/board`

If you don't see the team or project you want, select  [Azure DevOps](#) to [browse all projects and teams](#).

- Select **Boards** to open the Kanban board.



- From your web browser, open the project for your organization and select **Azure Boards**. If you don't have a project, [create one now](#). If you haven't been added as a team member, [get invited now](#).

The URL follows this pattern: `https://dev.azure.com/fabrikamfiber/_backlogs/board`

If you don't see the team or project you want, select  [Azure DevOps](#) to [browse all projects and teams](#).

- Select **Board** to open the Kanban board.

The screenshot shows the 'Backlogs' tab selected in the navigation bar. On the left, a sidebar lists 'Features' and 'Stories' under 'Current'. Under 'Iteration 1', there is one item: 'Technician can report busy/late on Windows Phone' (status: 3). Under 'Iteration 2', there is one item: 'Technician can see service tickets on Windows Phone' (status: 0/2). The main area displays the 'Stories' backlog with two columns: 'New' and 'Active'. The 'Active' column contains three items: 'Add an information form', 'Welcome back page', and 'Secure sign in' (status: 1).

## Add tests

1. To add tests, open the menu for a work item.

The screenshot shows the 'Boards' tab selected in the navigation bar. On the left, a sidebar lists 'Overview', 'Work Items', 'Backlogs', 'Sprints', 'Queries', 'Repos', 'Pipelines', and 'Test Plans'. The 'Backlogs' section is expanded, showing 'Stories backlog'. In the main area, a story titled 'Technician can report busy/late on Windows Phone' is selected. A context menu is open for this item, with the 'Add Test' option highlighted and surrounded by a red box.

Inline tests are the same as test cases in a test suite. A default test plan and test suite automatically get created under which the manual test cases are grouped.

For example, a test suite is created for the following user story, and inline tests are added to that suite. User story 314 is highlighted. It has two manual tests defined with the IDs 337 and 341.

The screenshot shows the 'Test Plans' section of the Visual Studio Test Manager. A test suite named '314 : Technician can see service tickets on Windows Phone' is selected. This suite contains two manual tests with IDs 337 and 341, both marked as 'Active'. The interface includes standard navigation and filtering tools.

Outcome	Order	ID	Title
Active	1	337	Change colors on initial view
Active	2	341	Change initial page size

2. If you have a number of tests to add, enter each title and select **Enter**.

The screenshot shows the 'Add Test' dialog. It lists two inline tests for the selected user story: 'Change colors on initial view' and 'Change initial page size'. There is also a placeholder for another test entry.

To add details to the test case, open it. You can select the title, double-select the inline item, or open the context menu and choose **Open**.

The screenshot shows the details of a test case named "337 Change colors on initial view". The top navigation bar includes "TEST CASE 337", "Jamal Hartnett", "0 comments", "Add tag", "Save & Close", "Follow", and a three-dot menu. Below this, the "State" is "Design" and "Reason" is "New". The "Area" is "FabrikamFiber" and the "Iteration" is "FabrikamFiber\Release 1\Sprint 3". A tabs bar at the bottom includes "Steps" (selected), "Summary", "Associated Automation", a clock icon, a link icon with "(1)", and a clipboard icon.

**Steps**

Click or type here to add a step

**Development**

+ Add link  
Development hasn't started on this item.

**Related Work**

+ Add link ▾  
Tests  
314 Technician can see serv... Updated 2/28/2013, New

**Details**

Priority 2  
Automation status Not Automated

Parameter values

To learn more about how to define tests, see [Create manual tests](#).

Before you run the test, you must add details.

1. To add tests, open the menu for the work item.

The backlog board shows a user story titled "168 Hello World Web Site" by "Jamal Hartnett" with 1/4 completed. A context menu is open over another user story titled "169 Slow response on form". The menu items are: Open, Edit title, Add Task, Add Test (which is highlighted with a blue oval), Delete, Do exploratory testing, and New branch.

Inline tests are the same as test cases in a test suite. A default test plan and test suite automatically get created under which the manual test cases are grouped.

For example, a test suite gets created for each user story, and all inline tests are added to that suite. The following user story 152 is highlighted. It has three manual tests defined with the IDs 153, 155, and 161.

The screenshot shows the 'Test plan' section of a software application. At the top, there are tabs for HOME, CODE, WORK, BUILD, and TEST. Below the tabs, the 'Test plan' tab is selected, followed by 'Parameters', 'Runs', and 'Machines\*'. A dropdown menu shows 'FabrikamFiber: FabrikamFiber Team\_Stories\_FabrikamFiber (Id: 157)'. On the left, a tree view shows 'FabrikamFiber Team\_Stories\_FabrikamFiber' expanded, with three items under it: '151 : Customer log in (2)', '152 : Customer welcome page (3)', and '169 : Slow response on form (1)'. To the right, a table titled 'Test suite: 152: Customer welcome pa...' shows three tests:

Outcome	ID	Title
Active	153	Change colors on initial view
Active	155	Change initial page size
Active	161	Log in with email

To learn more about test plans and test suites, see [Plan your tests](#).

2. If you have a number of tests to add, enter each title and select **Enter**.

The screenshot shows a modal window for a specific test case. The title is '152 Customer welcome page'. It displays a user profile for 'Raisa Pokrovskaya'. Below the profile, there are two progress bars: one yellow labeled '0/3' and one orange labeled '0/3'. A button 'Add Test' is present. A list of three test cases is shown, each preceded by a blue circular icon:

- Change colors on initial view
- Change initial page size
- Log in with email

To add details to the test case, open it. You can select the title, double-select the inline item, or open the context menu and choose **Open**.

TEST CASE 153

### 153 Change colors on initial view

Design Raisa Pokrovskaya 0

Area Iteration  
Fabrikam Fiber Fabrikam Fiber\Iteration 1

Add Tag Steps Summary Associated Automation (1)

**Steps**

Action Click or type here to add a step

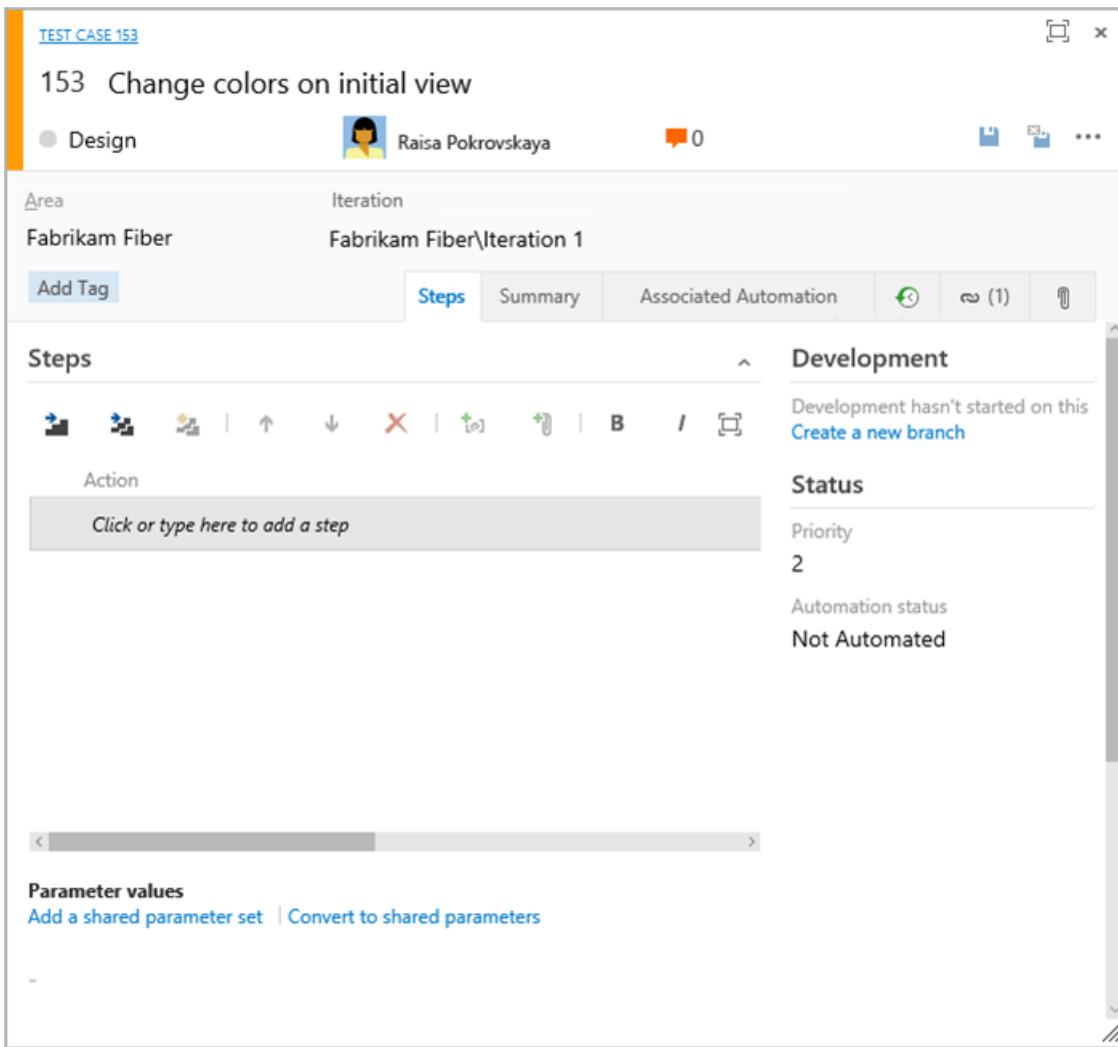
**Development**

Development hasn't started on this  
[Create a new branch](#)

**Status**

Priority 2  
Automation status Not Automated

Parameter values  
[Add a shared parameter set](#) | [Convert to shared parameters](#)

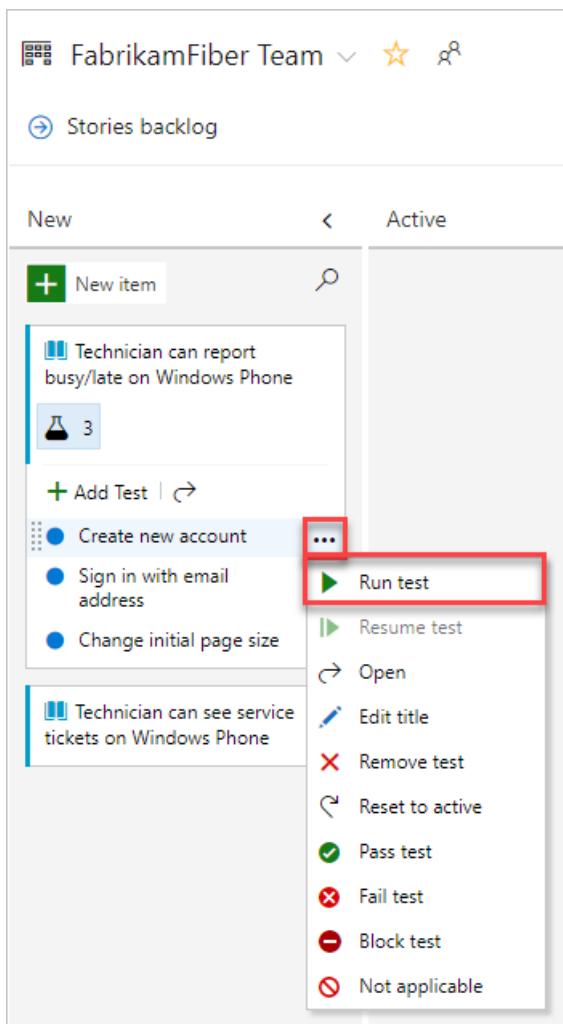
The screenshot shows a Microsoft Test Case management interface. At the top, there's a header with 'TEST CASE 153' and a title '153 Change colors on initial view'. Below the title, it shows 'Design' and the name 'Raisa Pokrovskaya' with a '0' notifications icon. There are tabs for 'Area' (Fabrikam Fiber) and 'Iteration' (Fabrikam Fiber\Iteration 1). A toolbar below the tabs includes 'Add Tag', 'Steps' (which is selected), 'Summary', 'Associated Automation', and three more icons. The main area is titled 'Steps' and contains a placeholder 'Click or type here to add a step'. To the right, there's a 'Development' section with a note that development hasn't started and a link to 'Create a new branch'. Below that is a 'Status' section with priority set to 2 and automation status as 'Not Automated'. At the bottom, there's a 'Parameter values' section with links to add or convert shared parameter sets.

To learn more about how to define tests, see [Create manual tests](#).

Before you run the test, you must add details.

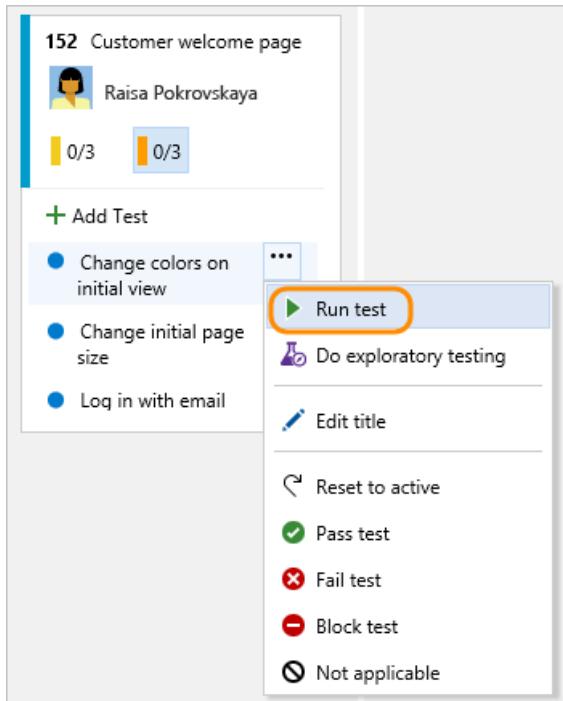
## Run a test

Run the test by selecting **Run test** from the actions menu for the inline test.



Microsoft Test Runner starts in a new browser instance. For information on how to run a test, see [Run manual tests](#).

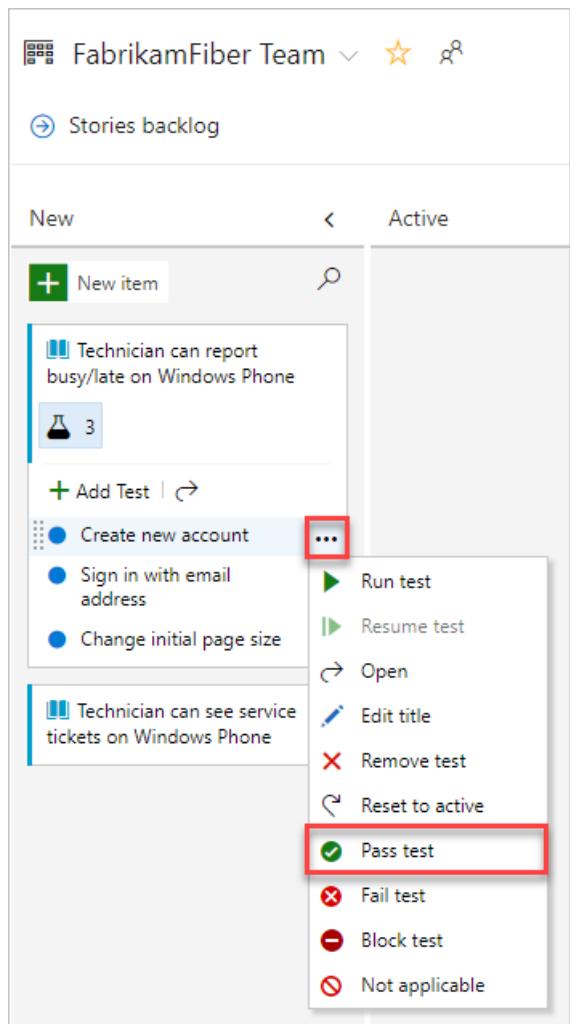
Run the test by selecting ► Run test from the ... actions menu for the inline test.



Microsoft Test Runner starts in a new browser instance. For information on how to run a test, see [Run manual tests](#).

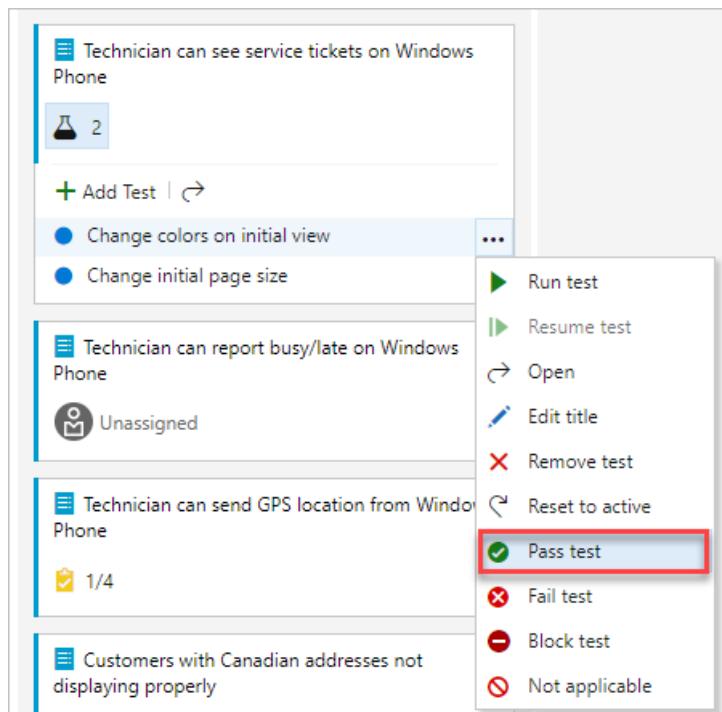
## Update the status of a test

You can update the status of the test from the \*\*\* actions menu.



When you update the status of tests, you can [track test results](#).

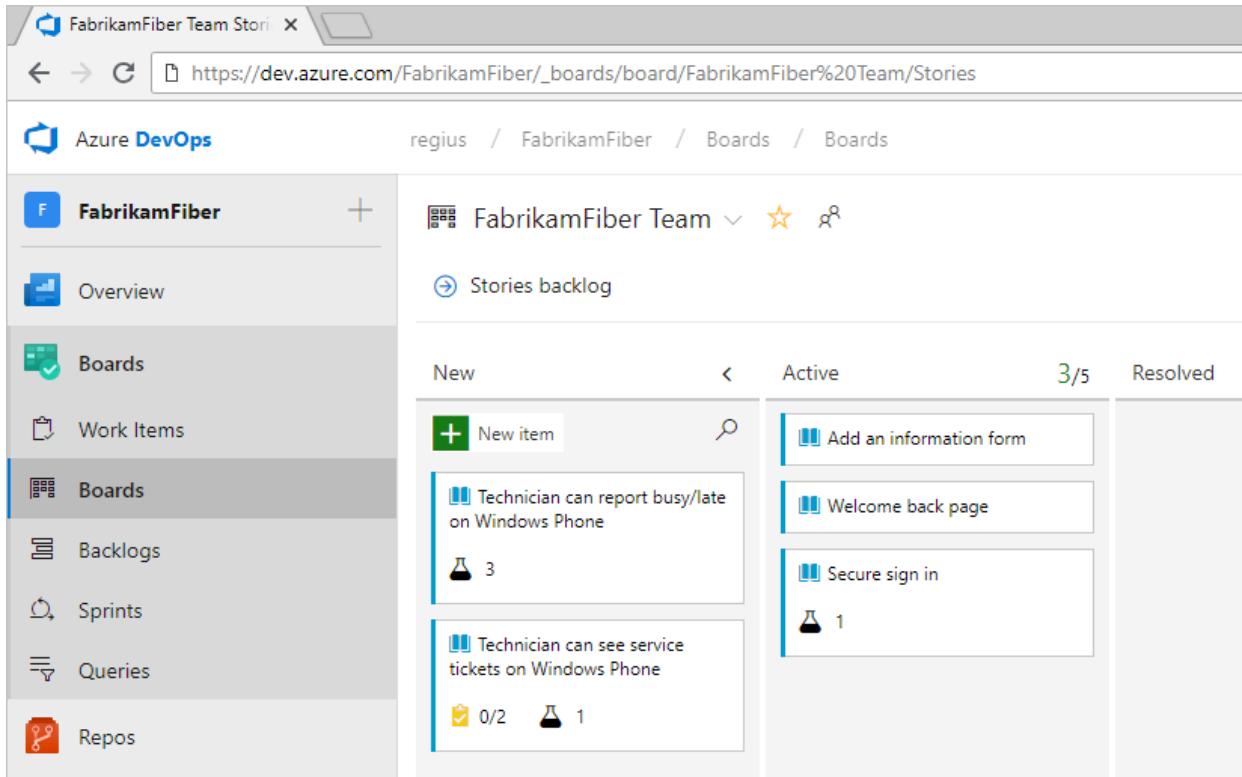
You can update the status of the test from the \*\*\* actions menu.



When you update the status of tests, you can [track test results](#).

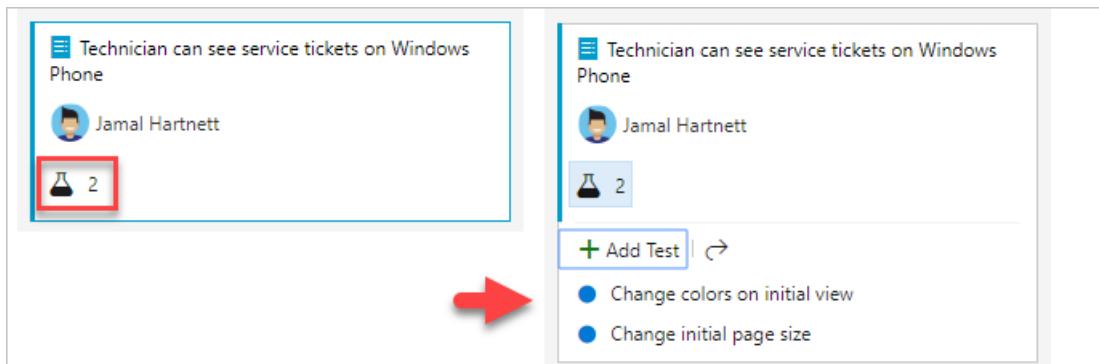
## Expand or collapse inline tests

When you first open the Kanban board, you'll see an unexpanded view of checklists and tests.



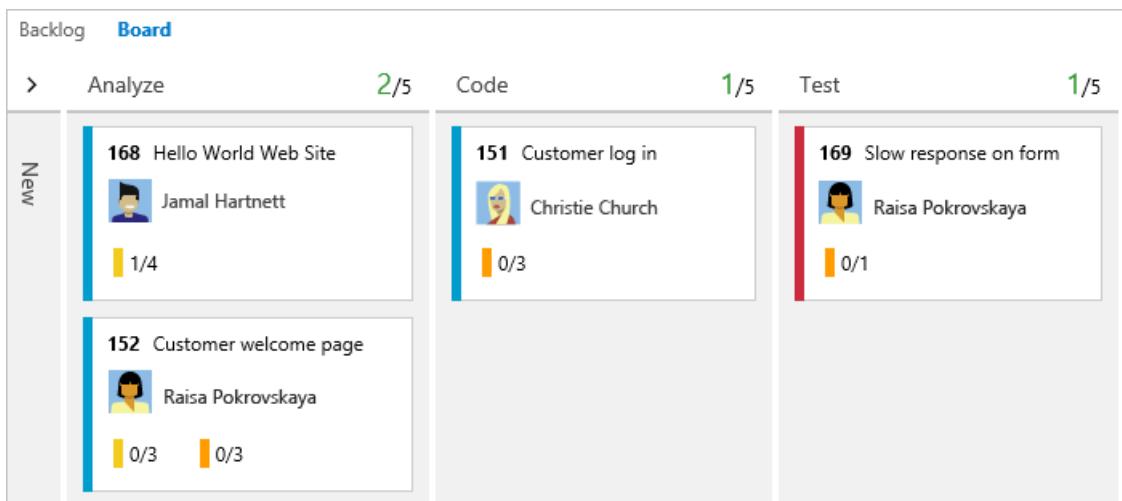
The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' team. The left sidebar has 'Boards' selected. The main area displays a backlog of stories. One specific story, 'Technician can see service tickets on Windows Phone', is highlighted. Its checklist is expanded, showing 2 items. A red arrow points from the collapsed state on the left to the expanded state on the right.

Select the inline test summary to expand a collapsed set of tests. Select the same summary to collapse an expanded list.



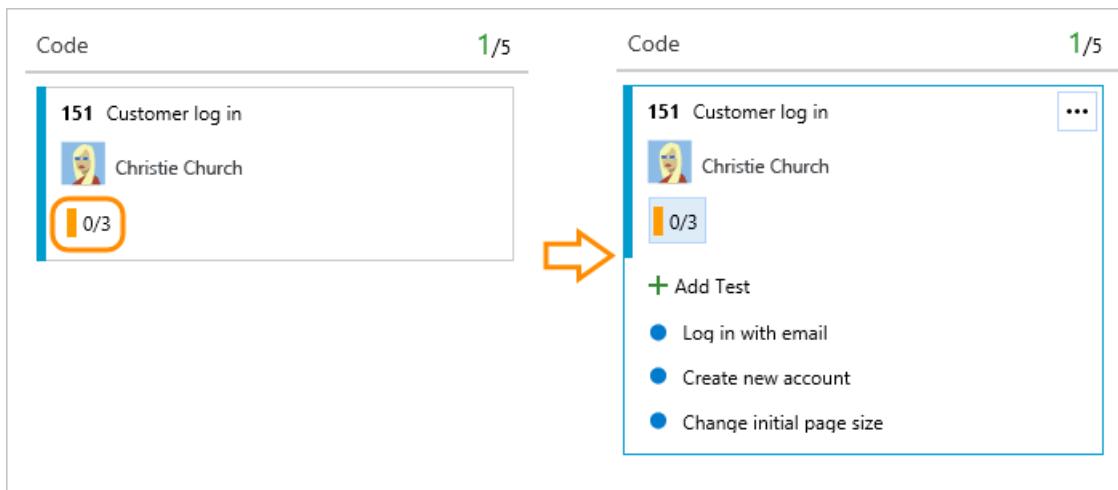
The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' team. The left sidebar has 'Boards' selected. The main area displays a backlog of stories. One specific story, 'Technician can see service tickets on Windows Phone', is highlighted. Its checklist is expanded, showing 2 items. A red arrow points from the collapsed state on the left to the expanded state on the right.

When you first open the Kanban board, you'll see an unexpanded view of checklists.



The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' team. The left sidebar has 'Boards' selected. The main area displays a backlog of stories. Three stories are visible: 'Hello World Web Site', 'Customer log in', and 'Slow response on form'. Each story has its checklist collapsed, indicated by a small number in a box.

Select the inline test summary to expand a collapsed set of tests. Select the same summary to collapse an expanded list.



## Next steps

[Kanban quickstart](#)

## Related articles

- [Learn more about test case management](#)
- [Exploratory test your web app directly in your browser](#)
- [Essential services](#)
- [Client-server tools](#)
- [Software development roles](#)

# Tutorial: Follow changes made to a user story, bug, or other work item or pull request

12/13/2022 • 5 minutes to read • [Edit Online](#)

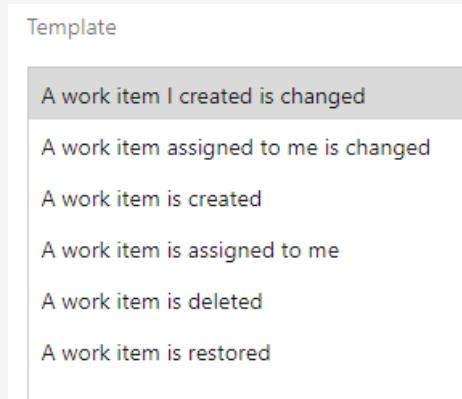
Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To get notified of changes made to a specific work item or a pull request, you can choose to follow them. The Follow feature provides an improvised way of getting notified on a case-by-case basis.

If you want to subscribe to receive notifications automatically based on changes that occur based on your targeted set of criteria, see [Manage personal notifications](#). For example, you can create a subscription to automatically get notified whenever a work item that you created or that was assigned to you is modified.

## NOTE

Notification subscriptions allow you to personalize the notifications you receive automatically based on additional criteria you specify for [yourself](#), a team, or a project. For example, you can create a subscription and add field criteria to receive changes based on one or more of the following templates.



This article shows you how to:

- Follow a work item
- Follow a pull request
- Manage work items that you're following

[Configure an SMTP server](#) in order for team members to receive notifications.

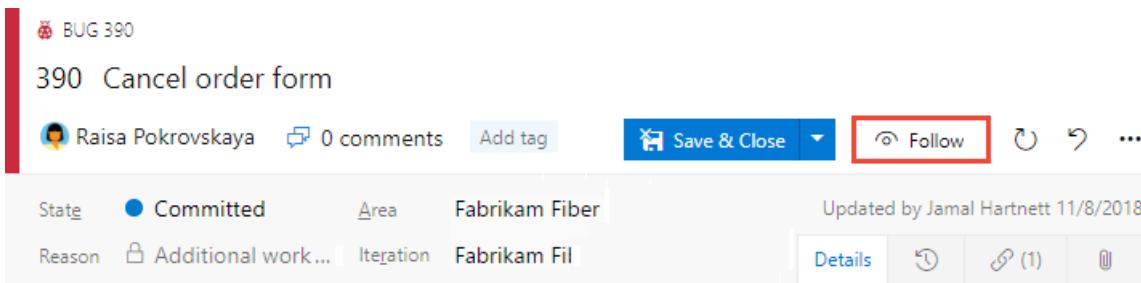
## Prerequisites

- Connect to a project. If you don't have a project yet, [create one](#).
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, [Add users to a project or team](#).
- To view or follow work items, you must be granted Stakeholder access or higher. For details, see [About access levels](#). Also, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see [Set permissions and access for work tracking](#).
- To view or follow pull requests, you must have **Basic** access or higher.

- You must connect to a project. If you don't have a project yet, [create one](#).
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, [Add users to a project or team](#).
- To view or follow work items, you must be granted **Stakeholder** access or higher. For details, see [About access levels](#). Also, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see [Set permissions and access for work tracking](#).
- To view or follow pull requests, you must have **Basic** access or higher.

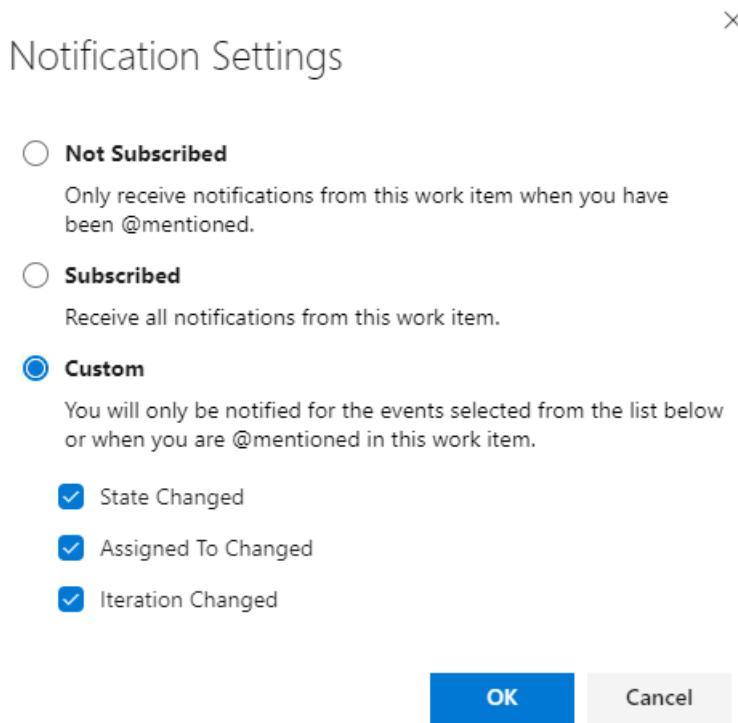
## Follow a work item

When you want to track the progress of a single work item, choose the  [Follow](#) follow icon. This signals the system to notify you when changes are made to the work item.



A screenshot of a work item details page for a bug titled "BUG 390". The page shows basic information like the assignee (Raisa Pokrovskaya), comments (0), and tags (Add tag). A prominent red box highlights the "Follow" button, which is located next to the "Save & Close" button. Below the main header, there's a summary bar with fields for State (Committed), Area (Fabrikam Fiber), Reason (Additional work...), Iteration (Fabrikam Fil), and last updated by (Jamal Hartnett on 11/8/2018). At the bottom right of the summary bar, there are buttons for Details, a timer icon, a link icon, and a refresh icon.

If you want to specify conditions on when you'll get notified of changes, choose the  gear icon and choose from the options provided.



The screenshot shows the "Notification Settings" dialog box. It contains three radio button options: "Not Subscribed" (unchecked), "Subscribed" (unchecked), and "Custom" (checked). The "Custom" section includes a list of checked notification types: "State Changed", "Assigned To Changed", and "Iteration Changed". At the bottom of the dialog are "OK" and "Cancel" buttons.

By default, you're **Subscribed** to receive a notification when any change is made to the work item. Choose **Not Subscribed** to receive notification only when you're @mentioned. Or choose **Custom** to receive notifications when one of the checked fields changes, **State**, **Assigned To**, or **Iteration Path**.

BUG 390

## 390 Cancel order form



Raisa Pokrovskaya

0 comments

Add tag

Save & Close

Follow



State

Committed

Area

Fabrikam Fiber

Updated by Jamal Hartnett 11/8/2018

Reason

Additional work...

Iteration

Fabrikam Fil

Details



(1)



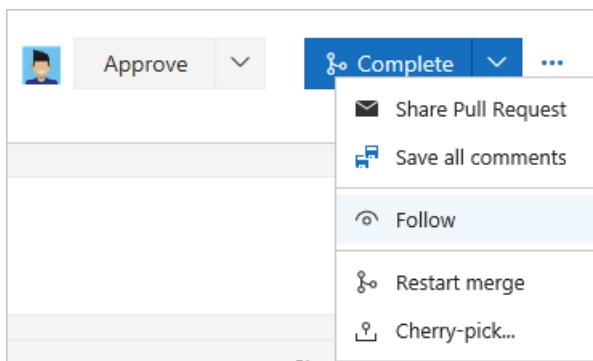
You'll only receive notifications when other members of your team modify the work item, such as adding to the discussion, changing a field value, or adding an attachment.

Notifications are sent to your preferred email address, which [you can change from your user profile](#).

To stop following changes, choose the following icon.

## Follow a pull request

To track the progress of a single pull request, choose the actions icon for the pull request, and select the Follow option. This signals the system to notify you when changes are made to the PR.



You'll only receive notifications when other members of your team modify the PR, such as adding to the discussion or adding an attachment.

Notifications are sent to your preferred email address, which [you can change from your user profile](#).

To stop following changes, open the PR context menu and choose the Following icon.

## Manage work items that you're following

You can review and manage all the work items you've selected to follow.

Open **Boards>Queries**, choose **All**, and under **My Queries**, choose **Followed work items**.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. On the left, the navigation bar has 'Queries' selected, indicated by a red box. The top navigation bar shows 'Fabrikam Fiber' and a '+' icon. Below the navigation bar are links for 'Overview', 'Boards', 'Work Items', 'Backlogs', 'Sprints', 'Plans', and 'Code'. The 'Queries' link is also highlighted with a red box. The main area is titled 'Queries' and shows a list of queries under 'My Queries'. The 'All' tab is selected, also highlighted with a red box. The list includes: 'Active bugs' (starred), 'All Items' (starred), 'Assigned to me' (starred), 'Closed bugs', 'Fabrikam Fiber Team - Backlog items', 'Followed work items' (highlighted with a red box), and 'Following - my query'.

From this view, you can view all items you're following across all projects. Also, you can complete similar actions supported with a query results view, such as:

- Refresh the view
- Add or remove visible columns
- Sort the order of specific columns
- Filter results by text or tags
- Set work item pane
- Enter full screen mode.

You can also view and manage work that you're following from **Boards>Work Items** and pivot to **Following**.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. On the left, the navigation bar has 'Work Items' selected, indicated by a red box. The top navigation bar shows 'Azure DevOps' and 'fabrikam / Fabrikam Fiber'. Below the navigation bar are links for 'Overview', 'Boards', 'Work Items', 'Backlogs', 'Sprints', and 'Queries'. The 'Work Items' link is also highlighted with a red box. The main area is titled 'Work Items' and shows a table of work items under the 'Following' dropdown menu, which is highlighted with a red box. The table has columns for ID, Assigned To, State, and Title. The data is as follows:

ID	Assigned To	State	Title
375	Jamal Hartnett	● Committed	Check service status
361	Christie Church	● Approved	Interim save on long form
384	Christie Church	● Committed	Secure sign-in
360	Raisa Pokrovskaya	● New	Change initial view
436	Jamal Hartnett	● Committed	Hello World Web Site

Open **Work>Queries** and choose **Followed work items**.

The screenshot shows the 'Followed work items' section in the 'Queries' view of Team Services. On the left, there's a sidebar with sections like 'Assigned to me', 'Followed work items' (which is selected and highlighted with a blue border), and 'Unsaved work items'. Below these are sections for 'My favorites' and 'Team favorites', each with a placeholder for dragging queries. On the right, the main area is titled 'Followed work items' and 'Results'. It includes a toolbar with icons for 'Save query', 'Refresh', 'Run', 'Edit', 'Copy', 'Link', and 'Email', followed by 'Column options'. A table lists three work items: a Bug titled 'Slow response on form' (Resolved), a User Story titled 'Cancel order form' (Active), and another User Story titled 'Welcome page' (Active). The table columns are 'ID', 'Work Item Type', 'Title', and 'State'.

From this view, you can view all items you're following across all projects. Also, you can complete similar actions supported with a query results view, such as:

- Refresh the view
- Add or remove visible columns
- Sort the order of specific columns
- Filter results by text or tags
- Set work item pane
- Enter full screen mode.

You can also view and manage work that you're following from your Project pages. To learn more, see [Work across projects](#).

## Query work items that you're following

You can use the `@Follows` macro in a work item query to filter a list based on work items you're following along with other query filters.

For example, the following query shows how to query across all projects for active work items that you're following. You use the `ID` field and the `In` operator with the `@Follows` macro.

The screenshot shows the 'My Queries' query editor. At the top, it says '3 work items' and '1 selected'. Below that is a toolbar with 'Results' (selected), 'Editor', 'Charts', 'GANTT', 'Export', 'Run query', and a 'More' button. The main area starts with 'Type of query' set to 'Flat list of work items' and 'Query across projects' checked. Below this is a 'Filters for top level work items' section with a table. The table has columns for 'And/Or', 'Field\*', 'Operator', and 'Value'. There are three clauses: 1) 'Work Item Type' is set to '=' and '[Any]'. 2) 'State' is set to '=' and 'Active'. 3) 'ID' is set to 'In' and '@Follows'. At the bottom of the filter table is a link 'Add new clause'.

## Try this next

[Add, update, and follow a work item](#)

## Related articles

- [Manage personal notifications](#)
- [View and update work items via the mobile work item form](#)

### **Q: Can I add someone else to follow a work item or PR?**

A: No, you can't add another team member to follow a work item or pull request at this time. You can subscribe them to get notified based on select criteria, such as when a work item is created or modified, or a pull request is created. For details, see [Manage team notifications](#).

# Get started as a Stakeholder

12/13/2022 • 16 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Stakeholders are users with free but limited access to Azure DevOps features and functions. With Stakeholder access, you can add and modify work items, manage build and release pipelines, and view dashboards. You can check project status and provide direction, feedback, feature ideas, and business alignment to a team. For a quick overview of the features available to Stakeholders, see [Stakeholder access quick reference](#).

## NOTE

For public projects, Stakeholder access gives users greater access to features. To learn more, see [Default roles and access for public projects](#). For information about working with pipelines, see these articles: [Build your GitHub repository](#) and [Build OSS repositories](#).

Stakeholders are users with free but limited access to Azure DevOps features and functions. With Stakeholder access, you can add and modify work items, view and approve pipelines, and view dashboards. You can check project status and provide direction, feedback, feature ideas, and business alignment to a team.

Stakeholder access is one of several supported access levels as described in [Stakeholder access quick reference](#). To get access as a Stakeholder, ask your organization owner or Project Collection Administrator to [add you to a project with Stakeholder access](#).

Stakeholder access is one of several supported access levels as described in [Stakeholder access quick reference](#). To get access as a Stakeholder, ask your server administrator to [add you to a security group that has Stakeholder access](#).

## NOTE

Azure Boards supports several Agile methods such as Kanban and Scrum. Depending on what methods your team uses, you'll want to become familiar with other tools that Azure Boards supports. This article focuses on getting familiar with work items and the Kanban board. For additional information, see [Related articles](#) at the end of this article.

Use this tutorial to learn how to do the following tasks:

- Sign in to a project
- Understand which work item types are available to your project
- Open the Kanban board and open a work item
- Add details, tags, or comments to a work item
- View the product backlog
- Find work assigned to you, or query for other work items
- Understand what features are and aren't available to users with Stakeholder access

## Connect to the web portal of a project

You must have been added to the Azure DevOps project and been granted Stakeholder or higher access level.

1. Choose the link provided in the email invitation you should have received. Or, open a browser window and enter the URL for the web portal.

<https://dev.azure.com/OrganizationName/ProjectName>

<http://ServerName:8080/tfs/DefaultCollection/ProjectName> For example, to connect to the server named *FabrikamPrime* and project named *Contoso*, enter  
<http://FabrikamPrime:8080/tfs/DefaultCollection/Contoso>.

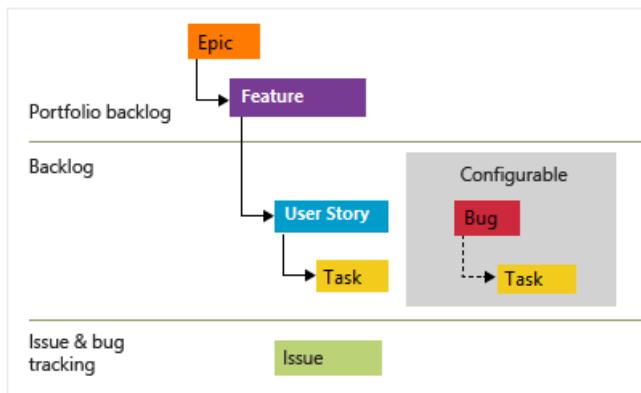
2. Enter your credentials. If you can't sign in, ask the organization owner or Project Administrator to add you as a member of the project with Stakeholder access.

## Understand work items and work item types

Work items support planning and tracking work. Each work item represents an object stored in the work item data store. Each work item is based on a work item type and is assigned an identifier which is unique within an organization or project collection. Different work items are used to track different types of work as described in [About work items](#). The work item types available to you are based on the [process used when your project was created](#)—Agile, Basic, Scrum, or CMMI—as illustrated in the following images.

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

The following image shows the Agile process backlog work item hierarchy. User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Each team can configure how they manage Bugs—at the same level as User Stories or Tasks—by configuring the [Working with bugs](#) setting. To learn more about using these work item types, see [Agile process](#).

## Open your Kanban board from the web portal

You can start viewing work items once you connect to a project.

1. Check that you selected the right project, and select **Boards > Boards**. Then select the correct team from the team selector menu.

To select another team's board, open the selector. Then select a different team, or select the [\*\*Browse all team boards\*\*](#) option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

**TIP**

Select the  star icon to make a team board a favorite. Favorite artifacts ( favorite icon) appear at the top of the team selector list.

2. Check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level.

1. Check that you selected the right project, and select **Boards > Boards**. Then select the correct team from the team selector menu.

To select another team's board, open the selector. Then select a different team, or select the  **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

**TIP**

Select the  star icon to make a team board a favorite. Favorite artifacts ( favorite icon) appear at the top of the team selector list.

2. Check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level. Here we have selected **Backlog Items** for the Scrum process.

1. To view your Kanban board, open your project from a web browser. Select **Work > Backlogs > Stories**, and then select **Board**.

If you don't see **Work**, your screen size might be reduced. Select the three dots () icon. Then select **Work > Backlogs > Board**.

2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.

Your Kanban board appears.

## Add work items

From the Kanban board, you can add work items, open them, and modify them. To add work items, open the backlog by choosing the **Backlog** link. To add a work item, select the  plus sign, enter a title, and then press Enter.

The screenshot shows a Kanban board with several columns: New, Active, Resolved, and Closed. The New column has a red box around the '+ New item' button. Below it is a card with the title 'Hello World'. The other columns are empty.

Or, you can add work items to the bottom of the product backlog. Open the backlog by choosing the **Backlog** link.

From the Kanban board, you can't add work items, but you can open them and annotate them. To add work items, open the backlog by choosing the **Backlog** link. Also, you can't update the status of a work item by drag-and-drop to a different column or reorder cards within a column.

## Update status of work items

As work completes in one stage, update the status of an item by dragging it to a downstream stage.

The screenshot shows a Kanban board with five columns: Backlog, Analyze, Develop, Test, and another Develop column. In the 'Develop' column, there is a card titled 'Customer can find the nearest Fabrikam Fiber location' with an orange arrow pointing towards the 'Test' column. The 'Test' column has a card titled 'Welcome back page'.

### NOTE

The ability for Stakeholders to drag-and-drop cards to different columns requires installation of Azure DevOps Server 2020.1 update. To learn more, see [Azure DevOps Server 2020 Update 1 RC1 Release Notes, Boards](#).

## Add details to a work item

To add information to a work item, open it by double-clicking the title or by selecting it and then typing Enter. Change one or more field values, add a description, [add a tag](#), or add a comment in the **Discussion** section. You can also choose the **Attachments** tab and drag-and-drop or upload a file to share with others.

You can only assign work to a user who has been added to the project.

### NOTE

The work item form you see may differ from those shown in the following images. The basic functionality is the same, however, changes have been made with different versions of Azure DevOps.

- Agile process

- Basic process
- Scrum process
- CMMI process

For example, here we assign the story to Raisa Pokrovskaya and we add a discussion note, at-mentioning Raisa. Choose **Save & Close** when done.

The screenshot shows a 'User Story' card with the following details:

- Title:** USER STORY 1\*
- Assignee:** Raisa Pokrovskaya
- Comments:** 0 comments
- Tags:** Add tag
- Actions:** Save & Close, Follow, Refresh, More
- Fields:**
  - State:** New
  - Reason:** New
  - Area:** Fabrikam Fiber
  - Iteration:** Fabrikam Fiber
- Description:** Switch initial view to the updated design.
- Acceptance Criteria:** Click to add Acceptance Criteria
- Discussion:** A comment from Raisa Pokrovskaya: '@Raisa Pokrovskaya - Can you make this happen in the next week?' with a timestamp of 10/10/2018 10:10 AM.
- Planning:**
  - Story Points:**
  - Priority:** 2
  - Risk:**
- Classification:** Value area: Business
- Development:**
- Related Work:**

## Field descriptions

### Field

### Usage

### Title

Enter a description of 255 characters or less. You can always modify the title later.

### Assigned To

Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu lists only team members or contributors to the project.

#### NOTE

You can only assign work to a single user. If you need to assign work to more than one user, add a work item for each user and distinguish the work to be done by title and description. The Assigned To field only accepts user accounts that have been [added to a project or team](#).

### State

When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current status.

---

#### Reason

Use the default first. Update it when you change state as need. Each State is associated with a default reason.

---

#### Area (Path)

Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting. To change the dropdown list of areas, see [Define area paths and assign to a team](#).

---

#### Iteration (Path)

Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later during a planning meeting. To change the drop-down list of iterations, see [Define iteration paths and configure team iterations](#).

---

#### Description

Provide enough detail to create shared understanding of scope and support estimation efforts. Focus on the user, what they want to accomplish, and why. Don't describe how to develop the product. Do provide sufficient details so that your team can write tasks and test cases to implement the item.

---

#### Acceptance Criteria

Provide the criteria to be met before the work item can be closed. Define what "Done" means by describing the criteria for the team to use to verify whether the backlog item or bug fix is fully implemented. Before work begins, describe the [criteria for customer acceptance](#) as clearly as possible. Have conversations between the team and customers to determine the acceptance criteria. These criteria help ensure a common understanding within the team to meet customers' expectations. Also, this information provides the basis for acceptance testing.

---

#### Priority

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **1:** Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible.
  - **2:** Product cannot ship without the successful resolution of the work item, but it does not need to be addressed immediately.
  - **3:** Resolution of the work item is optional based on resources, time, and risk.
  - **4:** Resolution of the work item is not required.
- 

#### Value Area

A subjective rating of the issue or task it relates to the business. You can specify the following values:

- **Architectural:** Technical services to implement business features that deliver solution .
  - **Business:** Services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default).
- 

#### Effort, Story Points, Size

Provide a relative estimate of the amount of work required to complete an issue. Most Agile methods recommend that you set estimates for backlog items based on relative size of work. Such methods include

powers of 2 (1, 2, 4, 8) and the Fibonacci sequence (1, 2, 3, 5, 8, etc.). Use any numeric unit of measurement your team prefers.

The estimates you set are used to calculate [team velocity](#) and [forecast sprints](#).

## Add tags to a work item

Tags are useful for filtering backlogs, boards, and queries. As a Stakeholder, you can add existing tags to a work item, however, you can't add new tags.

From the web portal, open a work item and choose **Add tag** and type a keyword of an existing tag. Or, select from the list of previously assigned tags.



532 Hello World Web Site

Unassigned 0 comments Add tag

State	New	Area	Fabrikam Fiber A
Reason	New	Iteration	Fabrikam Fiber A

Tags that appear in the tag bar are already assigned to the work item. To unassign a tag, choose the x on the tag, [Web](#) .

### NOTE

By default, all Contributors and Stakeholders of public projects are granted permissions to add new and existing tags. Stakeholders in private projects can add tags that are already defined, but not add new tags. To grant or restrict permissions to create new tags, you set the project-level [Create tag definition](#) permission. To learn more, see [Change project-level permissions](#).

## Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.

### Discussion



Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.



Jamal Hartnett commented just now

@Christie Church - Assigning this to you



Christie Church commented less than a minute ago

I've updated the storyboard per our discussions yesterday.



Helena Petersen commented 9 minutes ago

@Christie Church, @Jamal Hartnett - Let's do an A/B test on the colors used in the form.

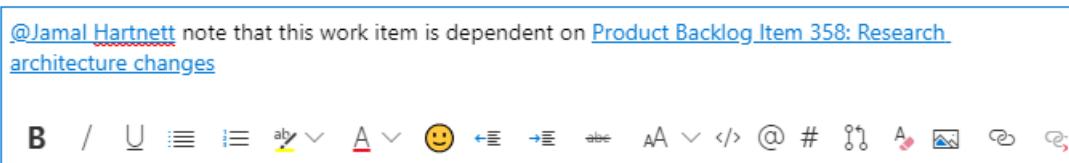


Jamal Hartnett commented 21 hours ago

Make sure the standards guidelines are written in a similar manner to those done for account setup.

The rich text editor tool bar displays below the text entry area. It appears when you click your cursor within each text box that supports text formatting.

## Discussion



@Jamal Hartnett note that this work item is dependent on [Product Backlog Item 358: Research architecture changes](#)

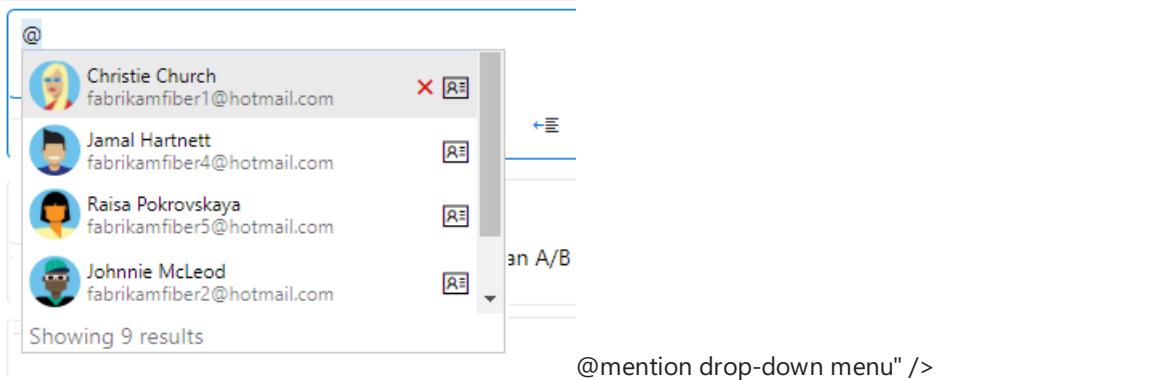
### NOTE

There is no Discussion work item field. To query work items with comments entered in the Discussion area, you filter on the [History field](#). The full content of the text entered into the Discussion text box is added to the History field.

### Mention someone, a group, work item, or pull request

Choose one of these icons —@, #, or !! — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or to open the same menu, you can type @, #, or !.

#### Discussion

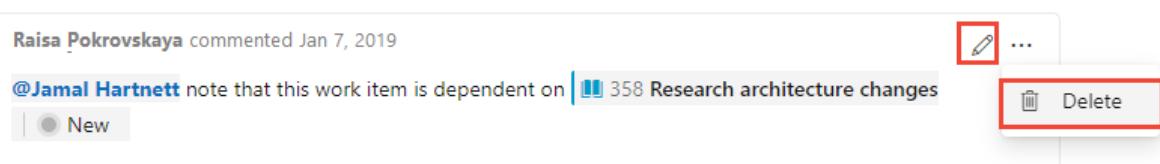


@mention drop-down menu" />

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing @ and the group name, such as a team or security group.

### Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose  **Edit** or choose the  actions icon and then choose **Delete**.



### NOTE

Editing and deleting comments requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the [@mention control](#) to notify another team member about the discussion. Simply type @ and their name. To reference a work item, use the [#ID control](#). Type # and a list of work items that you've recently

referenced will appear from which you can select.

To reference a work item, use the #ID control. Type # and a list of work items that you've recently referenced will appear from which you can select.

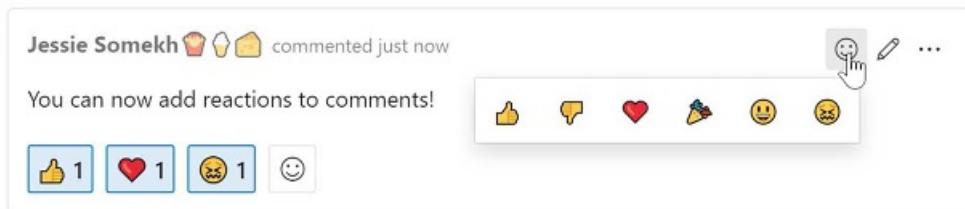
You can't edit or delete comments once you've entered them.

#### IMPORTANT

For on-premises Azure DevOps Server, you must configure an SMTP server in order for team members to receive notifications.

### Add a reaction to a comment

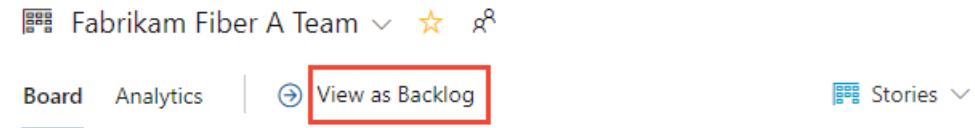
Add one or more reactions to a comment by choosing a smiley icon at the upper-right corner of any comment. Or, choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, choose the reaction on the bottom of your comment. The following image shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.



### Check the backlog and prioritized work

You can check the product backlog to see how the team has prioritized work. Backlog items appear in priority order. Work item types may include bugs depending on the settings made for the team.

From the Kanban board, choose **View as backlog**.



From the Kanban board, choose **View as backlog**.



From the Kanban board, choose **Backlog**.

The screenshot shows the 'Backlogs' section of the Azure DevOps interface. The 'Backlog' tab is highlighted with a red box. The main area displays 'Backlog items' with various status filters like 'Current' and 'Sprint 1'. Below the list are buttons for 'New', 'Create query', 'Column options', and a search icon.

You should see the list of backlog items listed in priority order. You can add a backlog item which will be placed at the bottom of the list. With Stakeholder access, you can't re-prioritize work.

To view or edit a work item, select it and choose **Enter**.

## Find work assigned to you, or query for other work items

1. Choose **Boards > Work Items**, and then select **Assigned to me**.

The screenshot shows the 'Work Items' page with the 'Assigned to me' pivot selected, indicated by a red box. The interface includes a search bar and filters for 'Types' and 'States'. Below the pivot, there are sections for 'Following', 'Mentioned', 'My activity', and activity logs for 'Recently updated', 'Recently completed', and 'Recently created' items.

Type	State	Assignee
with permissions	New	Fabrikam Fiber\Voice
hitecture changes	New	Fabrikam Fiber\Voice
	In Progress	Fabrikam Fiber\Voice

You can focus on relevant items inside a project using one of the seven pivots as described next.

Additionally, you can filter and sort each pivot view. For details, see [View and add work items using the Work Items page](#).

2. To query for work items, see [View, run, or email a work item query](#).

1. Open **Work > Queries** and select **Assigned to me** to see the list of work items assigned to you.

The screenshot shows the 'Queries' page in Visual Studio Team Foundation Server 2015. The 'Assigned to me' query is selected. The results table shows two items:

ID	Work Item Type	Title	State
190	Bug	Simplify the search experience	New
191	Bug	Log-in button needs to be more prominent	New

Below the table, the title 'Bug 190: Simplify the search experience' is shown, along with its details and edit options.

2. Or, open any of the queries defined in the Shared Queries folder.

The screenshot shows the 'Queries' page in Visual Studio Team Foundation Server 2015. The 'Work in progress' query is selected. The results table shows multiple tasks:

ID	Work Item Type	State	Remaining Work
164	Task	In Progress	8
165	Task	In Progress	8
166	Task	In Progress	6
167	Task	In Progress	2
168	Task	In Progress	2
169	Task	In Progress	1
170	Task	In Progress	4
173	Task	In Progress	2
174	Task	In Progress	1.5
181	Task	In Progress	1
186	Task	In Progress	1

3. And, you can [create new queries or edit existing queries](#) and save them under My Queries folder.

New |

Assigned to me  
Unsaved work items

My favorites  
Drag queries here to add t...

Team favorites  
Drag queries here to add t...

My Queries

Shared Queries

Current Sprint

Blocked Tasks

Work in Progress

Results **Editor** Charts

Column options Copy query URL

Type of query Flat list of work items Query across projects

Filters for top level work items

And/Or	Field	Operator	Value
+	Iteration Path	Under	Fabrikam\Sprint 1
+	Work Item Type	In Group	Microsoft.TaskCategory
+	State	=	In Progress
+ Add new clause			

## Related articles

For a comparison chart of Stakeholder versus Basic access, see this [feature matrix](#). See also these quickstart guides:

- [Add work items](#)
- [Create your backlog](#)
- [Kanban quickstart](#)
- [Access levels](#)
- [Change access levels](#)

# View permissions for yourself or others

12/13/2022 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Learn how to view your permissions or the permissions that are set for others in Azure DevOps. If you don't have a permission to access a feature or function, you can request it from the right resource.

Permissions are set at the collection, project, and object level as described in [Get started with permissions, access, and security groups](#). So to view the permissions you have, you need to open the permissions at the object, project, or collection level.

## NOTE

This article shows how to view permissions assigned to a user at the project-level or collection-level. However, the steps are similar when you work from the Security dialog of an object.

## Prerequisites

- You must have a project to connect to. If you don't have a project yet, [create one](#).
- You must be a member of the Project Valid Users Group or Project Collection Valid Users Group to view permissions.

## View project-level permissions

## NOTE

To enable the preview feature, for the new user interface for the [Project Permissions Settings Page](#), see [Enable preview features](#).

- [Preview page](#)
- [Current page](#)

1. Choose **Project Settings** and then **Permissions**.

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' project. On the left, there's a sidebar with icons for Overview, Boards, Repos, Pipelines, and Artifacts. Below this is a 'Project settings' button, which is highlighted with a red box. The main content area is titled 'Project Settings' and contains sections for General, Boards, Repos, Pipelines, and Test. Under General, the 'Permissions' section is highlighted with a red box. To the right, there's a 'Permissions' panel with tabs for 'Groups' and 'Users'. The 'Groups' tab is selected, showing a list of groups with their names and icons: Build Administrators (BA), Contributors (C), Fabrikam Fiber group (FG), Project Administrators (PA), Project Valid Users (PU), Readers (R), Release Administrators (RA), Email team (ET), and FabrikamFiber Team (FT). The 'Users' tab is also present but not selected.

2. Choose **Users**. To filter the list, enter a name into the *Search groups or users* box.

This screenshot shows the 'Permissions' page with the 'Users' tab selected. A search bar at the top contains the name 'Jamal'. Below the search bar, a list of users is displayed, each with a profile icon, name, and email address. The user 'Jamal Hartnett' (fabrikamfiber4@hotmail.com) is listed. A red box highlights the 'Users' tab in the top navigation bar.

Name	Email
AB	Azure Boards
MS	MyPublicProject Build Service (fabrikam)
DS	Demo 11 Build Service (fabrikam)
Christie Church	fabrikamfiber1@hotmail.com
CR	Chuck Reinhart
CR	fabrikamfiber3@hotmail.com
Jamal Hartnett	fabrikamfiber4@hotmail.com

3. Choose the name you want. The project-level permissions for that user displays. These permissions are based on the groups the user belongs to or the permissions set specifically for the user's account.

The screenshot shows the 'Users' page in Microsoft Project. On the left, there's a sidebar with a back arrow labeled 'Users' and a search bar. Below the search bar is a list of users, with 'Jamal Hartnett' selected. The main content area is titled 'Jamal Hartnett' and shows 'Permissions' and 'Member of' tabs. The 'Permissions' tab is active, displaying a table of permissions:

Permission	Status
Alter trace settings	Not set
Create new projects	Allow
Delete team project	Not set
Edit instance-level information	Not set
View audit log	Not set
View instance-level information	Allow (inherited)

Below this, under 'Service Account', are three more permissions:

Permission	Status
Make requests on behalf of others	Not set
Trigger events	Not set
View system synchronization information	Not set

Finally, under 'Boards', there are two permissions:

Permission	Status
Create process	Not set
Delete field from organization	Not set

4. Choose **Member of** to see which security groups and teams that the user belongs to.

Here we see that *Jamal Hartnett* belongs to several teams and the Project Collection Administrators group for several projects.

**Jamal Hartnett**

**Add member of**

**Permissions** **Member of** **Filter groups**

Name	User or scope
MT MyPublicProject Team	[MyPublicProject]
FS Fiber Suite	[Fabrikam Fiber A]
I Internet	[Fabrikam Fiber]
MT MyFirstProject Team	[MyFirstProject]
FT Fabrikam Team	[Fabrikam Fiber]
PA Project Administrators	[MyPublicProject]
PA Project Administrators	[MyFirstProject]

1. Open Project Settings. Choose the gear settings icon, and choose Security.

Fabrikam Fiber

Search

Overview Work **Security** Version Control Policies Agent queues Notifications Service Hooks

2. Begin entering the name into the *Filter users and groups* box. The system automatically shows the names that begin with the characters you enter.

Create group

Showing 1 result

Permissions	Members	Member of
Create tag definition	Allow (inherited)	
Create test runs	Allow (inherited)	
Delete and restore work items	Not set	
Delete team project	Not set	
Delete test runs	Allow (inherited)	
Edit project-level information	Not set	
Manage project properties	Not set	

3. Choose the name you want. The project-level permissions you have set are based on the groups you belong to or the permissions set for your account.

Create group < fabrikam > Jamal Hartnett

Jamal

 Jamal Hartnett

Permissions	Member of
Create tag definition	Allow (inherited)
Create test runs	Allow (inherited)
Delete and restore work items	Not set
Delete team project	Not set
Delete test runs	Allow (inherited)
Edit project-level information	Not set
Manage project properties	Not set
Manage test configurations	Allow (inherited)
Manage test environments	Allow (inherited)
Move work items out of this project	Not set
Permanently delete work items	Not set
Rename team project	Not set
View project-level information	Allow (inherited)
View test runs	Allow (inherited)

For a description of each permission, see [Permissions and groups reference](#).

4. Choose **Member of** to see which security groups the user belongs to.

Here we see that *Jamal Hartnett* belongs to several teams and the Project Collection Administrators group.

Create group < fabrikam > Jamal Hartnett

Jamal

 Jamal Hartnett

Permissions	Member of			
+ Add...		⟳		Search
Display Name	Username Or Scope			
 Customer Service	[Fabrikam Fiber]	<a href="#">Remove</a>		
 Fabrikam Fiber Team	[Fabrikam Fiber]			
 Web	[Fabrikam Fiber]			
 Project Collection Administris...	[fabrikam]			

For a description of each group, see [Permissions and groups reference](#).

## View organization or collection-level permissions

Open admin settings for the organization or a project collection.

1. Choose the  Azure DevOps logo to open **Projects**. Then choose **Organization settings**.

The screenshot shows the Azure DevOps interface. On the left, there's a sidebar with organization navigation: 'FabrikamFiber01' (selected), 'fabrikamfiber02', 'fabrikamfiber', '14 more organizations', and 'New organization'. Below this is a 'What's new' section with a 'Sprint 162 release notes' card. At the bottom of the sidebar is a 'Organization settings' button, which is highlighted with a red box. The main area is titled 'FabrikamFiber01' and contains tabs for 'Projects', 'My work items', and 'My pull requests'. A large 'Fabrikam Fiber' logo is displayed.

2. Choose **Permissions**, the **Project Collection Administrators** group, and then **Members**.

The screenshot shows the 'Project Collection Administrators' members list. The 'Members' tab is selected, indicated by a red box. The table lists the following members:

Name	Type	Username or scope
Christie Church fabrikamfiber1@hotmail.com	user	fabrikamfiber1@hotmail.com
PA Project Collection Service Accounts	group	[mseng]
Jamal Hartnett fabrikamfiber4@hotmail.com	user	fabrikamfiber4@hotmail.com
Raisa Pokrovskaya fabrikamfiber5@hotmail.com	user	fabrikamfiber5@hotmail.com
Helena Petersen fabrikamfiber8@hotmail.com	user	fabrikamfiber8@hotmail.com

3. Follow steps 2 through 4 in the procedure outlined previously for view project-level permissions.

1. Choose the Azure DevOps logo to open **Projects**. Then choose **Admin settings**.

The screenshot shows the Azure DevOps portal interface. On the left, there's a sidebar with organization navigation: 'FabrikamFiber01' (selected), 'fabrikamfiber02', 'fabrikamfiber', '14 more organizations', and 'New organization'. Below this is a 'What's new' section with a 'Sprint 162 release notes' summary. At the bottom of the sidebar is a red-bordered 'Organization settings' button. The main area is titled 'FabrikamFiber01' and contains tabs for 'Projects', 'My work items', and 'My pull requests'. A large card for 'Fabrikam Fiber' is visible. The 'Organization settings' button is highlighted with a red border.

2. Choose **Security**, the **Project Collection Administrators** group, and then **Members**.

The screenshot shows the 'Create group' interface on the left and the 'Project Collection Administrators' members list on the right. In the 'Create group' sidebar, 'Project Collection Administrators' is selected and highlighted with a red border. The main area shows the 'fabrikam > Project Collection Administrators' page with three tabs: 'Permissions' (disabled), 'Members' (selected and highlighted with a red border), and 'Member of'. The 'Members' tab displays a list of users with their display names, email addresses, and user IDs. The 'Permissions' tab is disabled.

Display Name	Username Or Scope
Project Collection Service ...	[REDACTED]
Christie Church	fabrikamfiber1@hotmail.com
Jamal Hartnett	fabrikamfiber4@hotmail.com
Raisa Pokrovskaya	fabrikamfiber5@hotmail.com

3. Follow steps 2 through 4 in the procedure outlined previously for view project-level permissions.

1. Choose the settings icon and select **Organization settings** or **Collection settings**.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. At the top, there's a navigation bar with links for Dashboards, Code, Work, Build and Release, Test, and more. On the far right of this bar is a gear icon, which is highlighted with a red box. A red arrow points from this box down to the 'Organization settings' link in a vertical dropdown menu on the right side of the screen. The dropdown menu lists various project management and development features: Overview, Work, Security, Version Control, Policies, Agent Queues, Notifications, Service Hooks, Services, Test, Release, Dashboards, Team settings, and Organization settings. The 'Organization settings' link is specifically highlighted with a red box.

2. Choose **Security**, **Project Collection Administrators** group, and then **Members**.

The screenshot shows the 'Security' page for the 'fabrikam' collection. The 'Security' tab is highlighted with a red box. Below it, under 'Create group', there's a 'Filter users and groups' input field. A list of Azure DevOps Groups is shown, with 'Project Collection Administrators' highlighted with a red box. To the right, the 'fabrikam > Project Collection Administrators' section is displayed. It has tabs for 'Permissions' (highlighted with a red box) and 'Members'. The 'Members' tab is also highlighted with a red box. Under 'Permissions', there are 'Add...', 'Edit', and 'Search' buttons. The 'Members' table lists three users:

Display Name	Username Or Scope
Project Collection Service ...	
Christie Church	fabrikamfiber1@hotmail.com
Jamal Hartnett	fabrikamfiber4@hotmail.com
Raisa Pokrovskaya	fabrikamfiber5@hotmail.com

3. Follow steps 2 through 4 in the procedure outlined previously for view project-level permissions.

## View object-level permissions

You can define the security or permissions for a number of objects. You access them from the context menu of the object.

From the web portal, open the Security dialog for the object whose permissions you want to set. For specific instructions, see the following articles:

## Area

### Task

Wiki & Dashboard permissions

- [README & Wiki](#)
- [Dashboards](#)

Azure Repos, Azure Pipelines/DevOps (code, build, test, release) permissions

- [Git branch](#)
- [Git repository](#)
- [TFVC](#)
- [Builds](#)
- [Release pipeline security](#)
- [Approvals and approvers](#)

Azure Boards/Work tracking permissions

- [Area and iteration paths](#)
- [Work item query and folder](#)
- Plan permissions

## Next steps

[Look up a member of the Project Administrators group](#)

## Related articles

- [Troubleshoot permissions](#)
- [Permissions and role lookup guide](#)

# Sign up for Azure DevOps

12/13/2022 • 2 minutes to read • [Edit Online](#)

## Azure DevOps Services

When you sign up for Azure DevOps, you get the following tier of free services:

- 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 Kanban boards
- **Azure Repos:** Unlimited private Git repos
- **Azure Artifacts:** Two GiB free per organization

You can sign up for Azure DevOps with either a [Microsoft](#) or [GitHub](#) account. For more information, see [What is Azure DevOps?](#)

## Sign up with a Microsoft account

1. If you don't have one, [create a Microsoft account now](#).
2. Go to [Azure DevOps](#) and select **Start free**.
3. Enter your account credentials and go through the sign-up process.

Azure DevOps does the following tasks:

- Creates an organization.
- Creates a project named after your *newly created* Microsoft account.
  - If you signed up with an existing Microsoft account, you need to [create a project](#) next.

Sign in to your organization at any time [https://dev.azure.com/{Your\\_Organization}](https://dev.azure.com/{Your_Organization}).

## Sign up with a GitHub account

### NOTE

If your GitHub email address is already associated with an organization in Azure DevOps that's [connected to Azure Active Directory \(Azure AD\)](#), you can't sign in with your GitHub account. You must sign in with your Azure AD account.

1. If you don't have one, [create a GitHub account now](#).
2. Go to [Azure DevOps](#) and select **Start free with GitHub**.
3. Enter your account credentials and go through the sign-up process. You're asked to **Authorize Microsoft-corp**.

Azure DevOps does the following tasks:

- Creates an organization. Sign in to your organization at any time [https://dev.azure.com/{Your\\_Organization}](https://dev.azure.com/{Your_Organization}).
- Turns on the *Invite GitHub users* policy by default.

Azure DevOps      fabrikamfiber / Organization Settings / Policies

<p><b>Organization Settings</b></p> <p>General</p> <ul style="list-style-type: none"><li><a href="#">Overview</a></li><li><a href="#">Projects</a></li><li><a href="#">Users</a></li><li><a href="#">Billing</a></li><li><a href="#">Global notifications</a></li><li><a href="#">Usage</a></li><li><a href="#">Extensions</a></li><li><a href="#">Azure Active Directory</a></li></ul> <p>Security</p> <ul style="list-style-type: none"><li><a href="#">Policies</a></li><li><a href="#">Permissions</a></li></ul>	<p><b>Policy</b></p> <p><b>Application connection policies</b></p> <table border="1"><tr><td>Alternate authentication credentials</td><td>On</td></tr><tr><td>Third-party application access via OAuth</td><td>On</td></tr><tr><td>SSH authentication</td><td>On</td></tr></table> <p><b>Security policies</b></p> <table border="1"><tr><td>Allow public projects</td><td>On</td></tr></table> <p><b>User policies</b></p> <table border="1"><tr><td>Invite GitHub users</td><td>Off</td></tr></table>	Alternate authentication credentials	On	Third-party application access via OAuth	On	SSH authentication	On	Allow public projects	On	Invite GitHub users	Off
Alternate authentication credentials	On										
Third-party application access via OAuth	On										
SSH authentication	On										
Allow public projects	On										
Invite GitHub users	Off										

Before you can invite users, you must create a project, as shown in **Next steps**. For more information about GitHub authentication, see [FAQs](#).

## Next steps

[Create a project](#) or

[Add users or groups to a team or project](#)

## Related articles

- [Plan your organizational structure in Azure DevOps](#)
- [Change the location of your organization](#)
- [Add users to your organization](#)
- [GitHub authentication FAQs](#)

# Create an organization

12/13/2022 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use an organization to connect groups of related projects, and help to scale up your enterprise. You can use a personal Microsoft account, GitHub account, or a work or school account. Use your work or school account to *automatically connect* your organization to your Azure Active Directory (Azure AD).

## NOTE

All organizations must be manually created via the web portal. We don't support automated creation of organizations. We do support automated organization configuration, project creation, and resource provisioning via [REST API](#).

## Prerequisites

- Understand how to [plan your organizational structure](#).
- Determine whether you want to use only Microsoft accounts or authenticate users with Azure AD. For more information, see [Choosing your organization administrator account type](#).

## IMPORTANT

Currently, you can only use letters from the English alphabet in your organization name. Start organization names with a letter or number, followed by letters, numbers or hyphens, and they must end with a letter or number.

## Create an organization

1. Sign in to [Azure DevOps](#).
2. Select **New organization**.

The screenshot shows the Azure DevOps interface for creating a new organization. On the left, a sidebar lists existing organizations: 'fabrikamfiberorg' (selected), 'FabrikamTestProject', 'Fabrikam-Fiber-Inc', and 'fabrikamprime', followed by a link to '3 more organizations'. Below this is a button labeled 'New organization' which is highlighted with a red box. On the right, the main area displays the selected organization 'fabrikamfiberorg' with its projects: 'FabrikamFiber' (1st project) and other projects represented by dots. At the bottom of the sidebar is a link to 'Organization settings'.

3. Confirm information, and then select **Continue**.

This screenshot shows the final configuration step before creating the organization. It features the Azure DevOps logo and the heading 'Almost done...'. The user is prompted to 'Name your Azure DevOps organization' with the input field containing 'dev.azure.com/fabrikamffiber'. Below this, it says 'We'll host your projects in' with a dropdown menu set to 'Central US'. A large blue 'Continue' button is centered at the bottom, with a red box highlighting it.

This screenshot shows the final step where the system takes the user to their new organization. It features the Azure DevOps logo and the text 'Taking you to your Azure DevOps organization...'. A small circular loading icon is centered at the bottom of the screen.

Congratulations, you're an organization owner!

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

## Create a project collection

A project collection is a container of projects. By grouping projects together, you can manage projects more efficiently and assign the same resources to those projects.

For more information about how to create a project collection, see [Create a project collection](#).

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 Kanban boards
- **Azure Repos:** Unlimited private Git repos
- **Azure Artifacts:** Two GiB free per organization

## Next steps

[Create a project](#)

## Related articles

- [Get started with Azure Repos and Visual Studio](#)
- [Rename your organization](#)
- [Change organization time-zone](#)
- [Change organization owner](#)
- [Delete your organization](#)
- [Resolve orphaned organization](#)

# Get started managing your project

12/13/2022 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

With most Azure DevOps Services, you can start using the service and configure resources as you go. No up-front work is required. Most settings define defaults.

If you've created a project or been added to the **Project Administrators** group, you'll want to be familiar with the administrative tasks you're charged with. There are a few tasks you might want to do to ensure a smooth operational experience.

## NOTE

This article provides an overview of tasks a member of the **Project Administrators** group should review and attend to. For information on tasks to be performed by members of the **Project Collection Administrators** group, see [Manage your organization or project collection](#).

## Add users to your project

You add users to a team or project so they can contribute to the team and project. Users can be added to multiple teams and projects.

Users that have been added to an organization, can easily be added to a project by adding them to a team or inviting them to contribute to a project.

Team administrators can add users to their team which automatically adds them to the project. By adding users to a team, you make team-specific tools aware of them, such as the team security group, Team Members widget, and sprint capacity planning tools. To learn more about teams, see [About teams and Agile tools](#).

Members of the **Project Administrators** group can add users to a project. Adding users to a team or project automatically adds them to the project's **Contributors** group. Members of this group have permissions to most features needed to contribute to work items, code, builds, and releases. For an overview of default permissions, see [Default permissions quick reference](#).

Once users have been added to a project or organization, you can browse for their display name or user name (email alias) from any people-picker tool. Users can connect to a project and access features available through a supported client or the web portal.

To learn more, see the following articles:

- [Add users or groups to a team or project](#)
- [Manage your organization or project collection, Add users to your organization](#)
- [Connect to a project](#)

## Share your project vision, set up a project wiki

Each project has a summary page that's useful for sharing information through README files. Or, redirect users to a project Wiki. For users who are new to your project, we recommend that you [set up your project summary page](#). Or, you can [provision a Wiki](#). Use these features to share established processes and procedures for your

project.

Each project has a summary page that's useful for sharing information through **README files**. For users who are new to your project, we recommend that you [set up your project summary page](#). Or, you can [provision a Wiki](#). Use these features to share established processes and procedures for your project.

## Remove unused services

To simplify the web portal user interface, you can disable select services. For example, if you use a project only to log bugs, then disable all services except for **Boards**. To learn more, see [Turn a service on or off](#).

This example shows that **Test Plans** is disabled:

The screenshot shows the 'Project Settings > Overview' page. On the left, there's a sidebar with 'General' expanded, showing 'Overview', 'Teams', 'Security', 'Notifications', 'Service hooks', and 'Dashboards'. Below that is 'Boards' expanded, showing 'Project configuration', 'Team configuration', and 'GitHub connections'. On the right, under 'Azure DevOps services', there are five service cards: 'Boards' (Flexible agile planning with boards and cross-product issues), 'Repos' (Repos, pull requests, advanced file management and more), 'Pipelines' (Build, manage, and scale your deployments to the cloud), 'Artifacts' (Continuous delivery with artifact feeds containing NuGet, npm, Maven, Universal, and Python packages), and 'Test Plans' (Structured manual testing at any scale for teams of all sizes). The 'Test Plans' toggle switch is set to 'Off'.

## Manage security and permissions

Access to select tasks is controlled by permissions and security groups. To quickly understand the defaults configured for your project, see [Default permissions and access](#).

The following table lists the permissions assigned at the project-level. All of these permissions are granted to members of the **Project Administrators** group, except for the **Delete shared Analytics views** and **Edit shared Analytics views** permissions which are not set. For a description of each permission, see [Permissions and groups reference, Groups](#).

The following table lists the permissions assigned at the project-level. All of these permissions are granted to members of the **Project Collection Administrators** group. For a description of each permission, see [Permissions and groups reference, Groups](#).

### NOTE

Permissions associated with Analytics requires that the Inherited process model is selected for an on-premises project collection.

### General

- Delete team project

- Edit project-level information
  - Manage project properties
  - Rename team project
  - Suppress notifications for work item updates
  - Update project visibility
  - View project-level information
- 
- Delete team project
  - Edit project-level information
  - Manage project properties
  - Rename team project
  - Suppress notifications for work item updates
  - View project-level information
- 
- Delete team project
  - Edit project-level information
  - Manage project properties
  - Rename team project
  - View project-level information

## Boards

- Bypass rules on work item updates
  - Change process of team project
  - Create tag definition
  - Delete and restore work items
  - Move work items out of this project
  - Permanently delete work items
- 
- Bypass rules on work item updates
  - Change process of team project
  - Create tag definition
  - Delete and restore work items
  - Move work items out of this project
  - Permanently delete work items
- 
- Create tag definition
  - Delete and restore work items
  - Permanently delete work items

## Analytics

- Delete shared Analytics views
- Edit shared Analytics views
- View analytics

## Test Plans

- Create test runs
- Delete test runs
- Manage test configurations
- Manage test environments

- View test runs

To learn more about security and setting permissions at the project-level, review the following articles:

- [Get started with permissions, access, and security groups](#)
- [Change permissions at the project-level](#)

### Add members to the Project Administrators group

The person who creates a project is automatically added as a member to the **Project Administrators** group. Members of this group have permissions to manage project configuration, repositories, pipeline resources, teams, and all project-level permissions.

It's always a good idea to have more than one person who has administrative privileges. To add a user to this group, see [Change permissions at the project level](#), [Add members to the Project Administrators group](#).

### Grant or restrict permissions

Permissions are managed at the following three levels and through role-based assignments.

- object
- project
- organization or collection

As a member of the **Project Administrators** group, you can grant or restrict permissions for all objects and at the project-level. To delegate specific tasks to others, we recommend that you add them to a built-in or custom security group or add them to a specific role. To learn more, see the following articles.

- [Role-based permissions](#)
- [Add or remove users or groups, manage security groups](#)
- [Grant or restrict access to select features and functions](#)
- [Set object-level permissions](#)

## Review and update notifications

A number of notifications are predefined for each project you add. Notifications are based on subscription rules. Subscriptions arise from the following areas:

- [Out-of-the-box or default subscriptions](#).
- [Team, project, and organization or collection subscriptions](#) defined by a team administrator or member of the **Project Administrators** or **Project Collection Administrators** groups.

If users believe they're getting too many notifications, you can direct them to [opt out of a subscription](#).

Description	Type	Notifies	State
<b>Build</b>			
 Build completes Notifies you when a build you queued or that was queued for you compl...	Build completed (any project)	 You	<input checked="" type="checkbox"/> On
<b>Code (Git)</b>			
 Pull request reviewers added or removed Notifies you when you are added to a pull request or when a user is add...	Pull request (any project)	 You	<input checked="" type="checkbox"/> On
 Pull request completion failures Notifies you when a pull request you created fails to complete	Pull request (any project)	 You	<input checked="" type="checkbox"/> On
 Pull request changes Notifies you when changes are made to a pull request you created or are...	Pull request (any project)	 You	<input checked="" type="checkbox"/> On
 A comment is left on a pull request Notifies you about comments made to a pull request you created or a di...	Pull request comment (any project)	 You	<input checked="" type="checkbox"/> On

## Determine traceability requirements

If you're using most of Azure DevOps Services—Boards, Repos, Pipelines, and Test Plans—you'll want to alert your teams to those features that support end-to-end traceability. To get started, we recommend that you review the following articles:

- [Cross-service integration and collaboration overview](#)
- [End-to-end traceability](#)

## Set DevOps policies

Set policies to support collaboration across your teams and automatically remove obsolete files. To set policies that govern Azure Repos, Azure Pipelines, and Azure Test Plans, review the following articles:

- [Manage branch policies](#)
- [Add Team Foundation Version Control \(TFVC\) check-in policies](#)
- [Set build and release pipeline retention policies](#)
- [Set test retention policies](#)

## Configure and customize Azure Boards

You can configure and customize Azure Boards to support a number of business requirements for planning and tracking work. At a minimum, you'll want to configure the following elements:

- Area paths to group work items by team, product, or feature area
- Iteration paths to group work into sprints, milestones, or other event-specific or time-related periods

If you're new to Azure Boards and want an in-depth overview of what you can configure and customize, see [Configure and customize Azure Boards](#).

### Define area and iteration paths to track work

If you support several products, you can assign work items by feature area by defining [area paths](#). To assign work items to specific time intervals, also known as sprints, you configure [iteration paths](#). To use the Scrum tools—sprint backlogs, taskboards, and team capacity—you need to configure several sprints. For an overview, see [About areas and iteration paths](#).

The image shows two side-by-side screenshots of the Azure DevOps interface. On the left, the 'Iterations' page is displayed, showing a table of sprints for the 'Fabrikam Fiber' project. A red box highlights the 'Iterations' tab in the top navigation bar. On the right, the 'Areas' page is shown, also with a red box highlighting the 'Areas' tab in the top navigation bar. Both pages include sections for general project settings, teams, permissions, notifications, service hooks, and dashboards.

## Customize work-tracking processes

You and your team can start using all work-tracking tools immediately after you create a project. But often, one or more users want to customize the experience to meet one or more business needs. You can customize the process easily through the user interface. As such, you'll want to establish a methodology for who will manage the updates and evaluate requests.

### NOTE

By default, organization owners and users added to the **Project Collection Administrators** security group are granted permission to create, edit, and manage processes used to customize the work-tracking experience. If you want to lock down who is able to perform these tasks, you can set permissions at the organization-level to **Deny**.

To learn more, see these articles:

- [About process customization and inherited processes](#)
- [Customize a project](#)
- [Add and manage processes](#)
- [On-premises XML process customization](#)
- [Add or modify a field to track work](#)
- [Add or modify a work item type](#)

## Integrate with other services

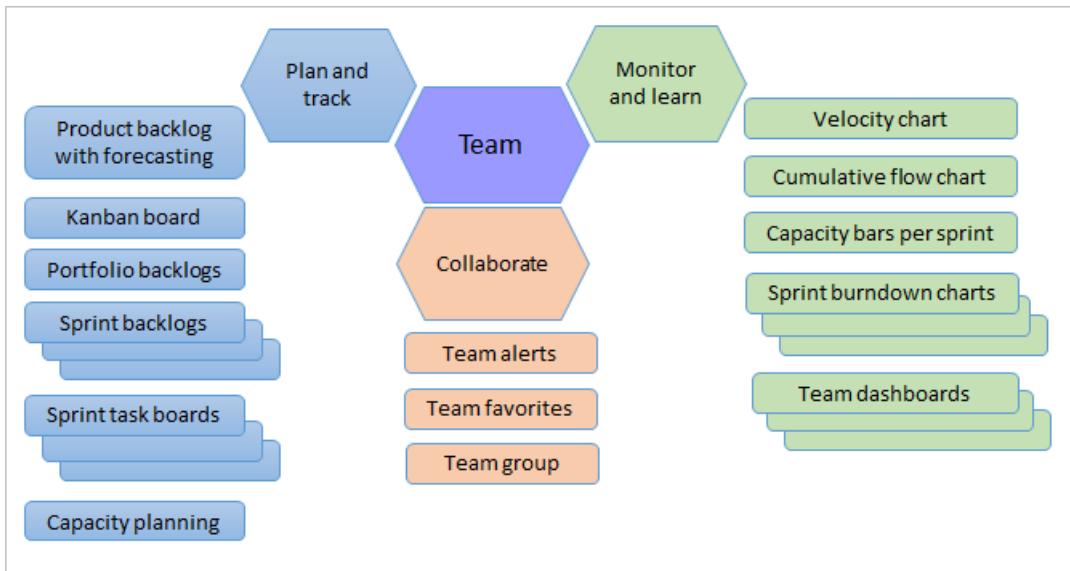
Azure DevOps supports integration with Azure, GitHub, and many other services. As a member of the **Project Administrators** group, you can configure integration with many of these services. To learn more, see the following articles.

- [Azure DevOps and GitHub integration overview](#)
- [Azure Boards and GitHub integration](#)
- Microsoft teams integration:
  - [Azure Boards with Microsoft Teams](#)
  - [Azure Repos with Microsoft Teams](#)
  - [Azure Pipelines with Microsoft Teams](#)
- Slack integration:
  - [Azure Boards with Slack](#)
  - [Azure Repos with Slack](#)

- Azure Pipelines with Slack
- Integrate with service hooks

## Add teams to scale your project

As your organization grows, we recommend that you add teams to scale your project. Each team gets [access to their own set of customizable Agile tools](#).



To learn more, see the following articles:

- [About projects and scaling your organization](#)
- [Add a team, move from one default team to several teams](#)
- [Add a team administrator](#)

## Next steps

[Share your project vision](#)

## Related articles

- [Project and team quick reference](#)
- [Get started managing your organization or project collection](#)
- [About user, team, project, and organization-level settings](#)
- [Project and team quick reference](#)
- [Get started managing your organization or project collection](#)
- [About user, team, project, and organization-level settings](#)
- [TFS administration](#)

# Get started managing your organization or project collection

12/13/2022 • 12 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

After you create an organization or project collection, you'll want to add contributors and configure policies, settings, and other options available to you. This article provides an overview of tasks you'll want to review to ensure you're setting up your organization or collection to get maximal use of your services.

Each organization is associated with one and only one collection. If you need to create another organization, see [Plan your organizational structure](#) and [Create an organization](#).

When you install Azure DevOps Server, you automatically create a default collection. If you need to create another project collection, see [Manage project collections](#).

## NOTE

This article provides an overview of tasks that require membership in the **Project Collection Administrators** group. For information on tasks to be performed by members of a **Project Administrators** group, see [Manage your project](#).

## Add users to your organization

For large enterprises, the recommended method to manage Azure DevOps users, is to connect Azure DevOps to Azure Active Directory (Azure AD) and manage user access through security groups defined in Azure AD. That way, when you add and remove users or groups from Azure AD, you automatically add and remove these same users and groups from Azure DevOps. You limit the maintenance of managing permissions and user access.

For small and large enterprises, you can add users and security groups directly through the web portal [Organization settings > Users](#) interface. All users added to an organization can be added to one or more projects defined for the organization.

For large enterprises, the recommended method to manage Azure DevOps users, is to connect Azure DevOps to Active Directory (AD) and manage user access through security groups defined in AD. That way, when you add and remove users or groups from AD, you automatically add and remove these same users and groups from Azure DevOps. Typically, you should install Active Directory before installing Azure DevOps. You limit the maintenance of managing permissions and user access.

For small and large enterprises, you add users to a server instance through the web portal [Access levels](#) interface. All users added to the server instance can be added to one or more projects defined within the project collection(s) defined in the server instance.

When you add users, you specify their *access level*/which determines the features they can use through the web portal. To learn more, review these resources:

- [Get started with permissions, access, and security groups](#)
- [About access levels](#)
- [Add organization users and manage access](#)
- [Connect your organization to Azure Active Directory](#)

#### **NOTE**

If the **Limit user visibility and collaboration to specific projects** preview feature is enabled for the organization, users added to the **Project-Scoped Users** group won't be able to access projects that they haven't been added to. To learn more, see [Limit user visibility for projects and more](#) later in this article.

- [Get started with permissions, access, and security groups](#)
- [About access levels](#)
- [Add users or groups to an access level](#)
- [Install Active Directory Domain Services \(Level 100\)](#)

#### **NOTE**

Even if you add a user or group to an access level, you must also [add them to a project](#) for them to connect to a project and access features available through a supported client or the web portal.

## Set up billing

Azure DevOps Services charges for the following services as described in [Pricing for Azure DevOps](#).

- Individual services:
  - Microsoft-hosted CI/CD parallel jobs
  - Self-hosted CI/CD parallel jobs
  - Storage of Azure Artifacts feeds
- User licenses for **Basic** or **Basic + Test Plans**.

All organizations are granted five free **Basic** licenses and unlimited users with **Stakeholder** access. For information on each access level, see [About access levels](#).

If your organization requires more than five contributors, then you'll need to set up billing. Users that have a Visual Studio subscription can be added without incurring any further billing charges. Billing is based on the access level, **Basic** or **Basic + Test Plans**, that you assign to the user. To learn more, see [Set up billing](#).

## Manage security and permissions

Access to select tasks is controlled by permissions and security groups.

The following table lists the permissions assigned at the organization or collection-level. All of these permissions, except for the **Make requests on behalf of others** permission, are granted to members of the **Project Collection Administrators** group. For a description of each permission, see [Permissions and groups reference, Groups](#).

### General

- Alter trace settings
- Create new projects
- Delete team project
- Edit instance-level information
- View instance-level information

### Service Account

- Make requests on behalf of others

- Trigger events
- View system synchronization information

## Boards

- Administer process permissions
- Create process
- Delete field from organization or account
- Delete process
- Edit process
- Delete field from organization or account

## Repos (TFVC)

- Administer shelved changes
- Administer workspaces
- Create a workspace

## Pipelines

- Administer build resource permissions
- Manage build resources
- Manage pipeline policies
- Use build resources
- View build resources
- Administer build resource permissions
- Manage build resources
- Use build resources
- View build resources

## Test Plans

- Manage test controllers

## Auditing

- Delete audit streams
- Manage audit streams
- View audit log

## Policies

- Manage enterprise policies

To learn more about security and setting permissions at the collection-level, review the following articles:

- [Get started with permissions, access, and security groups](#)
- [Change permissions at the organization or collection-level.](#)

## Add members to the Project Collection Administrators group

The person who creates an organization is automatically added as a member to the **Project Collection Administrators** group. Members of this group have permissions to manage the settings, policies, and processes for the organization, create and manage all projects defined in the organization, and install and manage extensions.

The person who creates a project collection is automatically added as a member to the **Project Collection Administrators** group. Members of this group have permissions to manage the settings, policies, and processes for the organization, create and manage all projects defined in the organization, and install and manage extensions.

It's always a good idea to have more than one person who has administrative privileges. To add a user to this group, see [Change permissions at the organization level](#), [Add members to the Project Collection Administrators group](#).

### Limit user visibility for projects and more

By default, users added to an organization can view all organization and project information and settings.

To restrict select users, such as Stakeholders, Azure Active Directory guest users, or members of a particular security group, you can enable the **Limit user visibility and collaboration to specific projects** preview feature for the organization. Once that is enabled, any user or group added to the **Project-SScoped Users** group, are restricted in the following ways:

- Restricted users to only access those projects to which they've been explicitly added to.
- Restricts views that display list of users, list of projects, billing details, usage data, and more that is accessed through **Organization Settings**.
- Limits the set of people or groups that appear through people-picker search selections and the ability to **@mention** people.

To enable this feature, see [Manage or enable features](#).

#### NOTE

All security groups are organization-level entities, even those groups that only have permissions to a specific project. From the web portal, visibility of some security groups may be limited based on user permissions. However, you can discover the names of all groups in an organization using the **azure devops** CLI tool or our REST APIs. To learn more, see [Add and manage security groups](#).

### Limit identity search and selection

For organizations that manage users and groups using Azure Active Directory (Azure AD), people pickers provide support for searching all users and groups added to Azure AD, not just those users and groups added to your project. People pickers support the following Azure DevOps functions:

- Selection of a user identity from a work tracking identity field such as **Assigned To**
- Selection of a user or group using **@mention** in a work item discussion or rich-text field, a pull request discussion, commit comments, or changeset or shelveset comments
- Selection of a user or group using **@mention** from a wiki page

As shown in the following image, you simply start typing into a people picker box until you find a match to a user name or security group.

## Discussion

The screenshot shows a search interface for users within a project. At the top left is a placeholder icon with the letter 'C'. Below it, the search term '@C' is entered. A list of results is displayed, starting with 'Christie Church' (fabrikamfiber1@hotmail.com) and 'Chuck Reinhart' (fabrikamfiber3@hotmail.com). Each result has a small 'RE' icon to its right. Below the list is a search bar with the placeholder 'Search' and a note 'Showing 2 results'. To the right of the search bar are several small icons for filtering and sorting.

### WARNING

When the **Limit user visibility and collaboration to specific projects** preview feature is enabled for the organization, project-scoped users are unable to search for users who were added to the organization through Azure Active Directory group membership, rather than through an explicit user invitation. This is an unexpected behavior and a resolution is being worked on. To self-resolve this issue, disable the **Limit user visibility and collaboration to specific projects** preview feature for the organization.

Users and groups who are added to the **Project-SScoped Users** group can only see and select users and groups in the project they are connected to from a people picker. To scope people pickers for all project members, see [Limit user visibility for projects and more](#) earlier in this article.

To limit the identity selection to just those users and groups added to a project, perform the following procedure for your organization and projects.

1. Enable the **Limit user visibility and collaboration to specific projects** preview feature for the organization. To learn how, see [Manage or enable features](#).
2. Add the users to your project(s) as described in [Add users to a project or team](#). Users added to a team are automatically added to the project and team group.
3. Open **Organizations Settings>Security>Permissions** and choose **Project-SScoped Users**. Choose the **Members** tab. Add all users and groups that you want to scope to the project(s) you've added them to. To learn more, see [Set permissions at the project- or collection-level](#). The **Project-SScoped Users** group only appears under the **Permissions>Groups** once **Limit user visibility and collaboration to specific projects** preview feature is enabled.

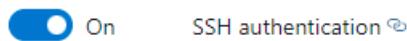
### Set security policies

Configure the security policies for your organization through the **Organization settings>Policies** page. These policies enable you to grant or restrict the following features:

- Third-party application access via OAuth
- SSH authentication
- Creation of public projects
- Invitation of GitHub user accounts

## Policies

### Application connection policies



### Security policies



### User policies



To learn more, see [Change application connection & security policies for your organization](#).

## Enable preview features for your organization

As new features are introduced to Azure DevOps Services, you can choose to enable them or not for an organization. Some features are introduced and automatically enabled. You can try them out, provide feedback, and work with those features that meet your requirements.

When you enable a feature at the organization level, you essentially turn it on for all users of your account. Each user can then disable the feature if they so choose. If you disable a feature at the organization level, user settings are not changed. Users can enable or disable the feature on their own.

To enable or disable a preview feature, see [Manage or enable features](#).

The following features are only enabled or disabled at the organization-level:

- [Limit identity search and selection](#)
- [Full Access to Azure Pipelines for Stakeholders](#)

## Install and manage extensions

An extension is an installable unit that adds new capabilities to your projects. Azure DevOps extensions support the following functions:

- Planning and tracking of work items, sprints, scrums, and so on
- Build and release flows
- Code testing and tracking
- Collaboration among team members

For example, to support [code search](#), install the [Code Search extension](#).

You want to tell your users about extensions and that they can [request an extension](#). To install and manage extensions, you must be an organization Owner, a member of the **Project Collection Administrators** group. Or, you can get added to the [Manager role for extensions](#).

### Install Code Search

Code Search is a free Marketplace extension that you must install to enable searching across all your source

repositories. To learn how, see [Install and configure Search](#).

## Enable or disable Analytics

The Analytics service is the reporting platform for Azure DevOps, replacing the previous platform based on SQL Server Reporting Services. Built for reporting, Analytics is optimized for fast read-access and server-based aggregations. Use it to answer quantitative questions about the past or present state of your projects.

To learn more, see [What is the Analytics service?](#) and [Install or enable the Analytics service](#).

## Adjust time zone and other organization settings

When you create an organization, you specify the name of your organization and select the region where your organization is hosted. The default **Time zone** is set to *UTC*. You can update the **Time zone** and specify a Privacy URL from the **Organization settings > Overview** page. To learn more about these settings, see the following articles:

- [Time zone settings and usage](#)
- [Add a privacy policy URL for your organization](#)

## Configure DevOps settings

There are a few settings that you define at the organization-level to support devops work. These include the following items:

- [Add agent pools](#)
- [Define pipeline retention settings](#)
- Define repository settings:
  - [Default branch name for new repositories](#)
  - [Gravatar images](#).

## Customize work-tracking processes

All work-tracking tools are available immediately after you create a project. Often, one or more users may want to customize the experience to meet one or more business needs. Processes are easily customized through the user interface. However, you may want to establish a methodology for who manages the updates and evaluates requests.

To learn more, see the following articles:

- [About process customization and inherited processes](#)
- [Customize a project](#)
- [Add and manage processes](#)

All work-tracking tools are available immediately after you create a project. Often, one or more users may want to customize the experience to meet one or more business needs. But, you may want to establish a methodology for who manages the updates and evaluates requests.

To learn more, see [On-premises XML process model](#).

## Alert users with information banners

You can quickly communicate with your Azure DevOps users through information banners. Use banners to alert your Azure DevOps users to upcoming changes or events without sending mass emails. To learn how, see [Add and manage information banners](#).

## Review and update notifications

A number of notifications are predefined at the organization or collection level. You can disable or modify these subscriptions, or add new subscriptions as described in [Manage notifications for a team, project, or organization](#).

Notifications			
Default subscriptions		Subscribers	Statistics
Description	Type	Settings	
<hr/>			
Build			
 Build completes	Build completed (any project)		Build completes
Code (Git)			
 Pull request reviewers added or removed	Pull request (any project)		Notifies the team when it is added or removed as a reviewer for a pull request
 Pull request completion failures	Pull request (any project)		Pull request completion failures
 Pull request changes	Pull request (any project)		Notifies the team when changes are made to a pull request the team is a reviewer for
 A comment is left on a pull request	Pull request comment (any project)		A comment is left on a pull request
<hr/>			
Extension management			
 Extensions have been modified	Extension		Extensions have been modified
 Extensions are requested or requests are updated	Extension request (batch)	 ...	Extensions are requested or requests are updated

### Configure an SMTP server

In order for team members to receive notifications, [you must configure an SMTP server](#).

## Scale your organization or collection

To learn about scaling your organization, review the following articles.

- [About projects and scaling your organization](#)
- [Plan your organizational structure](#)

## Related articles

- [Project and team quick reference](#)
- [FAQs about signing up and getting started](#)
- [Organization management](#)
- [About user, team, project, and organization-level settings](#)
- [Project and team quick reference](#)
- [Security & identity](#)

- [About user, team, project, and organization-level settings](#)
- [Azure DevOps Server administration](#)

# Add users or groups to a team or project

12/13/2022 • 21 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You add users to a team or project so they can contribute to the team and project. For enterprise organizations with large user bases, we recommend you use Azure Active Directory to add and manage new users through security groups. However, to enable flexibility for all size organizations, the following operations are supported:

- Team and project administrators can add new users to their team or project, unless the policy [Allow team and project administrators to invite new users](#) is disabled. New users are ones that haven't been added to the organization.
- When adding new users through the team and project user interfaces, the system automatically assigns an access level to the user.
- Adding users to a team or project automatically adds them to the Contributors group for the project. Members of the Contributors group have permissions to most features needed to contribute.
- By adding users to a team, you make team-specific tools aware of them, such as the team security group, Team Members widget, and sprint capacity planning tools.
- Once users have been added to a project or organization, you can browse for their display name or user name (email alias) from any people-picker tool.

You add users to a team or project so they can contribute to the team and project. For enterprise organizations with large user bases, we recommend you use Active Directory or Windows Group to manage users through security groups. However, to enable flexibility for all size organizations, the following operations are supported:

- Team and project administrators can add existing users to their team or project. Existing users are ones known to the project collection through Active Directory or Windows group.
- Adding users to a team or project automatically adds them to the Contributors group for the project. Members of the Contributors group have permissions to most features needed to contribute.
- By adding users to a team, you make team-specific tools aware of them, such as the team security group, Team Members widget, and sprint capacity planning tools.
- Once users have been added to a project or organization, you can browse for their display name or user name (email alias) from any people-picker tool.

You add projects to an organization or project collection and you add teams to projects. To learn more, see:

- [Create a project](#)
- [Add team, go from one default team to others](#)

## IMPORTANT

### Version

Azure DevOps Services

Azure DevOps Server 2022

Azure DevOps Server 2020

Azure DevOps Server 2019

TFS 2018

Previous versions

To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.

To learn which on-premises version you are using, see [Look up your Azure DevOps platform and version](#)

## Supported options for adding users

Depending on the interface you use, you can exercise different options for adding new or existing users to teams or projects.

Team and project administrators can add existing users to their team or project. Existing users are ones that are known to a project collection through the Active Directory or Windows Group created for the server that hosts the on-premises Azure DevOps Server.

**Administrator level**

**Interface**

**Supported tasks**

Team administrators

[Team Members dashboard widget](#)

Add new or existing users to a team. Send new users an invite.

Team administrators

[Project Settings>Teams>Team>Members](#)

Add existing users or groups to a team, or remove a member.

Project Administrators

[Project Summary page, Invite](#)

Add new or existing users. Send new users an invite. Optionally add users to one or more teams.

Project Administrators

[Project Settings>Permissions>Groups>Group Members](#)

Add existing users or groups to a security group. By adding to a team group, you effectively add them to the

team. Optionally remove a user from a group.

Project Collection Administrators

## Organization Settings > Users

Add new users to an organization and send an invite. Must specify the access level. Optionally add them to select projects. Can use Group rules to further manage groups being added.

Project Collection Administrators

`az devops user` CLI

Add new users to an organization and send an invite. Must specify the access level.

Azure Active Directory Administrators

Azure Active Directory

Users you add to Azure Active Directory connected to Azure DevOps Services are added to the Project Collection Valid Users group. To learn more, see [Connect your organization to Azure Active Directory](#).

Active Directory Administrators

Active Directory or Windows Group

Users you add to Active Directory or Windows Group connected to Azure DevOps are added as members of the Project Collection Valid Users group. They have access to all projects within a project collection. To learn more, see [Set up groups for use in Azure DevOps on-premises](#).

## Prerequisites

- You must have an organization and project. If you don't have a project yet, [create one](#).
- To add users to or remove users from a team, you must be added as a [team administrator](#), or be a member of one of the administrative groups.
- To add users to or remove users from a project, you must be a member of the [Project Administrators group](#).
- When the organization is connected to Azure Active Directory, the [Allow team and project administrators to invite new users](#) policy must be enabled for team administrators or members of the Project Administrators group to add new users.
- To add users or manage users for an organization, you must be a member of the [Project Collection Administrators group](#). Organization owners are automatically members of this group.
- If you don't have a project yet, [create one](#)
- To add users to or remove users from a team, you must be added as a [team administrator](#), or be a member of one of the administrative groups.
- To add users to or remove users from a project, you must be a member of the [Project Administrators group](#).
- To add users or manage users for a server, you must be a member of the [Azure DevOps Administrators group](#).

If you're new to Azure DevOps, you may want to familiarize yourself with the information provided in these articles:

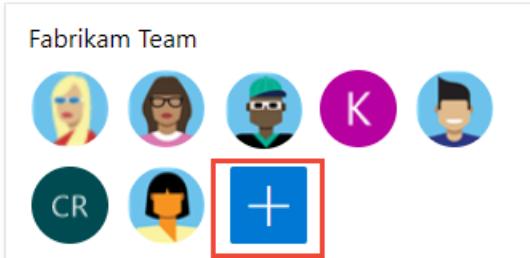
- [Get started with permissions, access levels, and security groups](#)
- [About projects and scaling your organization](#)
- [Default permissions and access quick reference](#)

- About teams and Azure Boards tools

## Add a user from the Team Members widget

As a team administrator, you can add new or existing members from the **Team Members** dashboard widget. To add this widget to a dashboard, see [Add widgets to a dashboard](#).

1. To invite someone to your team, choose the plus button on the Team Members widget.



2. For new users, enter their email address. For existing users, type their name until it resolves as a known name to the system. You can add several email addresses or account names by separating them with a semicolon (;).

Choose the entry listed under **Add users** to complete the entry.

**NOTE**

Any valid email address is acceptable. When the user accepts the invitation and signs into Azure DevOps, they register their email address as a Microsoft account and choose a password.

Choose the name that appears to complete the entry.

**Invite members to Fabrikam Team** X

Search and add users to your Fabrikam Team

Users

Add users

Search and add users to your team. You can add several email addresses or account names by separating them with a semicolon (;). You can't add users from other teams.

Cancel Add

## Invite members to Fabrikam Team X

Search and add users to your Fabrikam Team

Users

 Jia-hao Tseng  
fabrikamfiber9@hotmail.com

CancelAdd

3. Complete the addition.

When the user is unknown, you'll get a notification that an access level must be assigned. To complete the invitation, choose **Add**.

Choose **Add** to complete adding the user. Known users don't receive an invitation.

## Invite members to Fabrikam Team X

Search and add users to your Fabrikam Team

Users

i fabrikamfiber11@hotmail.com has not been assigned an access level, we will attempt to assign Stakeholder.  
[Learn more](#)

CancelAdd

## Invite members to Fabrikam Team X

Search and add users to your Fabrikam Team

Users

Use semicolons to separate multiple email addr

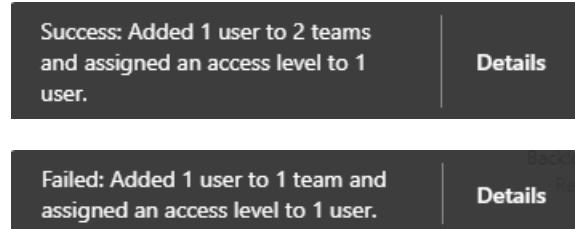
CancelAdd

When adding a new user, the system assigns Stakeholder as the access level when all five Basic access levels have been assigned. Active contributors to a project need to have Basic access as a minimum. A Project Collection Administrator can change the access level and resend invitations from the [Organization Settings > Users](#) page.

**NOTE**

Users that have limited access, such as Stakeholders, won't be able to access select features even if granted permissions to those features. To learn more, see [Permissions and access](#).

4. (Optional) A message will briefly display on the screen to indicate success or failure. Choose **Details** to open the notification and review details.



A success message indicates the status of adding the user to the system.

A failure message indicates why the addition of the user failed.

## Notifications

System issues are problems in the system that require admin attention. Session notifications are triggered by user activities in this session.

A screenshot of a "Session notifications" dialog. At the top, it says "Session notifications 3" and has a "Dismiss all" button. Below that, a message is listed: "Added 1 user to 1 team and assigned an access level to 1 user." with a timestamp "Just now" and a close button. Underneath, there's a "Less details" button, followed by a table with columns "User" and "Message". One row shows "fabrikamfiber11@hotmail.com" and "Success". An ellipsis "..." is at the bottom right.

User	Message
fabrikamfiber11@hotmail.com	Success

## Notifications



System issues are problems in the system that require admin attention. Session notifications are triggered by user activities in this session.

Session notifications 1

Dismiss all

✖ Add 1 user

Just now

▲ Less details

User	Message
fabrikamfiber11@hotmail.com	You are trying to invite a user from outside your directory, but the security setting of this organization doesn't allow it. <a href="#">Learn more</a>

5. New users receive an email inviting them to sign in to the project. Existing users don't receive any formal notification.

## Add users or groups to a team

Add existing users or security groups to a team from the **Project settings > Teams** page. From this interface you can view, add, or remove users and security groups to/from a team. To add a custom security group, see [Add or remove users or groups, manage security groups](#).

### NOTE

To enable the new user interface for managing teams, enable the **New Teams Page** from the **Preview features** tool. To learn how, see [Manage or enable features](#).

- [Preview UI](#)
- [Current UI](#)

You can toggle between direct or expanded membership views. The **Direct Members** view displays users and groups that have been added to the team. The **Expanded Members** view replaces any Azure DevOps groups with the members that belong to those groups. Azure Active Directory or Active Directory groups aren't expanded.

1. Open a backlog or board for a team and choose the team profile icon. Then choose **Team Settings**.

Here we open the Board for the Web team and from there the team profile.

The screenshot shows the Azure DevOps Boards interface. On the left, the navigation bar has 'Boards' selected, indicated by a red box and the number 1. In the center, the 'Fabrikam Fiber Team' board is displayed with a red box around the 'Fabrikam Fiber Team' dropdown menu, labeled 2. On the right, a sidebar titled 'Fabrikam Fiber Team' shows 'Members (1)' and a list of team members, with a red box around the 'Team Settings' link, labeled 3.

2. If you need to switch the team context, use the team selector within the breadcrumbs.

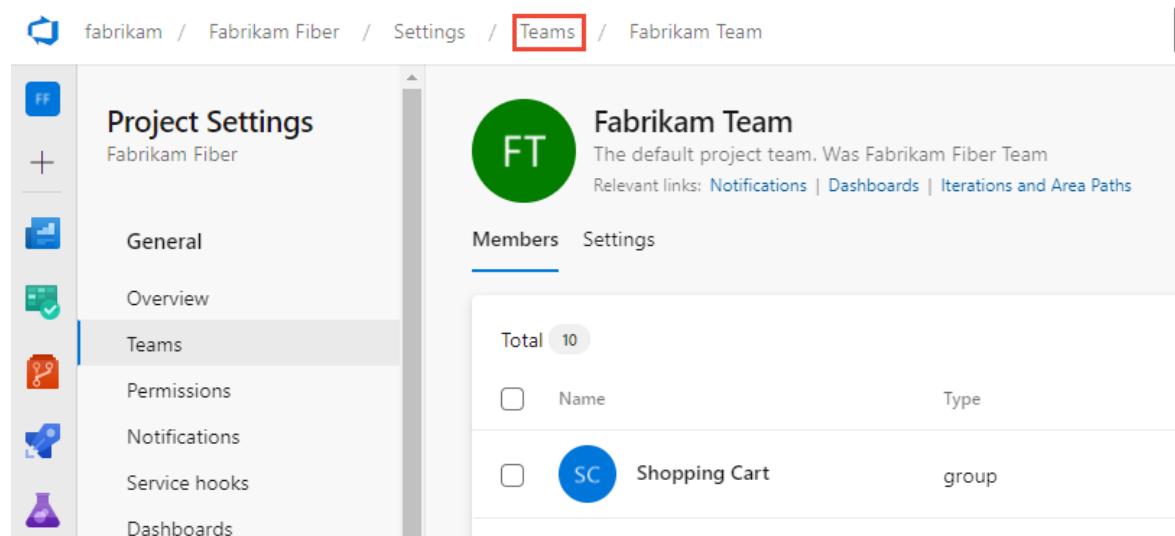
The screenshot shows the 'Project Settings' page for 'Fabrikam Fiber'. The 'Teams' section is highlighted with a red box, labeled 1. The breadcrumb path shows 'Teams' selected, indicated by a red box and the number 2. The right panel displays the 'Fabrikam Team' settings, including a list of members, with a red box around the 'Add' button in the top right corner, labeled 3.

3. Choose Add.

The screenshot shows the 'Fabrikam Team' members list. The 'Members' tab is selected. A search bar at the top right contains the placeholder 'Search users and groups'. Below it, a 'Direct Members' dropdown is shown with a red box around the 'Add' button, labeled 1. The list of members includes three users: Christie Church, Chuck Reinhart, and Jamal Hartnett, each with their email address listed to the right. A red box highlights the 'Add' button in the top right corner of the member list area, labeled 2.

4. Enter the sign-in addresses or display name for each account you want to add. You can also add a project security group—such as another team group, custom group, or Azure Active Directory group when used by the organization. Add them one at a time or all at the same time. You can enter several identities into

the text box, separated by commas.



The screenshot shows the Azure DevOps interface for managing teams. The top navigation bar includes 'fabrikam / Fabrikam Fiber / Settings / Teams / Fabrikam Team'. The 'Teams' link is highlighted with a red box. On the left, a sidebar menu for 'Project Settings' (Fabrikam Fiber) lists 'General', 'Overview', 'Teams' (which is selected and highlighted with a blue border), 'Permissions', 'Notifications', 'Service hooks', and 'Dashboards'. The main content area is titled 'Fabrikam Team' with a green circular icon containing 'FT'. It describes the team as the default project team and provides links for 'Notifications', 'Dashboards', and 'Iterations and Area Paths'. Below this, there are two tabs: 'Members' (selected) and 'Settings'. The 'Members' section shows a total of 10 members. A table lists two entries: 'Name' (checkbox) and 'SC Shopping Cart' (checkbox). The 'SC Shopping Cart' entry is highlighted with a blue circle containing 'SC'. To the right of the table, columns show 'Type' (User) and 'group'. A green box labeled 'TIP' contains the note: 'You must enter user and group names one at a time. However, after entering a name, the account is added to the list, and you can enter another name in the Identities text box before choosing to save your changes.'

**TIP**  
You must enter user and group names one at a time. However, after entering a name, the account is added to the list, and you can enter another name in the Identities text box before choosing to save your changes.

You may need to choose the  refresh icon to see your updates.

5. To add an account as a team administrator, choose the Settings page and then choose **Add** under the Administrators section. For details, see [Add a team administrator](#)

Choose the **Current page** tab for information on adding a user to a team. The **New Teams Page** preview feature is only available for Azure DevOps Services at this time.

## Remove users or groups from a team

From the team's **Members** page, you can remove members.

- [Preview UI](#)
- [Current UI](#)

1. To remove members, open the team's **Members** page, choose **Direct Members**, check the checkbox of the user you want to remove, choose  **More options**, and then choose **Remove**.

Total	11	Type	Username or scope
<input type="checkbox"/>	Name		
<input type="checkbox"/>	Christie Church fabrikamfiber1@hotmail.com	user	fabrikamfiber1@hotmail.com
<input type="checkbox"/>	Chuck Reinhart fabrikamfiber3@hotmail.com	user	fabrikamfiber3@hotmail.com
<input checked="" type="checkbox"/>	fabrikamfiber11@hotmail.com fabrikamfiber11@hotmail.com	user	fabrikamfiber11@hotmail.com
<input type="checkbox"/>	fabrikamfiber12@hotmail.com fabrikamfiber12@hotmail.com	user	fabrikamfiber12@hotmail.com

**TIP**

To remove a team administrator as a team member, you must first remove them as an administrator.

2. Confirm the removal by choosing **Delete** in the confirmation message.

**Delete Member**

Are you sure you want to remove "fabrikamfiber11@hotmail.com" from the "Fabrikam Team" team?

**Cancel** **Delete**

Choose the **Current page** tab for information on adding a user to a team. The **New Teams Page** preview feature is only available for Azure DevOps Services at this time.

## Invite users from the Summary page

As a member of the Project Administrators group, you can add members to a project from the **Summary** page and optionally add them to one or more teams. To learn more about the **Summary** page, see [Share your project vision, view project activity](#).

**NOTE**

For on-premises Azure DevOps, all email actions require an [SMTP server to be configured](#).

1. Open the **Project>Summary** page, and choose **Invite**.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. The left sidebar has a 'Summary' tab selected. The main area displays the project's name 'Fabrikam Fiber' and a section titled 'About this project' with a note about guidance and source control. A red box highlights the 'Invite' button in the top right corner of the main content area.

1. Open the Project>Summary page, and choose the Add button.

The screenshot shows the 'Members' section of the project settings. It displays three user icons and a blue plus sign button for adding more members.

1. For new users, enter their email address. For existing users, type their name until it resolves as a known name to the system. You can add several email addresses or account names by separating them with a semicolon (;).

Choose the entry listed under **Add users** to complete the entry.

If you're adding a user known by the organization or collection, type the name or email address and then choose the name that appears to complete the entry.

The screenshot shows the 'Invite members to Fabrikam Fiber' dialog box. It includes a search bar, a 'Users' section with a list item 'fabrikamfiber11@hotmail.com', and an 'Add users' section with the same email address. A red box highlights the email address in the 'Add users' input field. At the bottom are 'Cancel' and 'Add' buttons.

## Invite members to Fabrikam Fiber

X

Search and add users to your Fabrikam Fiber

Users

 Jia-hao Tseng  
fabrikamfiber9@hotmail.com

Jia-hao Tseng  
fabrikamfiber9@hotmail.com



Cancel

Add

### NOTE

Any valid email address is acceptable. When the user accepts the invitation and signs into Azure DevOps, they register their email address as a Microsoft account and choose a password.

2. Optionally, select the teams you want to add the user to and then choose **Add** to complete the invitation.

When the user is unknown, you'll get a notification that an access level must be assigned. To complete the invitation, choose **Add**.

Choose **Add** to complete the invitation.

## Invite members to Fabrikam Fiber

X

Search and add users to your Fabrikam Fiber

Users

 X | Use semicolons to separate multi

Add to team(s)



 fabrikamfiber12@hotmail.com has not been assigned an access level, we will attempt to assign Stakeholder.  
[Learn more](#)

Cancel

Add

## Invite members to Fabrikam Fiber X

Search and add users to your Fabrikam Fiber

Users

 Johnnie McLeod	X	+
--	---	---

Add to team(s)

Service Delivery (+3) ▼

Shopping Cart  
 Service Delivery  
 Service Status  
 Fabrikam Team  
 Customer Profile  
 Voice  
 Organization Management  
 TV  
 Phone

[Add all](#)

Cancel

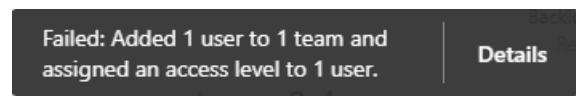
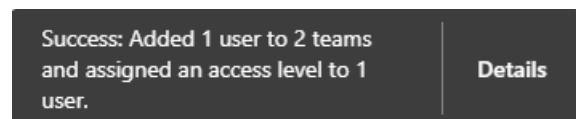
**Add**

When adding a new user, the system assigns Stakeholder as the access level when all free five Basic access levels have been assigned. Active contributors to a project need to have Basic access as a minimum. A Project Collection Administrator can change the access level from the [Organization Settings>Users page](#).

### NOTE

Users that have limited access, such as Stakeholders, won't be able to access select features even if granted permissions to those features. To learn more, see [Permissions and access](#).

3. (Optional) A message will briefly display on the screen to indicate success or failure. Choose **Details** to open the notification and review details.



A success message indicates the status of adding the user to the system.

A failure message indicates why the addition of the user failed.

## Notifications



System issues are problems in the system that require admin attention. Session notifications are triggered by user activities in this session.

Session notifications 3

Dismiss all

✓ Added 1 user to 1 team and assigned an access level to 1 user.

Just now

Less details

User

Message

Success 1

fabrikamfiber11@hotmail.com

...

## Notifications



System issues are problems in the system that require admin attention. Session notifications are triggered by user activities in this session.

Session notifications 1

Dismiss all

✗ Add 1 user

Just now

Less details

User

Message

Failed 1

fabrikamfiber11@hotmail.com

You are trying to invite a user from outside your directory, but the security setting of this organization doesn't allow it. [Learn more](#)

4. New users receive an email inviting them to sign in to the project. Existing users don't receive any formal notification.

## Add users or groups to a project

As a member of the **Project Administrators** group, you can add users or groups to a project from the **Project settings > Permissions** page by adding them to a security group. To add a custom security group, see [Add or remove users or groups, manage security groups](#).

### NOTE

To enable the **Project Permissions Settings Page** preview page, see [Enable preview features](#).

- Preview UI
- Current UI

1. Open the web portal and choose the project where you want to add users or groups. To choose another project, see [Switch project, repository, team](#).
2. Choose **Project settings**, and then **Permissions**.

Name	Description
BA Build Administrators	Members can create, modify and delete build definitions and manage queued and completed builds.
C Contributors	Members can add, modify, and delete items within the team project.

3. Under **Groups**, choose one of the following options:

- **Readers**: To add users who require read-only access to the project, choose.
- **Contributors**: To add users who contribute fully to this project or who have been granted Stakeholder access.
- **Project Administrators**: To add users who need to administrate the project. To learn more, see [Change project-level permissions](#).

Or, you can choose any team group to add users to a specific team.

Here we choose the **Contributors** group.

Name	Description
[Fabrikam Fiber]\Build Administrators	Members of this group can create, modify and delete build definitions and manage queued and completed builds.
<b>C [Fabrikam Fiber]\Contributors</b>	Members of this group can add, modify, and delete items within the team project.

4. Next, choose the **Members** tab.

The default team group, and any other teams you add to the project, get included as members of the

**Contributors** group. Add a new user as a member of a team instead, and the user automatically inherits Contributor permissions.

**TIP**

Managing users is much easier [using groups](#), not individual users.

5. Choose **Add** to add a user or a user group.

The screenshot shows the 'Contributors' page in Azure DevOps. At the top, there's a navigation bar with tabs: 'Permissions', 'Members' (which is selected), 'Member of', and 'Settings'. Below the navigation is a search bar labeled 'Search users and groups' with a magnifying glass icon. The main area is titled 'Members' and contains a table with two rows. The first row has columns: a checkbox, 'Name', 'Type', and 'Username or scope'. The second row shows a checkbox next to a green profile picture of a person named 'P', with 'Phone' in the Name column, 'group' in the Type column, and '[Fabrikam Fiber]' in the Username or scope column. To the right of the table is a blue 'Add' button with a white outline, which is highlighted with a red box.

6. Enter the name of the user account into the text box. You can enter several identities into the text box, separated by commas. The system automatically searches for matches. Choose the match(es) that meets your requirements.

The screenshot shows the 'Invite members to Contributors' dialog. At the top, it says 'Invite members to Contributors' and has a close button. Below that is a search bar with placeholder text 'Search and add users and/or groups to your group'. Underneath is a section titled 'Add users and/or groups' with a search input field containing 'Ch'. A dropdown menu lists two results: 'Christie Church' (fabrikamfiber1@hotmail.com) and 'Chuck Reinhart' (fabrikamfiber3@hotmail.com). Each result has a small profile picture and a 'Select' button to the right. At the bottom of the dialog are 'Cancel' and 'Save' buttons.

**NOTE**

The first time you add a user or group to Azure DevOps, you can't browse to it or check the friendly name. After the identity has been added, you can just enter the friendly name.

Choose **Save** when done.

7. You may customize user permissions for other functionality in the project. For example, in [areas and iterations](#) or [shared queries](#).

Choose the **Current page** tab for information on adding a user to a project. The Project Permissions Settings Page preview feature is only available for Azure DevOps Services at this time.

## Manage users or resend invitations

Project Collection Administrators can update user assignments and resend invitations. The various options they have are:

- Change the access level
- Manage user - add them to select projects
- Resend invite
- Remove direct assignments
- Remove from organization

To learn more, see [Add account users for Azure DevOps](#).

The screenshot shows the 'Users' page in the Azure DevOps interface. At the top, there are tabs for 'All users' (selected) and 'Group rules'. Below the tabs is a 'Filter users' input field and dropdowns for 'Access Level' and 'License Source'. On the right, there are 'Summary' and 'Add users' buttons. The main area displays a table of users with columns: Name, Access Level, License Source, Date Added, and Last Accessed. A user named 'fabrikamfiber11@hotmail.com' has a checkmark next to their name. A context menu is open for this user, with a red box highlighting the following options:

- Change access level
- Manage user
- Resend invite
- Remove direct assignments
- Remove from organization

Total 16 Selected 1	Name	Access Level	License Source	Date Added	Last Accessed ↑
<input checked="" type="checkbox"/> fabrikamfiber11@hotmail.com fabrikamfiber11@hotmail.com	Stakeholder	Direct	6/16/2021	Never	<ul style="list-style-type: none"><li>Change access level</li><li>Manage user</li><li>Resend invite</li><li>Remove direct assignments</li><li>Remove from organization</li></ul>
<input type="checkbox"/> fabrikamfiber12@hotmail.com fabrikamfiber12@hotmail.com	Stakeholder	Direct	6/16/2021	Never	
<input type="checkbox"/> CR Chuck Reinhart fabrikamfiber3@hotmail.com	Stakeholder	Direct	2/23/2018	8/1/2021	
<input type="checkbox"/> Francis Totten fabrikamfiber7@hotmail.com	Stakeholder	Direct	2/23/2018	1/2/2021	
<input type="checkbox"/> Johnnie McLeod fabrikamfiber2@hotmail.com	Stakeholder	Group Rule	2/23/2018	4/2/2021	

## List team members or team details

From the Azure DevOps CLI command, you can see details about a team or list the individual members of that team. To first see a list of all teams in your organization, use the [az devops team list](#) command.

[List team members](#) | [Show team details](#)

### NOTE

You can use the `az devops user` command to add users to an organization. There is no comparable command for adding users to a team or project.

### List team members

You can list the individual members of a team in your organization with the [az devops team list-member](#)

command. To get started, see [Get started with Azure DevOps CLI](#).

```
az devops team list-member --team
    [--org]
    [--project]
    [--skip]
    [--top]
```

#### Parameters

- **team**: Required. Name or ID of the team to show.
- **org**: Azure DevOps organization URL. You can configure the default organization using  
`az devops configure -d organization=ORG_URL`. Required if not configured as default or picked up using  
`git config`. Example: `--org https://dev.azure.com/MyOrganizationName/`.
- **project**: Name or ID of the project. You can configure the default project using  
`az devops configure -d project=NAME_OR_ID`. Required if not configured as default or picked up using  
`git config`.
- **skip**: Optional. Number of members to skip.
- **top**: Optional. Maximum number of members to return.

#### Example

The following command lists the first five members of the team named **Fabrikam Team** and returns the details in table format.

```
az devops team list-member --team "Fabrikam Team" --top 5 --output table
```

ID	Name	Email
3b5f0c34-4aec-4bf4-8708-1d36f0dbc468	Christie Church	fabrikamfiber1@hotmail.com
19d9411e-9a34-45bb-b985-d24d9d87c0c9	Johnnie McLeod	fabrikamfiber2@hotmail.com
8c8c7d32-6b1b-47f4-b2e9-30b477b5ab3d	Chuck Reinhart	fabrikamfiber3@hotmail.com
d291b0c4-a05c-4ea6-8df1-4b41d5f39eff	Jamal Hartnett	fabrikamfiber4@hotmail.com
bd30c189-db0f-4dd6-9418-5d8b41dc1754	Raisa Pokrovskaya	fabrikamfiber5@hotmail.com

#### Show team details

You can view details about a team in your organization with the `az devops team show` command. To get started, see [Get started with Azure DevOps CLI](#).

```
az devops team show --team
    [--org]
    [--project]
```

#### Parameters

- **team**: Required. Name or ID of the team to show.
- **org**: Azure DevOps organization URL. You can configure the default organization using  
`az devops configure -d organization=ORG_URL`. Required if not configured as default or picked up using  
`git config`. Example: `--org https://dev.azure.com/MyOrganizationName/`.
- **project**: Name or ID of the project. You can configure the default project using  
`az devops configure -d project=NAME_OR_ID`. Required if not configured as default or picked up using  
`git config`.

#### Example

The following command shows information about the team in your organization named **Fabrikam Team** and returns the details in table format.

```
az devops team show --team "Fabrikam Team" --output table
```

ID	Name	Description
a48cb46f-7366-4f4b-baf5-b3632398ed1e	Fabrikam Team	The default project team. Was Fabrikam Fiber Team

## Add users or groups to an access level

For on-premises deployments, you may need to set the access level for a user or group, particularly if those groups don't belong to the default access level. To learn more, see [Change access levels](#).

## Add users or groups to SQL Server Reports

If your on-premises deployment is integrated with SQL Server Reports, you need to manage membership for those products separately from their websites. See [Grant permissions to view or create SQL Server reports in Azure DevOps](#).

## Next steps

[Manage your project](#)

## Related articles

- [Add users and manage access](#)
- [Resources granted to project members](#)
- [Manage your organization, Limit identity search and selection](#)
- [Manage your organization, Limit user visibility for projects and more](#)
- [Manage permissions with command line tool](#)
- [Grant or restrict access using permissions.](#)
- [Resources granted to project members](#)
- [Grant or restrict access using permissions.](#)

# Manage and configure team tools

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As a team administrator, you can customize your backlogs and board to best meet how your team works. If you need to have a team created, request a member of your Project Administrators group do so. It only takes a minute to add a new team. Team settings are managed by the team administrator role. Users assigned as team administrator can configure and manage all team tools.

Team administrators should do the following tasks:

- [Add team members](#)
- [Add another team administrator](#)
- [Configure areas and iteration paths](#)
- [Configure backlogs, boards, and general settings](#)

Also, consider the following optional tasks:

- [Configure and manage team dashboards](#)
- [Configure team notifications](#)

## Prerequisites

- To perform any team configuration task, you need to be added as a team administrator for the team to be modified, or be a member of the **Project Administrators** group. See [Change project-level permissions](#).
- To add a team, you must be a member of the **Project Administrators** group. For more information, see [Add teams](#).

### NOTE

For guidance on configuring and customizing your project and teams to support your business needs, review [Configuration and customization of Azure Boards](#).

## Open your team profile

Open your team profile to quickly access items defined for your team.

1. Sign in to your organization (<https://dev.azure.com/{yourorganization}>), and then open your project.
2. Select **Project settings > Teams > your team name**.

The screenshot shows the 'Project Settings' page for the 'Fabrikam Fiber' project in Azure DevOps. A red circle labeled '1' highlights the 'Project settings' button at the bottom of the left sidebar. A red circle labeled '2' highlights the 'Teams' section in the main content area. A red circle labeled '3' highlights the 'Fabrikam Fiber Team' card in the 'Teams' list on the right.

## Add users to a team

Several tools, such as capacity planning, team alerts, and dashboard widgets, are team-scoped. These tools automatically reference the users that are as members of a team to support planning activities or sending alerts.

To add users to a team, see [Add users to a project or specific team](#).

The screenshot shows the 'Fabrikam Fiber Team' settings page. The left sidebar displays the team profile with a purple icon, name ('Fabrikam Fiber Team'), and description ('The default project team.'). The right panel shows the 'Members' section with a table. A red box highlights the '+ Add...' button. The table lists two members:

Display Name	Username Or Scope
Security Service Group	[FabrikamFiber01]
Fabrikam Fiber Build Servic...	Build\99612ac2-9c5f-4191-9e8...
Jamal Hartnett	fabrikamfiber4@hotmail.com

All members of a team can favorite team artifacts and define work item templates. For more information, see:

- [Set personal or team favorites](#)

- Use templates to add and update work items.

If team members don't have access to all the features they want, make sure they have [the permissions needed for those features](#).

## Add an administrator

When you add a team to a project, a Project Administrator should [add one or more team administrators](#).

The screenshot shows the 'Fabrikam Fiber Team' settings page. On the left, there's a 'Team Profile' section with a purple icon, a name ('Fabrikam Fiber Team'), a description ('The default project team.'), and an 'Administrators' list containing 'Jamal Hartnett' with a red 'X'. Below this is a '+ Add' button, which is also highlighted with a red box. On the right, under the 'Members' heading, there's a table showing three entries:

Display Name	Username Or Scope
Security Service Group	[FabrikamFiber01]
Fabrikam Fiber Build Servic...	Build\99612ac2-9c5f-4191-9e8...
Jamal Hartnett	fabrikamfiber4@hotmail.com

## Configure team areas and iterations

Many Agile tools depend on the area and iteration paths that are configured for the team. To learn more about configuring team areas and iterations, see [About teams and Agile tools](#).

Once project administrators [add area paths](#) and [iteration paths](#) for a project, team administrators can select the area and iteration paths associated with their team. These settings affect many Agile tools available to the team.

**Team Profile**



Name  
Fabrikam Fiber Team

Description  
The default project team.

Administrators  
Jamal Hartnett X

+ Add

Manage other settings for this team

**Notifications**

**Dashboards**

**Iterations and areas**

Fabrikam Fiber Team

**Members**

+ Add... | ○

Display Name	Username Or Scope
 Security Service Group	[FabrikamFiber01]
 Fabrikam Fiber Build Servic...	Build\99612ac2-9c5f-4191-9e8...
 Jamal Hartnett	fabrikamfiber4@hotmail.com

Settings include making the following associations for each team:

- **Select team Area Paths**  
Can select the default area path(s) associated with the team. These settings affect many Agile tools available to the team.
- **Select team Iteration Paths or sprints** Can select the default area path(s) associated with the team. These settings affect many Agile tools available to the team.

For more information, see [Define area paths and assign to a team](#) and [Define iteration paths and configure team iterations](#).

## Configure team backlogs, boards, and general settings

Team administrators can choose which backlog levels are active for a team. For example, a feature team may choose to show only the product backlog and a management team may choose to show only the feature and epic backlogs. Also, administrators can choose whether bugs are treated similar to user stories and requirements or as tasks.

Team administrators can also choose which days are non-working days for the team. Sprint planning and tracking tools automatically consider days off when calculating capacity and sprint burndown.

You can configure most of your team settings from the common configuration dialog.

### NOTE

To understand the differences between backlogs, boards, taskboards, and Delivery plans, see [Backlogs, boards, and plans](#). If your backlog or board doesn't show the work items that you expect or want, see [Set up your backlogs and boards](#).

1. Check that you selected the correct project, and then choose **Boards > Boards**, and select the correct team from the team selector dropdown menu. For more information, see [Use breadcrumbs and selectors to navigate and open artifacts](#).

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. On the left, a sidebar lists various project management sections: Overview (1), Boards (highlighted with a red box), Work items, Boards (highlighted with a red box) (2), Backlogs, Sprints, Queries, Repos, Pipelines, Test Plans, and Artifacts. The main area is titled 'Fabrikam Fiber Team' (3). It displays a Kanban board with two columns: 'To Do' and 'Doing'. Under 'To Do', there is a card for '1 Bug' in state 'To Do'. A 'New item' button is also present.

2. Choose **Team settings** to configure the board and set general team settings.

The screenshot shows the top navigation bar of the Azure DevOps interface for the 'Fabrikam Fiber Team'. The bar includes buttons for 'New Work Item', 'View as Board', 'Column Options', '...', 'Issues', and 'General'. The 'General' button (represented by a gear icon) is highlighted with a red box.

3. Choose a tab under any of the sections—**Cards**, **Board**, **Charts**, and **General**—to configure the cards or boards, the cumulative flow chart, or other team settings. When you're done configuring the settings, select **Save and close**.

Settings

Cards

**Fields \***

- Styles
- Tag colors
- Annotations
- Tests

Board

- Columns
- Swimlanes
- Card reordering
- Status badge

Charts

- Cumulative Flow

General

- Backlogs
- Working days

Fields

Show the important information to your team. Fields are editable directly on the card.

**Issue**

**Core fields**

Show ID

Show Assigned To as:

Avatar and full name (default)

Show Effort

Show Tags

**Additional fields**

Add up to 10 fields in the order that you want them to appear on the card.

+ Field

State

Show empty fields

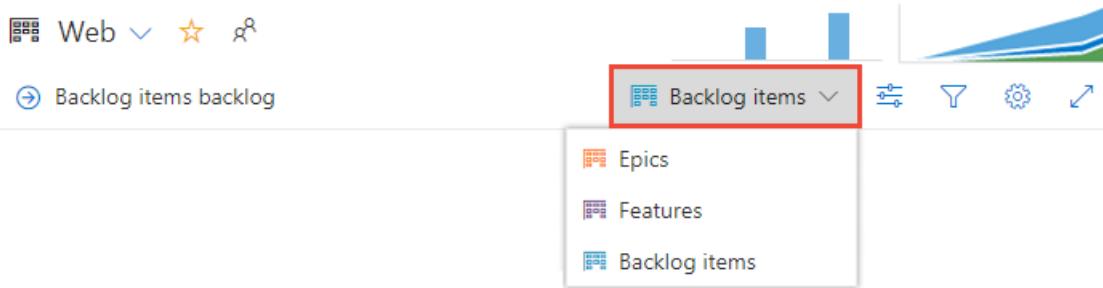
Check if you want to display fields, even when they are empty.

**Save and close**    **Cancel**

- Check that you selected the right project, (2) choose Boards > Boards, and then (3) select the correct team from the team selector menu.

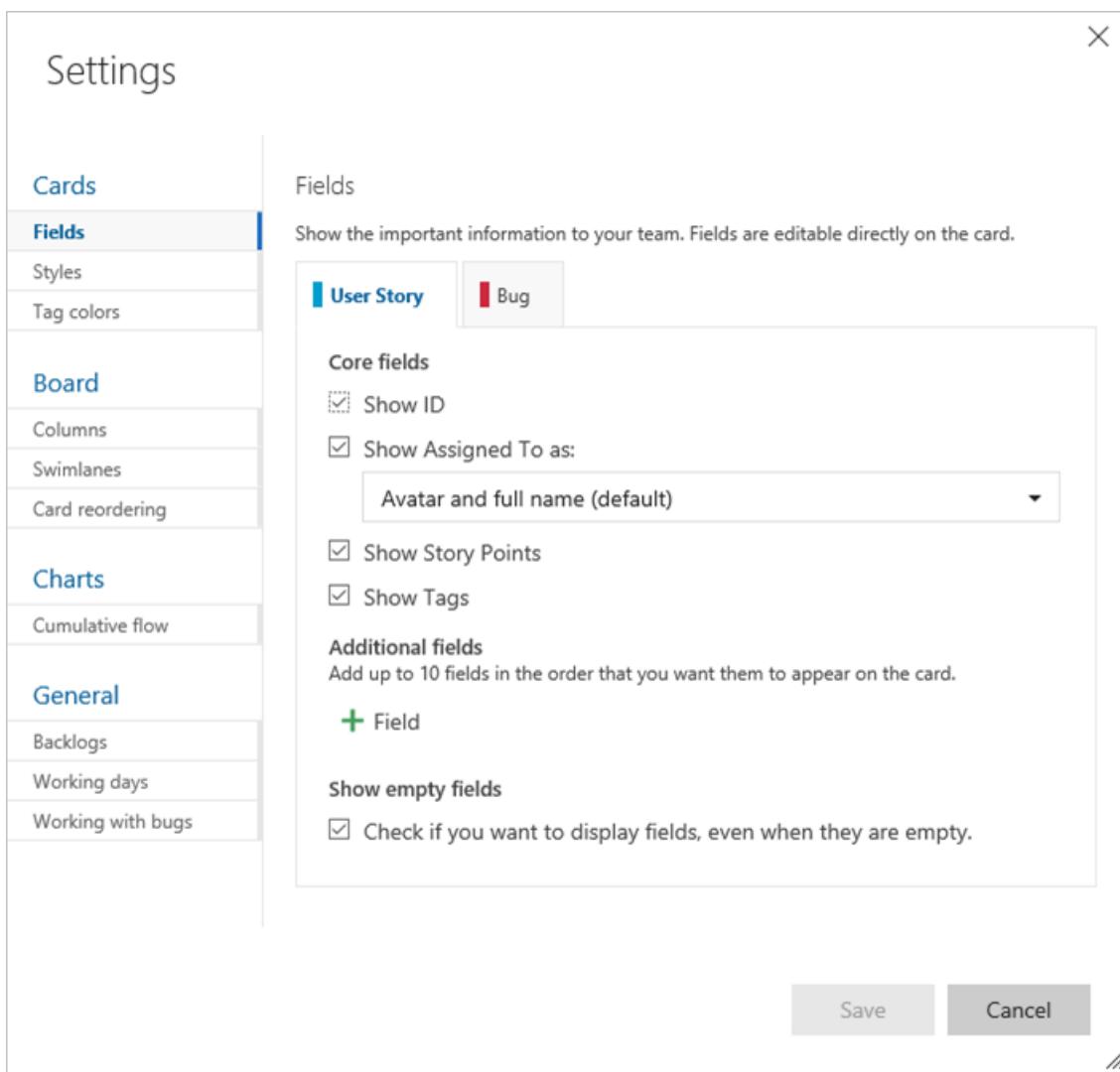
The screenshot shows the Azure DevOps interface. At the top left is the 'Azure DevOps' logo. Next to it is a dropdown menu 'fabrikam / Fabrikam Fiber' with a red box around it and a circled '1'. Below the logo is a breadcrumb path 'Fabrikam Fiber Team' with a red box around it and a circled '3'. On the far right is a 'Backlog items' button. The left sidebar has several options: 'Overview', 'Boards' (which is selected and highlighted with a red box and circled '2'), 'Work Items', 'Backlogs', and 'Sprints'. The main area displays a 'Backlog' board with columns 'Backlog', 'Analyze', 'Develop', and 'Slow form'. A 'New item' button is at the top left of the backlog column. Below it is a card for 'Add an information form' by 'Raisa Pokrovskaya' in 'Iteration ... Sprint 3'. The 'Develop' column shows '5/10' items. The 'Slow form' column shows two items: 'Slow form' by 'Johnnie McLeod' in 'Iteration ... Sprint 3' and 'Slow form' by 'Jam' with '0/2' items.

- Make sure that you select the team backlog or board that you want to configure using the team selector. To learn more, see [Use breadcrumbs and selectors to navigate and open artifacts](#).
- Choose the product or portfolio backlog from the board-selection menu.



4. Choose **Team settings**  to configure the board and set general team settings.

5. Choose a tab under any of the sections—**Cards**, **Board**, **Charts**, and **General**—to configure the cards or boards, the cumulative flow chart, or other team settings.



1. Make sure that you select the team from the project/team selector. You can switch your team focus to one that you've recently viewed from the project/team selector. If you don't see the team or project you want, choose **Browse...** or choose **Azure DevOps**  to access the **Projects** page.

Fabrikam Fiber Home

Recent projects/teams

- Agile 11
- FabrikamFiber
- Fabrikam Fiber A
- Fabrikam Fiber PB
- Browse...
- New team

A README.md file is intended to quickly orient readers to what your project can do.  
Learn more about Markdown.

2. Open Work > Backlogs > Board.

Work Items\* Backlogs Queries Plans

Epics Features Stories

Past Current Sprint 5

Stories

Backlog Board

Backlog Active 6/5 Resolved

<a href="#">New item</a>	<a href="#">Cancel order form</a>	<a href="#">Implement a factory abstracts</a>
<a href="#">Slow response on form</a>	<a href="#">Jamal Hartnett</a>	<a href="#">Jamal Hartnett</a>
<a href="#">Christie Church</a>	13	0/1
<a href="#">Add animated emoticons</a>	<a href="#">Phone Service Web</a>	<a href="#">Bug 6</a>
<a href="#">Christie Church</a>	3	<a href="#">Raisa Pokrovskaya</a>
<a href="#">Welcome back page</a>	3	<a href="#">Raisa Pokrovskaya</a>
<a href="#">Welcome back page</a>		

3. Choose the board you want to configure and then choose Team settings to configure the board and set general team settings.

For example, from the Kanban board ...

Stories

Backlog Board

Backlog Analyze Develop Test

4/10 4/5

4. Choose a tab under Cards or Board to configure the cards and Kanban board columns and swimlanes.

![Common configuration dialog team settings]../../boards/boards/media/customize-cards/common-config-141.png)

Team administrators can fully customize the team's Kanban boards associated with the product and portfolio backlogs. You configure a Kanban board by first defining the columns and WIP limits from the common configuration dialog. For guidance, see [Kanban basics](#).

For more information on each configuration option, see the following articles:

---

## General

- [Backlogs](#)
- [Working days](#)
- [Working with bugs](#)

## Cards

- [Add fields](#)
- [Define styles](#)
- [Add tag colors](#)
- [Enable annotations](#)
- [Configure inline tests](#)

## Boards

- [Add columns](#)
- [Split columns](#)
- [WIP limits](#)
- [Definition of Done](#)
- [Add swimlanes](#)
- [Card reordering](#)
- [Configure status badges](#)
  
- [Add columns](#)
- [Split columns](#)
- [WIP limits](#)
- [Definition of Done](#)
- [Add swimlanes](#)
- [Card reordering](#)

## Chart

- [Configure cumulative flow chart](#)

---

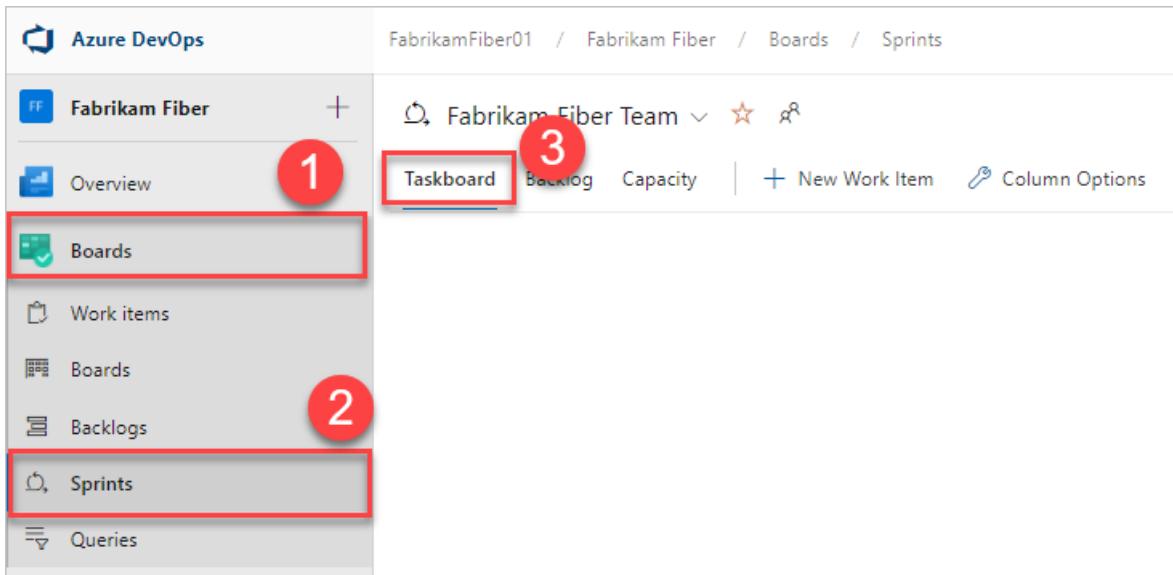
## Kanban board

Backlog	Active	Resolved	Closed
<div><a href="#"> New item</a>  <div><a href="#"> 532 Hello World Web Site</a>  Jamal Hartnett  Phone Service Web   0/1</div> <a href="#"> 398 Cancel order form</a>  Jamal Hartnett  13</div>	<div><a href="#"> 486 Welcome back page</a>  Raisa Pokrovskaya 3</div> <div><a href="#"> 346 Add animated emoticons</a>  Christie Church 3</div> <div><a href="#"> Slow response on form</a>  Christie Church 8</div>	<div><a href="#"> 344 Implement a factory which abstracts</a>  Jamal Hartnett 8</div> <div> 0/1</div> <div><a href="#"> 405 GPS locator</a>  Jamal Hartnett 8</div>	

## Configure sprint Taskboards

Similar to Kanban boards, each sprint Taskboard can be customized to support information-rich, color-coded cards as well as addition of customized columns. For details, see [Customize sprint Taskboards](#).

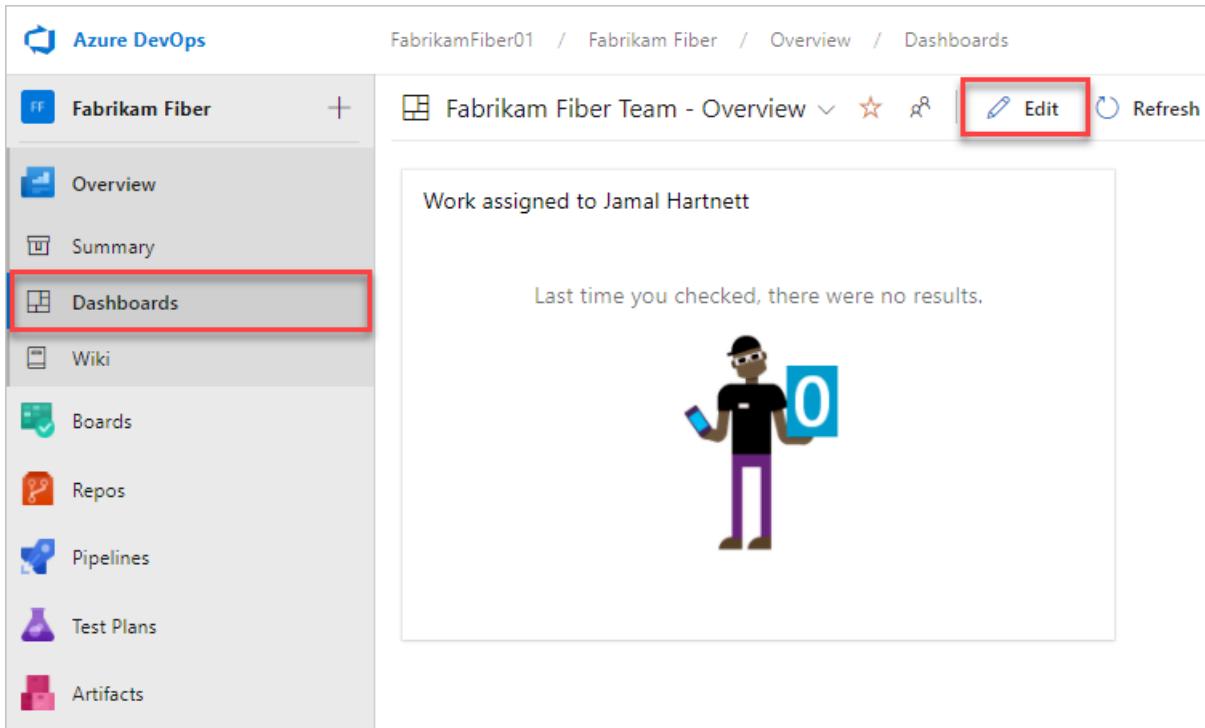
Similar to Kanban boards, each sprint Taskboard can be customized to support information-rich, color-coded cards. For details, see [Customize sprint Taskboards](#).



## Add and manage team dashboards

By default, all team members can add and edit team dashboards. In addition, team administrators can manage permissions for team dashboards. For details, see [Add and manage dashboards](#).

Team administrators can add, configure, and manage permissions for team dashboards. For details, see [Add and manage dashboards](#).



## Update team name, description, and image

Team settings also include the team name, description, and team profile image. To add a team picture, select the image icon. The maximum file size is 2.5 MB.

The screenshot shows the 'Team Image' section of the Azure DevOps 'Settings' page for a team named 'Azure D Boards'. It includes a placeholder image, a file upload button ('Upload image'), and a note about rights and terms. A red box highlights the 'Upload image' button.

Azure D Boards

Azure DevOps Agile, Report, Security, Navigation content areas

Relevant links: Notifications | Dashboards | Iterations and Area Paths

Members Settings

Team Image

Select an image file on your computer (2.5MB max)

+ Upload image Reset

By uploading a file you certify that you have the right to distribute  
① this picture and you agree to the [Terms of Service](#), [Privacy Statement](#),  
and [Code of Conduct](#).

Team settings also include the team name, description, and team profile image. To add a team picture. Open the Team Profile and choose the picture icon. The maximum file size is 4 MB.

## Manage notifications

Team administrators can add and modify alerts so that the team can receive email notifications as changes occur to work items, code reviews, source control files, and builds. Many alerts are defined for each team. For details, see [Manage team alerts](#).

The screenshot shows the 'Team Profile' section of the Azure DevOps 'Teams' page for the 'Fabrikam Fiber Team'. It displays the team name, members, and various settings. A red box highlights the 'Notifications' link in the sidebar.

/ Settings / Teams / Fabrikam Fiber Team

Team Profile

Fabrikam Fiber Team

Members

+ Add... |

Display Name	Username Or Scope
Security Service Group	[FabrikamFiber01]
Fabrikam Fiber Build Servic...	Build\99612ac2-9c5f-4191-9e8...
Jamal Hartnett	fabrikamfiber4@hotmail.com

Name

Fabrikam Fiber Team

Description

The default project team.

Administrators

Jamal Hartnett

+ Add

Manage other settings for this team

Notifications

Dashboards

Iterations and areas

## Related articles

- [About projects and scaling your organization](#)

- [About teams and Agile tools](#)
- [Add teams](#)
- [Add a team administrator](#)

# Request an increase in permission levels

12/13/2022 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To get access to certain tasks, you might need to request an increase to your permissions or be added to a security role. Typically, you'll do this upon receiving an information or error message indicating you have insufficient permissions. Such a message will indicate the permission levels you need.

Prior to requesting a change in permission levels, make sure you understand the basics by reviewing [Get started with permissions, access, and security groups](#).

## Common permissions to request

Most users added to the **Contributors** group are granted the permissions they need to perform most tasks. However, the following tasks require membership in the **Project Administrators** group or a change in permissions.

- **Work tracking**
  - Add or change **Area Paths** or **Iteration Paths**: Requires elevated permissions to an Area Path or Iteration Path node. To learn more, see [Set work tracking permissions](#), [Create child nodes](#).
  - Create shared queries or query folders: Requires elevated permissions set for a shared query folder. To learn more, see [Set work tracking permissions](#), [Set permissions on queries or query folders](#).
  - Change team settings—such as Kanban board settings: Requires addition as a team administrator. To learn more, see [Add or remove a team administrator](#)
- **Source code, Git repositories**, the following tasks require elevated permissions for Git repositories or a specific repository. To learn more, see [Set Git repository permissions](#).
  - Create, delete, or rename a Git repository
  - Manage repository permissions
  - Bypass policies

The following tasks require membership in the **Project Collection Administrators** group or a change in permissions at the collection-level or addition to a specific role.

- **Collection-level configurations**
  - Create projects: Requires elevated permissions at the [collection level](#).
  - Add, edit, or manage a process: Requires elevated permissions at the collection level or [process-level permissions](#).
  - Install, uninstall, or disable extensions: Requires addition to the [Manager role](#) for extensions.

For an overview of built-in security groups and default permission assignments, see [Default permissions and access](#).

## Prerequisites

- To view permissions, you must be a member of the **Project Valid Users** group. Users added to a project are automatically added to this security group. To learn more, see [View permissions for yourself or others](#).
- To look up an administrator for your project or project collection, you must be a member of the **Project Valid Users** group.

#### **NOTE**

Users added to the Project-Scoped Users group won't be able to access **Organization Settings** other than the **Overview** section if the **Limit user visibility and collaboration to specific projects** preview feature is enabled for the organization. To learn more, see [Manage your organization, Limit user visibility for projects and more](#).

## Review your permission assignments

Before you request a change to permission levels, review your permission assignments as described in [View permissions for yourself or others](#).

Verify that your permission assignments are preventing you from accomplishing a task you need to perform.

## Request a change to a permission level or role change

To request a change or increase in your permission levels, take the following actions:

1. Identify the permissions you need and at what level. Permissions are set at the object, project, and project-collection level. Also, permissions are granted through various roles. To identify the level and permission you need, review the [Permissions lookup guide](#).
2. Identify a person in your organization who can grant you the permissions you need. For example:
  - To get permissions to manage team settings, [identify the team administrator for your team](#) or a [member of the Project Administrators group](#).
  - To change an object-level permission, identify the owner of the object or a member of the [Project Administrators group](#). To learn how, see [Set object-level permissions](#).
  - To change a project-level permission, identify a member of the [Project Administrators group](#). See [Look up a project administrator](#).
  - To change a project collection-level permission, identify a member of the [Project Collection Administrators group](#). See [Look up a project collection administrator](#).
3. Contact the person you identified in step 2 and make your request. Make sure you specify the permission you want changed.

## Refresh or re-evaluate your permissions

After your permission levels are changed, you may need to refresh your permissions for Azure DevOps to recognize the changes. This step is recommended when a change is made to your permission level, role level, or if you are added to a new or different Azure DevOps, Azure Active Directory, or Active Directory security group. When you are added to a new or different security group, your inherited permissions may change.

By refreshing your permissions, you cause Azure DevOps to re-evaluate your permission assignments. Otherwise, your permission assignments won't be refreshed until you sign-off, close your browser, and sign-in again.

To refresh your permissions, choose **User settings**, on the **Permissions** page, you can select **Re-evaluate permissions**. This function reevaluates your group memberships and permissions, and then any recent changes take effect immediately.

## User settings

Account

Profile

Time and Locale

Permissions

Preferences

Notifications

Theme

## Permissions

Re-evaluate permissions

This tool will re-evaluate your group memberships and permissions; any recent changes will take effect immediately. Normally, changes to group memberships and permissions can take up to one hour to reflect in Azure DevOps.

## Related articles

- [Permissions lookup guide](#)
- [Default permissions and access](#)
- [Troubleshoot permissions](#)
- [Look up a project administrator](#)
- [Look up a project collection administrator](#)

# Grant or restrict access using permissions

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can grant or restrict access to resources that you manage in Azure DevOps. You may want to open up or close down access to a select set of features and for a select set of users. While the built-in security groups provide a standard set of permission assignments, you may need additional security requirements not met by these assignments.

If you're new to administrating permissions and groups, review [Get started with permissions, access, and security groups](#)to learn about permission states and inheritance.

In this article you learn how to do the following tasks:

- Recommended method for granting and restricting permissions
- Delegate tasks by assigning select permissions to specific roles
- Limit visibility to organization information
- Limit people picker to project users and groups
- Restrict access to view or modify objects
- Restrict modification of work items based on a user or group
- Recommended method for granting and restricting permissions
- Delegate tasks by assigning select permissions to specific roles
- Restrict access to view or modify objects
- Restrict modification of work items based on a user or group

## TIP

Because you set many permissions at an object-level, such as repositories and area paths, how you structure your project determines the areas you can open up or close down.

## Recommended method for granting and restricting permissions

For maintenance purposes, we recommend you use either the built-in security groups or [custom security groups to manage permissions](#).

You can't change the permission settings for the Project Administrators group or the Project Collection Administrators group, which is by design. However, for all other groups, you can change the permissions.

If you manage a small number of users, then you may find changing individual permissions a valid option. However, custom security groups allow you to better track roles and permissions assigned to those roles.

## Delegate tasks to specific roles

As an administrator or account owner, it's a good idea to delegate administrative tasks to those team members who lead or manage an area. Several of the main built-in roles that come with default permissions and role assignments are:

- Readers
- Contributors

- Team Administrator (role)
- Project Administrators
- Project Collection Administrators

For a summary of permissions for the above roles, see [Default permissions and access](#), or for the Project Collection Administrators, see [Change project collection-level permissions](#).

To delegate tasks to other members within your organization, consider creating a custom security group and then granting permissions as indicated in the following table.

## Role

### Tasks to perform

#### Permissions to set to Allow

Development lead (Git)

Manage branch policies

Edit policies, Force push, and Manage permissions

See [Set branch permissions](#).

Development lead (TFVC)

Manage repository and branches

Administer labels, Manage branch, and Manage permissions

See [Set TFVC repository permissions](#).

Software architect (Git)

Manage repositories

Create repositories, Force push, and Manage permissions

See [Set Git repository permissions](#)

Team administrators

Add area paths for their team

Add shared queries for their team

Create child nodes, Delete this node, Edit this node See [Create child nodes, modify work items under an area path](#)

Contribute, Delete, Manage permissions (for a query folder), See [Set query permissions](#).

Contributors

Add shared queries under a query folder, Contribute to dashboards

Contribute, Delete (for a query folder), See [Set query permissions](#)

View, Edit, and Manage dashboards, See [Set dashboard permissions](#).

Project or product manager

Add area paths, iteration paths, and shared queries

Delete and restore work items, Move work items out of this project, Permanently delete work items

Edit project-level information, See [Change project-level permissions](#).

Process template manager ([Inheritance process model](#))

Work tracking customization

Administer process permissions, Create new projects, Create process, Delete field from account, Delete process, Delete project, Edit process

See [Change project collection-level permissions](#).

Process template manager ([Hosted XML process model](#))

Work tracking customization

Edit collection-level information, See [Change project collection-level permissions](#).

Project management ([On-premises XML process model](#))

Work tracking customization

Edit project-level information, See [Change project-level permissions](#).

Permissions manager

Manage permissions for a project, account, or collection

For a project, Edit project-level information

For an account or collection, Edit instance-level (or collection-level) information

To understand the scope of these permissions, see [Permission lookup guide](#). To request a change in permissions,

See [Request an increase in permission levels](#).

You can also grant permissions to manage permissions for the following objects:

- [Set Git repository permissions](#)
- [Manage Git branch permissions](#)
- [Set TFVC repository permissions](#)
- [Administer build and release permissions](#)
- [Manage Wiki permissions](#).

## Limit visibility to organization and project information

By default, users added to an organization can view all organization and project information and settings. To restrict access to only those projects that you add users to, you can enable the **Limit user visibility and collaboration to specific projects** preview feature for the organization. To enable this feature, see [Manage or enable features](#).

With this feature enabled, users added to the **Project-SScoped Users** group can't view most **Organization settings** and can only connect to those projects to which they've been added.

## Limit people picker to project users and groups

For organizations that manage their users and groups using Azure Active Directory (Azure AD), people pickers provide support for searching all users and groups added to Azure AD, not just those added to a project. People pickers support the following Azure DevOps functions:

- Selection of a user identity from a work tracking identity field such as **Assigned To**
- Selection of a user or group using **@mention** in a work item discussion or rich-text field, a pull request discussion, commit comments, or changeset or shelveset comments
- Selection of a user or group using **@mention** from a wiki page

As shown in the following image, you simply start typing into a people picker box until you find a match to a user name or security group.

## Discussion

The screenshot shows a search interface for users. At the top, there's a placeholder text '@C'. Below it, two user entries are listed: 'Christie Church' with the email 'fabrikamfiber1@hotmail.com' and 'Chuck Reinhart' with the email 'fabrikamfiber3@hotmail.com'. Each entry has a small 'RE' icon to its right. Below the list is a search bar with the placeholder 'Search' and a note 'Showing 2 results'.

Users and groups who are added to the **Project-SScoped Users** group can only see and select users and groups in the project they are connected to from a people picker. To scope people pickers for all project members, see [Manage your organization, Limit identity search and selection](#).

## Restrict access to view or modify objects

Azure DevOps is designed to enable all valid users to view all objects defined in the system. You can restrict access to resources by setting the permission state to **Deny**. You can set permissions for members that belong to a custom security group or for an individual user. To learn more about how to set these types of permissions, see [Request an increase in permission levels](#).

### Area to restrict

#### Permissions to set to Deny

View or contribute to a repository

View, Contribute

See [Set Git repository permissions](#) or [Set TFVC repository permissions](#).

View, create, or modify work items within an area path

Edit work items in this node, View work items in this node

See [Set permissions and access for work tracking](#), [Modify work items under an area path](#).

View or update select build and release pipelines

Edit build pipeline, View build pipeline

Edit release pipeline, View release pipeline

You set these permissions at the object level. See [Set build and release permissions](#).

Edit a dashboard

View dashboards

See [Set dashboard permissions](#).

## Restrict modification of work items or select fields

For examples that illustrate how to restrict modification of work items or select fields, see [Sample rule scenarios](#).

## Next steps

[Remove user accounts](#)

## Related articles

- [Troubleshoot permissions](#)
- [Rules and rule evaluation](#)
- [Default permissions and access](#)
- [Permission lookup guide](#)
- [Get started with permissions, access, and security groups](#)
- [Permissions and groups reference](#)
- [Change project-level permissions](#)
- [Change project collection-level permissions](#)

# Security best practices

12/13/2022 • 14 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Security should always be your topmost concern when you're working with information and data, especially when you're working in a cloud-based solution, like Azure DevOps Services. Microsoft keeps the underlying cloud infrastructure secure, but it's up to you to configure security in Azure DevOps.

You don't have to implement best practices when using Azure Devops, but doing so will likely help you have a better, more secure experience. We've gathered some best practices for keeping your Azure DevOps environment secure, with the following goals in mind:

- Properly scope [service accounts](#), [service connections](#), and [permissions](#)
- Maintain tight control of administrators and service account groups
- Manage security with security groups, policies, and settings at the organization/collection, project, or object level
- Secure services, like [Azure Artifacts](#), [Azure Boards](#), [Azure Pipelines](#), [Azure Repos](#), [Azure Test Plans](#), and [Azure DevOps in general](#).

## Scope service accounts

- Ensure [service accounts](#) have zero interactive sign-in rights.
- Restrict service account privileges to the bare minimum necessary.
- Use a different identity for the report reader account, if you use domain accounts for your service accounts. For more information, see [Service accounts and dependencies](#).
- Use local accounts for user accounts, if you're installing a component in a workgroup. For more information, see [Service account requirements](#).
- Use [service connections](#) when possible. Service connections provide a secure mechanism to connect to assorted services without the need to pass in secret variables to the build directly. - Restrict these connections to the specific place they should be used and nothing more.
- Monitor service account activity and create [audit streaming](#). Auditing allows you to detect and react to suspicious sign-ins and activity.

For more information, see [Common service connection types](#).

## Scope service connections

- Scope [Azure Resource Manager](#), and [other service connections](#), only to the resources and groups to which they need access. Service connections shouldn't have broad contributor rights on the entire Azure subscription.
- Don't give users generic or broad contributor rights on the Azure subscription.
- Don't use Azure Classic service connections, as there's no way to scope the permissions.
- Make sure the resource group only contains Virtual Machines (VMs) or resources that the build needs access to.
- Use a purpose-specific team service account to authenticate a service connection.

For more information, see [Common service connection types](#).

- Disable Personal Access Token (PAT)-based authentication, so the OAuth flow gets used with the GitHub service connection.
- Never authenticate GitHub service connections as an identity that's an administrator or owner of any repositories. [Check your policies](#).
- Never use a full-scope GitHub PAT (Personal Access Token) to authenticate GitHub service connections.
- Don't use a personal GitHub account as a service connection with Azure DevOps.

## Scope permissions

The system manages permissions at different levels - individual, external, server, collection, project, object, and - and assigns them to one or more built-in groups by default.

### Individual permissions

For more information, see [Set individual permissions](#).

### External guest access

- Block external guest access entirely by disabling the "[Allow invitations to be sent to any domain](#)" policy. It's a good idea to do so if there's no business need for it.
- Use a different email or user principal name (UPN) for your personal and business accounts, even though it's allowed. This action eliminates the challenge of disambiguating between your business and personal accounts when the email/UPN is the same.
- Put all the external guest users in a single Azure AD group and manage the permissions of that group appropriately. You can easily manage and audit this way.
  - Remove direct assignments so the group rules apply to those users. For more information, see [Add a group rule to assign access levels](#).
  - Reevaluate rules regularly on the Group rules tab of the Users page. Clarify whether any group membership changes in Azure AD might affect your organization. Azure AD can take up to 24 hours to update dynamic group membership. Every 24 hours and anytime a group rule changes, rules get automatically reevaluated in the system.

For more information, see [B2B guests in the Azure AD](#).

## Manage security groups, policies, and settings

### Security and user groups

See the following recommendations for assigning permissions to security groups and users groups.

DO ✓	DON'T ✗
Use Azure Active Directory, Active Directory, or Windows security groups when you're managing lots of users.	Don't change the default permissions for the Project Valid Users group. This group can access and view project information.
When you're adding teams, consider what permissions you want to assign to team leads, scrum masters, and other team members who may need to create and modify area paths, iteration paths, and queries.	Don't add users to multiple security groups that contain different permission levels. In certain cases, a <i>Deny</i> permission level may override an <i>Allow</i> permission level.
When you're adding many teams, consider creating a <i>Team Administrators</i> custom group where you allocate a subset of the permissions available to <i>Project Administrators</i> .	Don't change the default assignments made to the valid users groups. If you remove or set the <i>View instance-level information</i> permission to <i>Deny</i> for one of the <i>Valid Users</i> groups, no users in the group can access the project, collection, or deployment, depending on the group you set.

DO ✓	DON'T ✗
Consider granting the work item query folders <i>Contribute</i> permission to users or groups who require the ability to create and share work item queries for the project.	Don't assign permissions that are noted as <i>Assign only to service accounts</i> to user accounts.
Keep groups as small as possible. Access should be restricted, and the groups should be frequently audited.	
Take advantage of built-in roles and default to Contributor for developers. Admins get assigned to the Project Administrator security group for elevated permissions, allowing them to configure security permissions.	

## Server-level groups

See the following table of built-in security groups, which users to add, and best practice tips.

### Built-in security group

#### Add these users

#### Best practice tips

##### Team Foundation Administrators

People who need to perform all server-level operations.

This group should be restricted to the smallest possible number of users who need total administrative control over server-level operations. If your deployment uses SharePoint or Reporting, consider adding the members of this group to the Farm Administrators and Site Collection Administrators groups in SharePoint and the Team Foundation.

##### Team Foundation Valid users

People who need to view server instance-level information.

This group contains all users known to exist in the server instance. You can't modify the membership of this group.

## Project-level permissions

- Limit access to projects and repos to reduce the risk of leaking sensitive information and deploying insecure code to production.
- Use either the built-in security groups or custom security groups to manage permissions. For more information, see [Grant or restrict permissions to select tasks](#).

### Built-in security group

#### Add these users

#### Best practices

##### Project Collection Administrators

People who need total administrative control over the collection.

This group should be restricted to the smallest possible number of users who need total administrative control over the collection. For Azure DevOps, assign to administrators who customize work tracking.

If your deployment uses Reporting Services, consider adding the members of this group to the Team Foundation Content Managers groups in Reporting Services

---

#### Project Collection Build Administrators

People who need to administer build resources and permissions for the collection.

Limit this group to the smallest possible number of users who need total administrative control over build servers and services for this collection.

---

#### Project-scoped users

People who need limited access to view organization settings and projects other than those projects they're specifically added to.

Add users to this group when you want to limit their visibility and access to those projects that you explicitly add them to. Do not add users to this group if they are also added to the Project Collection Administrators group.

---

#### Removing users

- If your organization uses MSA accounts, then remove inactive users directly from the organization, as you have no other way to prevent access. When you do so, you can't create a query for work items assigned to the removed user account. For more information, see [Delete users from Azure DevOps](#).
- If your organization is backed by Azure AD, then you can disable or delete the Azure AD user account while leaving their Azure DevOps account active. In this way, you can continue to query their work item history using their account name.
- [Revoke user PATs](#).
- Revoke any special permissions that may have been granted to individual user accounts.
- Reassign work from users you're removing to current team members.

## Choose the right authentication method

Select your [authentication methods](#) from the following sources:

- [Multi-factor authentication](#)
- [Azure Active Directory \(Azure AD\)](#)
- [Personal access tokens \(PATs\)](#)

#### Require multi-factor authentication

Require all users to use multi-factor authentication (MFA), as a basic security feature. Multi-factor authentication requires use of more than one verification method, which adds a second layer of security to all Azure DevOps transactions.

#### Use Azure AD

Integrate Azure DevOps with Azure AD to have a single plane for identity. Consistency and a single authoritative source increases clarity and reduces security risks from human errors and configuration complexity. The key to end to end governance is to have multiple role assignments (with different role definitions and different resource scopes to the same Azure AD groups). Without Azure AD, you're solely responsible for controlling organization access.

For more information, see the following articles:

- [About accessing your organization with Azure AD](#)
- [Add AD/Azure AD users or groups to a built-in security groups](#)

#### Use PATs seldomly

Always authenticate with identity services rather than cryptographic keys when available. Managing keys securely with application code is difficult and regularly leads to mistakes like accidentally publishing sensitive access keys to code repositories like GitHub. But, if you're using PATs, see the following recommendations:

- Administrators should audit all PATs using the [REST APIs](#) and revoke any PATs that don't meet the following criteria for PATs in use:
  - Should always be scoped (roles).
  - Shouldn't be global (can access more than one organization).
  - Shouldn't allow write or manage permissions on build or releases.
  - Should have an expiration date.
  - Should be kept secret. Your tokens are as critical as passwords.
  - Should have an expiration date.
  - Shouldn't be hardcoded. It can be tempting to simplify code to get a token for a prolonged period and store it in your application, but don't do that. They could end up in source code that could be stolen.
- Keep your PATs secret. Your tokens are as critical as passwords.
- Store your tokens in a safe place.
- Don't hard code tokens in applications. It can be tempting to simplify code to obtain a token for a long period of time and store it in your application, but don't do that.
- Give tokens an expiration date.

For more information, check out the following articles:

- [Manage PATs with policies - for administrators](#)
- [Use PATs](#)

## Limit access by location

Limit access to specific IP (Internet Protocol) address ranges with Azure AD Conditional Access Policy Validation. For example, you can configure a location so that MFA isn't required for internal IP addresses.

For more information, see [Using the location condition in a Conditional access policy](#).

## Secure your network

Set up an [allowlist](#).

## Use Web application firewalls

Implement Web application firewalls (WAFs), so they can filter, monitor, and block any malicious web-based traffic to and from Azure DevOps.

- Always use encryption.
- Validate certificates.
- This shouldn't be the only planned safety mechanism to reduce the volume and severity of security bugs in your applications.

For more information, see [Application management best practices](#)

## Secure projects

- Enable the *Limit user visibility for projects* preview feature for the organization, which restricts access to only those projects that you add users to.
- Add users to the *Project-scoped users* group, so they can only see and select users and groups in the project that they're connected to from a people picker.

- Disable "Allow public projects" in your organization's policy settings to prevent every organization user from creating a public project. Azure DevOps Services allows you to change the visibility of your projects from public to private, and vice-versa. If users haven't signed into your organization, they have read-only access to your public projects. If users have signed in, they can be granted access to private projects and make any permitted changes to them.
- Don't allow users to create new projects. Use EasyStart "Governed Projects," which require approval once they're submitted.
- Check out the following articles for more in-depth information about setting sub-project permissions.
  - [Set wiki permissions](#)
  - [Set feedback permissions](#)
  - [Set dashboard permissions](#)
  - [Set Analytics permissions](#)

## Secure Azure Artifacts

Make sure you understand the difference between feeds, project, and project collection administrators. For more information, see [Configure Azure Artifacts settings](#). For more information, see [Set feed permissions](#).

## Secure Azure Boards

- Review [Configure and customize Azure Boards](#) before you customize a process.
- See the following articles:
  - [Set work tracking and plan permissions](#)
  - [Default permissions and access to Azure Boards](#)
  - [Set query permissions](#)

## Secure Azure Pipelines

Use [extends templates](#). For more information about how to set permission levels for pipelines, see [Set pipeline permissions](#).

### Policies

- Require at least one reviewer outside of the original requester. The approver takes co-ownership of the changes and should be held equally responsible for any impact it may have.
- Require CI build to pass, which is useful for establishing baseline code quality, such as code linting, unit tests, and even security checks like virus and credential scans.
- Ensure that the original pull requester can't approve the change.
- Disallow completion of a PR (Pull Request), even if some reviewers vote to wait or reject.
- Reset code reviewer votes when recent changes get pushed.
- Lock down release pipelines by running them only on specific production branches.
- Enable "Enforce settable at queue time for variables" in your organization's pipeline settings.
- Don't allow "Let users override this value when running this pipeline," for variables set in the editor.

### Agents

- Grant permissions to the smallest possible number of accounts.
- Have the most restrictive firewall that leaves your agents usable.
- Update pools regularly to ensure the build fleet isn't running vulnerable code that can be exploited by a malicious actor.
- Use a separate agent pool for build artifacts that get shipped or deployed to production.
- Segment "sensitive" pool from non-sensitive pools, and only allow the use of credentials in build definitions

that are locked to that pool.

## Definitions

- Manage pipeline definitions with YAML (Yet Another Markup Language). YAML is the preferred method for managing pipeline definitions, as it provides traceability for changes and can follow approval guidelines.
- Secure the pipeline definition *Edit* access to the minimum number of accounts.

## Input

- Include sanity checks for variables in build scripts. A sanity check can mitigate a command injection attack through the settable variables.
- Set as few build variables as possible to "Settable at release time."

## Tasks

- Avoid remotely fetched resources, but, if necessary, use versioning and hash checking.
- Don't log secrets.
- Don't store secrets in pipeline variables, use Azure KeyVault. Regularly scan your build pipelines to ensure secrets aren't being stored in build pipeline variables.
- Don't let users run builds against arbitrary branches or tags on security-critical pipelines.
- Disable inheritance on the pipeline, as inherited permissions are broad and don't accurately reflect your needs for permissions.
- Limit job authorization scopes in all cases.

## Repositories and branches

- Set the "Require a minimum number of reviewers," policy to *on*, so that every pull request gets reviewed by at least two approvers.
- Configure security policies specific to each repository or branch, instead of project wide. Security policies reduce risk, enforce change management standards, and improve your team's quality of code.
- Store production secrets in a separate KeyVault and ensure that access is only granted on a need-to-know basis to keep non-production builds separate.
- Don't mix test environments with production, including use of credentials.
- Disable forking. The more forks there are, the harder it is to keep track of each fork's security. Also, a user can easily fork a copy of a repository to their own private account.
- [Don't provide secrets to fork builds.](#)
- [Consider manually triggering fork builds.](#)
- [Use Microsoft-hosted agents for fork builds.](#)
- For Git, check your production build definitions in the project's git repository, so they can be scanned for credentials.
- Configure a branch control check so that only pipelines running in the context of the `production` branch may use the `prod-connection`.

For more information, see [Other security considerations](#).

For more information about granular permission controls that can be configured according to the project's needs, see [Security groups, service accounts, and permissions in Azure DevOps](#).

## Secure Azure Repos

[Improve code quality with branch policies](#). For more information about branch permissions and policies, see [Set branch permissions](#).

## Secure Azure Test Plans

Check out the following articles:

- [Set permissions and access for testing](#)
- [Supported scenarios and access requirements](#)

## Secure Azure DevOps - general

- Disable inheritance where possible. Due to the allow-by-default nature of inheritance, unexpected users can get access or permissions. For more information, read about [inheritance](#).
- Only give users and services the minimum amount of access to perform their business functions.
- Periodically [review audit events](#) to monitor and react to unexpected usage patterns by administrators and other users.
- Check out the following articles:
  - [Permissions and role lookup guide](#)
  - [Permissions, security groups, and service accounts reference](#)

## Related articles

- [Valid user groups](#)
- [Project-scoped user groups](#)
- [Manage conditional access](#)
- [Unit testing best practices with .NET Core and .NET Standard](#)

# Plan your organizational structure

12/13/2022 • 15 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Use your business structure as a guide for the number of organizations, projects, and teams that you create in Azure DevOps. This article helps you plan for different structures and scenarios for Azure DevOps.

Consider the following structures for your business and collaborative work in Azure DevOps:

- [Number of organizations](#)
- [Number of projects under an organization](#)

You also may want to plan for the following scenarios:

- [Map your organizations and projects](#) in Azure DevOps to your enterprise, business unit, and team structure
- [Structure your repositories \(repos\)](#)
- [Structure your teams](#)- it can either help or hinder teams to be Agile and autonomous
- [Manage access to data](#) - who needs to have access and who doesn't?
- [Reporting needs](#)
- Promote common practices - [use foundational elements to create an agile mindset and culture](#)

Have at least one organization, which may represent your company, your larger collection of code projects, or even multiple related business units.

## What's an organization?

An organization in Azure DevOps is a mechanism for organizing and connecting groups of related projects.

Examples include business divisions, regional divisions, or other enterprise structures. You can choose one organization for your entire company, one organization for yourself, or separate organizations for specific business units.

Each organization gets its own *free tier* of services (up to five users for each service type) as follows. You can use all the services, or choose only what you need to complement your existing workflows.

- [Azure Pipelines](#): One hosted job with 1,800 minutes per month for CI/CD and one self-hosted job
- [Azure Boards](#): Work item tracking and Kanban boards
- [Azure Repos](#): Unlimited private Git repos
- [Azure Artifacts](#): Package management
- Unlimited Stakeholders
  - 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 Kanban boards
  - [Azure Repos](#): Unlimited private Git repos
  - [Azure Artifacts](#): Two GiB free per organization

#### **NOTE**

While Azure DevOps cloud-based load testing service is deprecated, [Azure Load Testing Preview](#) is available. Azure Load Testing Preview is a fully managed load testing service that enables you to use existing Apache JMeter scripts to generate high-scale load. To learn more, see [What is Azure Load Testing Preview?](#). To learn more about the deprecation of Azure DevOps load testing and other, alternative services see [Changes to load test functionality in Visual Studio and cloud load testing in Azure DevOps](#).

## How many organizations do you need?

Start with one organization in Azure DevOps. Then, you can add more organizations—which may require different security models—later. A single code repo or project only needs one organization. If you have separate teams that need to work on code or other projects in isolation, consider creating separate organizations for those teams. They'll have different URLs. Add projects, teams, and repos, as necessary, before you add another organization.

Take some time to review your work structure and the different business groups and participants to be managed. For more information, see [Map your projects to business units](#) and [Structure considerations](#).

#### **TIP**

For company-owned Azure AD organizations, consider restricting users from creating new organizations as a way to protect your IP. For more information, see [Restrict organization creation via Azure AD tenant policy](#). Users can create organizations using their MSA or GitHub accounts with no restrictions.

## What's a team?

A team is a unit that supports many [team-configurable tools](#). These tools help you plan and manage work, and make collaboration easier.

### **Create a team for each distinct product or feature team**

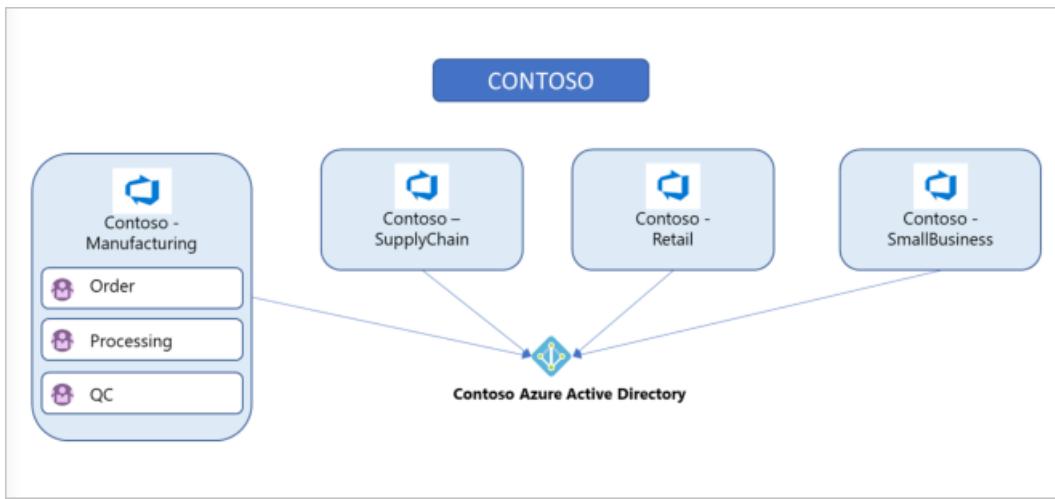
Each team owns their own backlog. To create a new backlog, you create a new team. [Configure teams and backlogs into a hierarchical structure](#), so program owners can more easily track progress across teams, manage portfolios, and generate rollup data. A team group gets created when you create a team. You can use this group in queries or to set permissions for your team.

## What's a project?

A project in Azure DevOps contains the following set of features:

- Boards and backlogs for agile planning
  - Pipelines for continuous integration and deployment
  - Repos for version control and management of source code and artifacts
  - Continuous test integration throughout the project life cycle
- Each organization contains one or more projects

In the following image, the fictitious Contoso company has four projects within their Contoso-Manufacturing organization.



## How many projects do you need?

Have at least one project to start using an Azure DevOps service, such as Azure Boards, Azure Repos, or Azure Pipelines. When you create your organization, a default project gets created for you. In your default project, there's a code repo to start working in, backlog to track work, and at least one pipeline to begin automating build and release.

Within an organization, you can do either of the following approaches:

- Create a single project that contains many repos and teams
- Create many projects, each with its own set of teams, repos, builds, work items, and other elements

Even if you have many teams working on hundreds of different applications and software projects, you can manage them within a single project in Azure DevOps. However, if you want to manage more granular security between your software projects and their teams, consider using many projects. At the highest level of isolation is an organization, where each organization is connected to a single Azure AD tenant. A single Azure AD tenant, however, can be connected to many Azure DevOps organizations.

### NOTE

If the **Limit user visibility and collaboration to specific projects** preview feature is enabled for the organization, users added to the **Project-SScoped Users** group won't be able to access projects that they haven't been added to. For more information, see [Manage your organization, Limit user visibility for projects and more](#).

### Single project

A single project puts all of the work at the same "portfolio" level for the entire organization. Your work has the same set of repos and iteration paths. With a single project, teams share source repos, build definitions, release definitions, reports, and package feeds. You might have a large product or service that's managed by many teams. Those teams have tight inter-dependencies across the product life cycle. You create a project and divide the work using teams and area paths. This setup gives your teams visibility into each other's work, so the organization stays aligned. Your teams use the same taxonomy for work item tracking, making it easier to communicate and stay consistent.

### TIP

When multiple teams work on the same product, having all teams on the same iteration schedule helps keep your teams aligned and delivering value on the same cadence. For example, our organization in Azure DevOps has over 40 feature teams and 500 users within a single project - this works well because we're all working on a common product set with common goals and a common release schedule.

A high volume of queries and boards can make it hard to find what you're looking for. Depending on the architecture of your product, this difficulty can bleed into other areas such as builds, releases, and repos. Make sure to use good naming conventions and a simple folder structure. When you add a repo to your project, consider your strategy and determine whether that repo could be placed into its own project.

## Many projects

You can best determine project structure by how you ship the product. Having several projects shifts the administration burden and gives your teams more autonomy to manage the project as the team decides. It also provides greater control of security and access to assets across the different projects. Having team independence with many projects creates some alignment challenges, however. If each project is using a different process or iteration schedule, it can make communication and collaboration difficult if the taxonomies aren't the same.

### TIP

If you use the same process and iteration schedules across all your projects, your ability to roll-up data and report across teams improves.

Azure DevOps provides cross-project experiences for managing work.

You may want to add another project due to the following scenarios:

- To prohibit or manage access to the information within a project
- To support custom work tracking processes for specific business units within your organization
- To support entirely separate business units that have their own administrative policies and administrators
- To support testing customization activities or adding extensions before rolling out changes to the working project

When you're considering many projects, keep in mind that Git repo portability makes it easy to migrate repos (including full history) between projects. Other history can't be migrated between projects. Examples are push and pull request history.

When you map projects to business units, your company gets a single organization and sets up many projects with one or more projects representing a business unit. All Azure DevOps assets of the company are contained within this organization and located within a given region (for example, Western Europe). Consider the following guidance for mapping your projects to business units:

	ONE PROJECT, MANY TEAMS	ONE ORGANIZATION, MANY PROJECTS, AND TEAMS	MANY ORGANIZATIONS
General guidance	Best for smaller organizations or larger organizations with highly aligned teams.	Good when different efforts require different processes.	Useful as part of TFS legacy migrations and for hard security boundaries between organizations. Used with multiple projects and teams within each organization.
Scale	Supports tens of thousands of users and hundreds of teams, but best at this scale if all teams are working on related efforts.	Same as with one project, but many projects may be easier.	

	ONE PROJECT, MANY TEAMS	ONE ORGANIZATION, MANY PROJECTS, AND TEAMS	MANY ORGANIZATIONS
<b>Process</b>	Aligned processes across teams; team flexibility to customize boards, dashboards, and so on.	Independent processes for each project. For example, different work item types, custom fields, and so on.	Same as many projects.
<b>Collaboration</b>	Highest default visibility and reuse between work and assets of different teams.	Good visibility and reuse are possible, but it's easier to hide assets between projects whether intentional.	Poor visibility, collaboration, and reuse between organizations.
<b>Roll-up reporting and portfolio management</b>	Best ability to roll up across teams and coordinate between teams.	Good reporting possible across projects. More difficult for cross-project roll-up and team coordination.	No roll-up or coordination between organizations.
<b>Security/isolation</b>	Can lock down assets at a team level, but default is open visibility and collaboration.	Better ability to lock down between projects. By default, provides good visibility within projects and good isolation across projects.	Hard boundaries across organizations; excellent isolation and minimal ability to share across organizations.
<b>Context switching</b>	Easiest for teams to work together and for users to switch between efforts.	Relatively easy for users to work together and switch contexts between efforts.	More difficult for users having to work across different organizations.
<b>Information overload</b>	By default, all assets are visible to users who make use of "favorites" and similar mechanisms to avoid "information overload."	Reduced risk of information overload; most project assets hidden across project boundaries.	Assets across organizations are isolated, reducing risk of information overload.
<b>Administrative overhead</b>	Much administration is delegated down to individual teams. Easiest for user licensing and org-level administration. More work may be needed if alignment is required between efforts.	More administration at the project level. More overhead, but can be useful when projects have different administrative needs.	As with more projects, there's more administrative overhead, which enables more flexibility between orgs.

## Structure repos and version control within a project

Consider the specific strategic work scoped to one of the organizations you created previously and who needs access. Use this information to name and [create a project](#). This project has a URL defined under the organization you created it in and can be accessed at <https://dev.azure.com/{organization-name}/{project-name}>.

Configure your project in [Project settings](#).

The screenshot shows the Azure DevOps Project Overview page for 'FabrikamFiber'. The left sidebar contains a list of project management services: Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The 'Project settings' link is highlighted with a red box. The main content area features a cartoon illustration of a person working at a desk with a dog. Below the illustration, the text 'Welcome to the project!' is displayed, followed by 'What service would you like to start with?' and a row of buttons for Boards, Repos, Pipelines, Test Plans, and Artifacts. A blue 'Invite' button is also visible.

For more information about managing projects, see [Manage projects in Azure DevOps](#). You can move a project to a different organization by migrating the data. For more information about migrating your project, see [Migration options](#).

## Manage version control

In projects where the Azure Repos service is enabled, version control repos can store and revise code. Consider the following options when you're configuring repos.

### Git vs. Team Foundation Version Control (TFVC)

Azure Repos offers the following version control systems for teams to choose from:

- Git and TFVC. Projects can have repos of each type. By default, new projects have an empty Git repo. Git enables a great amount of flexibility in developer workflows and integrates with nearly every relevant tool in the developer ecosystem. Any project can use Git repos. There's no limit on the amount of Git repos that can be added to a project.

TFVC is a centralized version control system that is also available. Unlike Git, only one TFVC repository is allowed for a project. But, within that repo, folders, and branches are used to organize code for multiple products and services, if wanted. Projects can use both TFVC and Git, if appropriate.

### One vs. many repos

Do you need to set up multiple repos within a single project or have a repo set up per project? The following guidance relates to the planning and administration functions across those repos.

One project containing multiple repos works well if the products/services are working on a coordinated release schedule. If developers are frequently working with multiple repos, keep them in a single project to ensure the processes remain shared and consistent. It's easier to manage repo access within a single project, as access controls and options like case enforcement and max file size get set at the project level. You can manage the access controls and settings individually, even if your repos are in a single project.

If the products stored in multiple repos work on independent schedules or processes, you can split them into multiple projects. Git repo portability makes it easy to move a repo between projects and still keep full-fidelity commit history. Other history, such as pull requests or build history, aren't easily migrated.

Base your decision for one vs. many repos on the following factors and tips:

- code dependencies and architecture
- put each independently deployable product or service in its own repo
- don't separate a codebase into many repos if you expect to make coordinated code changes across those repos, as no tools can help coordinate those changes
- if your codebase is already a monolith, keep it in one repo. For more information about monolithic repos, see [How Microsoft develops modern software with DevOps](#) articles
- if you have many disconnected services, one repo per service is a good strategy

#### TIP

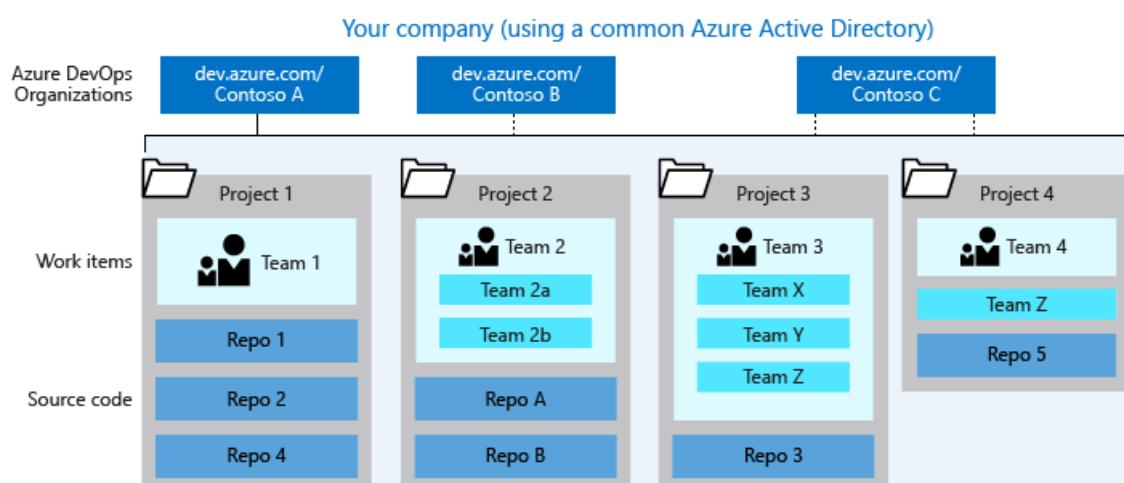
Consider [managing your permissions](#), so not everyone in your organization can [create a repo](#). If you have too many repos, it's hard to keep track of who owns which code or other content stored in those repos.

## Shared repo vs. forked repos

We recommend using a shared repo within a trusted organization. Developers use branches to maintain isolation of their changes from one another. With a good branching and release strategy, a single repo can scale to support concurrent development for more than a thousand developers. For more information about branching and release strategy, see [Adopt a Git branching strategy and Release Flow: Our Branching Strategy](#).

Forks can be useful when you're working with vendor teams that shouldn't have direct access to update the main repository. Forks can also be useful in scenarios where many developers contribute infrequently, such as in an open-source project. When you're working with forks, you may want to maintain a separate project to isolate the forked repos from the main repo. There may be added administrative overhead, but it keeps the main project cleaner. For more information, see the [Forks article](#).

The following image displays a sample of how "your company" could structure its organizations, projects, work items, teams, and repos.



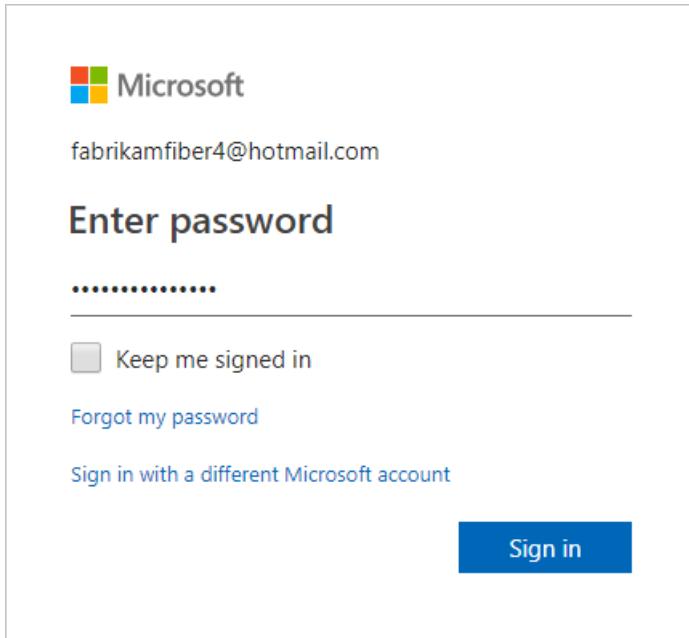
# More about organizational structure

## Choose your organization administrator account type

When you create an organization, the credentials that you sign in with define which identity provider your organization uses. Create your organization with a Microsoft account or Azure AD instance. Use those credentials to sign in as an administrator to your new organization at <https://dev.azure.com/{YourOrganization}>.

### Use your Microsoft account

Use your Microsoft account if you don't need to authenticate users for an organization with Azure AD. All users must sign in to your organization with a Microsoft account. If you don't have one, [create a Microsoft account](#).



If you don't have an Azure AD instance, create one for free from the [Azure portal](#) or use your Microsoft account to create an organization. Then, you can [connect your organization to Azure AD](#).

### Use your Azure AD account

You might have an Azure AD account already if you use Azure or Microsoft 365. If you work for a company that uses Azure AD to manage user permissions, you probably have an Azure AD account.

If you don't have an Azure AD account, [sign up for Azure AD](#) to automatically connect your organization to your Azure AD. All users must be members in that directory to access your organization. To add users from other organizations, use [Azure AD B2B collaboration](#).

Azure DevOps authenticates users through your Azure AD, so that only users who are members in that directory have access to your organization. When you remove users from that directory, they can no longer access your organization. Only specific [Azure AD administrators](#) manage users in your directory, so administrators control who accesses your organization.

For more information on managing users, see [Manage users](#).

### Map organizations to business units

Each business unit within your company gets its own organization in Azure DevOps, along with its own Azure AD tenant. You can [set up projects](#) within those individual organizations, as required, based on teams or ongoing work.

For a larger company, you can create multiple organizations using different user accounts (most likely Azure AD accounts). Consider what groups and users share strategies and work, and group them into specific organizations.

For example, the fictional Fabrikam company created the following three organizations:

- Fabrikam-Marketing
- Fabrikam-Engineering
- Fabrikam-Sales

Each organization has a separate URL, such as:

- <https://dev.azure.com/Fabrikam-Marketing>
- <https://dev.azure.com/Fabrikam-Engineering>
- <https://dev.azure.com/Fabrikam-Sales>

The organizations are for the same company, but are mostly isolated from each other. You don't need to separate anything this way. Only create boundaries when it makes sense to your business.

**TIP**

You can more easily partition an existing organization with projects, than combine different organizations.

## Related articles

- [Create an organization](#)
- [Create a project](#)
- [Connect your organization to Azure AD](#)
- [Set up billing](#)
- [Set user preferences](#)

# What is source control?

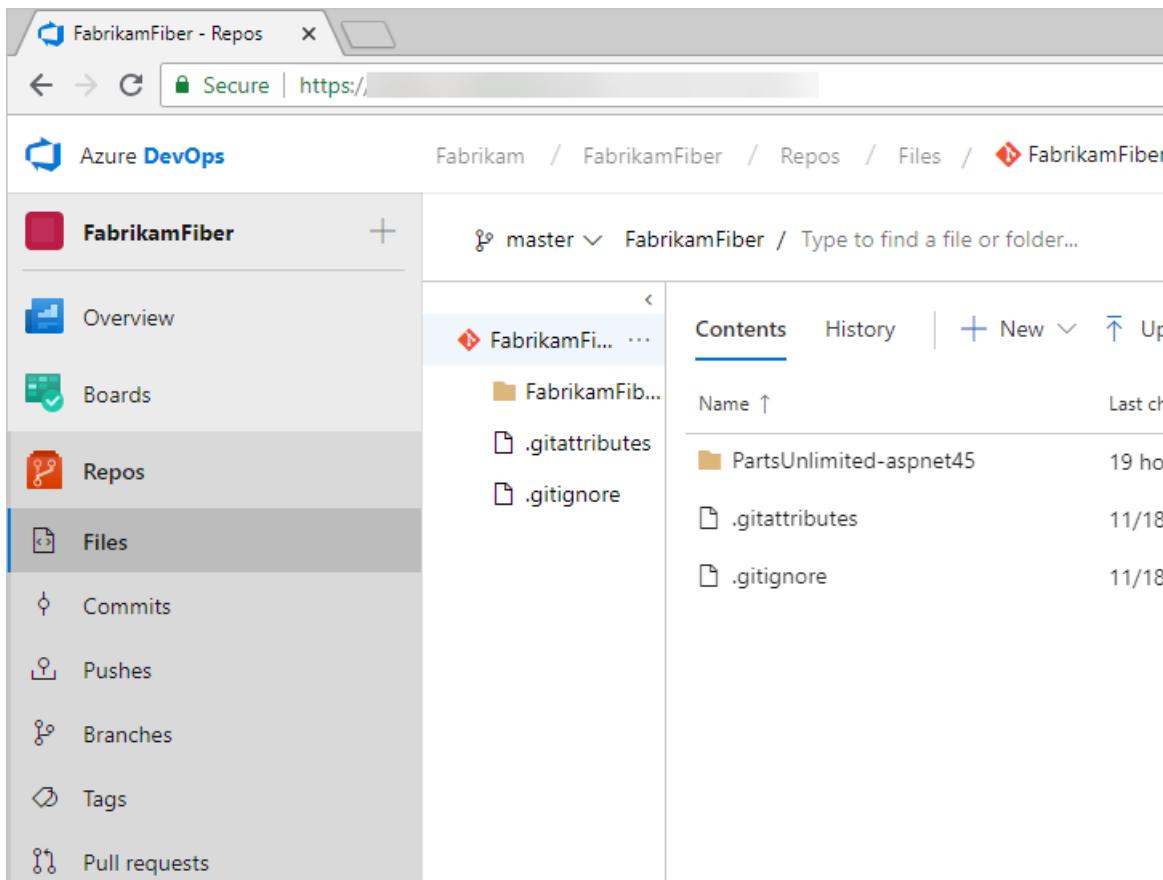
12/13/2022 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

A source control system, also called a *version control* system, allows developers to collaborate on code and track changes. Source control is an essential tool for multi-developer projects.

Our systems support two types of source control: Git (distributed) and Team Foundation Version Control (TFVC). TFVC is a centralized, client-server system. In both Git and TFVC, you can check in files and organize files in folders, branches, and repositories.

Manage your repos, branches, and other code development operations from [Azure Repos](#).



The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' repository. The left sidebar has 'FabrikamFiber' selected under 'Repos'. The main area shows the 'master' branch with three files: 'FabrikamFib...', '.gitattributes', and '.gitignore'. The '.gitignore' file was last modified 11/18.

Name	Last modified
FabrikamFib...	19 hours ago
.gitattributes	11/18
.gitignore	11/18

With Git, each developer has a copy of the source repository on their dev machine. The source repo includes all branch and history information. Each developer works directly with their local repository. Changes are shared between repositories as a separate step.

Developers can commit each set of changes and perform version control operations, such as history and compare without a network connection. Branches are lightweight. When developers need to switch contexts, they create a private local branch. Developers can quickly switch from one branch to another to pivot among different variations of the code base. Later, developers can merge, publish, or dispose of the branch.

## NOTE

Git in Visual Studio and Azure DevOps is standard Git. You can use Visual Studio with third-party Git services. You can also use third-party Git clients with Azure DevOps Server.

With TFVC, developers have only one version of each file on their dev machines. Historical data is maintained only on the server. Branches are path-based and are created on the server.

## Next steps

Start sharing your code or get your code by using source control.

[Code with Git](#)

## Related articles

- [Azure Repos documentation](#)
- [Git repositories documentation](#)

# Tools and clients that connect to Azure DevOps

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Our platform of software development tools began more than 20 years ago. We released Visual Basic and Visual Studio as an integrated development environment (IDE). Visual Studio supports many plug-ins that extend its functionality. In particular, the Team Explorer plug-in allows the Visual Studio client to connect to Azure DevOps to support source control, work tracking, build, and test operations.

## Desktop client developer tools

Developers have access to many tools through these versions of Visual Studio and plug-ins. To download any version of Visual Studio, go to the [Visual Studio Downloads page](#). To understand what features you get with the Visual Studio versions, see [Compare Visual Studio offerings](#).

- **Visual Studio Community:** A fully featured and extensible IDE for creating modern applications for Android, iOS, and Windows, including web applications and cloud services. (Replaces Visual Studio Express.)
- **Visual Studio Professional:** Development tools and services to support individual developers or small teams.
- **Visual Studio Enterprise:** Integrated, end-to-end development tools and solutions for teams of any size, and with a need to scale. It supports designing, building, and managing complex enterprise applications.
- **Visual Studio Test Professional:** Provides access to Microsoft Test and development tools to support quality and collaboration throughout the development process.
- **Visual Studio Team Explorer:** Free solution for non-developers to interact with Azure DevOps.
- **Eclipse/Team Explorer Everywhere:** Free plug in to support teams running Eclipse on Linux, macOS, or Windows that connects to Azure DevOps.
- **Android Studio with the Azure DevOps Services Plug-in for Android Studio:** Free plug in to support Android developers and connect to Git repositories on Azure DevOps.
- **IntelliJ with the Azure DevOps Services Plugin for IntelliJ:** Free plug in to support developers who use IntelliJ IDEA or Android Studio to connect to Git repositories on Azure DevOps.
- **Visual Studio Code:** Free, open-source code editor with a free extension to support connecting to Git repositories on Azure DevOps.

To get started with client libraries, see [Client library samples](#).

### Team Explorer plug-in

Team Explorer, a plug-in to all Visual Studio versions, connects Visual Studio to projects defined in Azure DevOps. You can manage source code, work items, and builds. To learn more, see [Work in Team Explorer](#).

## HOME PAGE WITH GIT

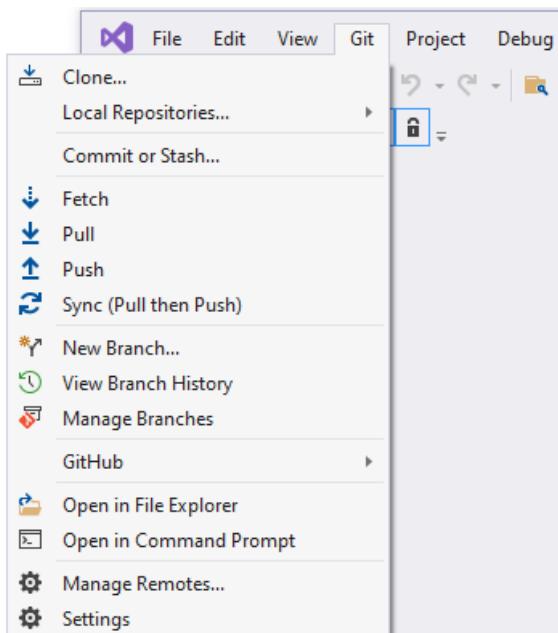
The screenshot shows the Team Explorer - Home page for a Git repository named "Fabrikam Fiber". The top navigation bar includes links for Home, Search Work Items, and a refresh icon. Below the navigation, the "Team Foundation Server" section displays the URL <http://vs-2015-test:8080/tfs/defaultco...>. The main content area is titled "Project" and contains links for Web Portal, Task Board, and Team Room. A sidebar on the left lists various Git-related items: Changes, Branches, Pull Requests, Sync, Work Items, Builds, Team Members, and Settings. A "Solutions" section at the bottom has links for New... and Open..., with a note stating "There were no solutions found."

## HOME PAGE WITH TFVC

The screenshot shows the Team Explorer - Home page for a TFVC repository named "Fabrikam Fiber". The top navigation bar includes links for Home, Search Work Items, and a refresh icon. Below the navigation, the "Team Foundation Server" section displays the URL <http://vs-2015-test:8080/tfs/defaultco...>. The main content area is titled "Project" and contains links for Web Portal, Task Board, and Team Room. A sidebar on the left lists various TFVC-related items: My Work, Pending Changes, Source Control Explorer, Work Items, Builds, Team Members, and Settings. A "Solutions" section at the bottom has links for New... and Open..., with a note stating "There were no solutions found."

## Visual Studio Git experience

Visual Studio 2019 and later versions provide a new Git experience through the **Git** menu as shown below. To learn more, see [Git experience in Visual Studio](#) and [Side-by-side comparison of Git and Team Explorer](#).



## Office integration tools

You can integrate the following Microsoft Office tools with Azure DevOps.

- [Excel](#): Use Excel to add and bulk modify work items.

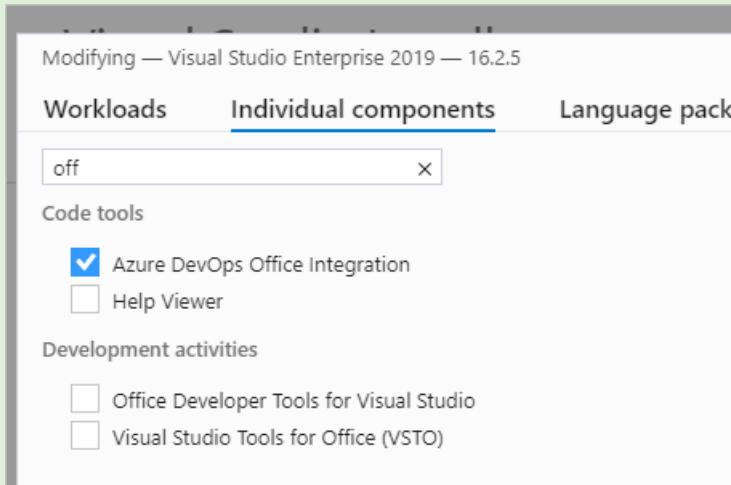
## IMPORTANT

Starting with Visual Studio 2019, the Team Foundation plug-in for Office is deprecating support for Microsoft Project. Project integration and the TFSFieldMapping command is not supported for Azure DevOps Server 2019 nor for Azure DevOps Services. However, you can continue to use Microsoft Excel.

- [Excel](#): Use Excel to add and bulk modify work items.

## TIP

Check to make sure the Azure DevOps Office Integration component is selected in the Visual Studio Installer, per the



following example.

When you install any edition of Visual Studio or [Team Foundation Server Standalone Office Integration 2015 \(free\)](#), the Team Foundation plug-in integrates work item tracking with select Office clients. The Team Foundation plug-in installs to your existing Office client. The plug-in supports Office 2007, Office 2010, or Office 2013 versions.

- [Excel](#): Use Excel to add and bulk modify work items.
- [Project](#): By using Project, you can plan projects, schedule tasks, assign resources, and track changes. You have access to features that TFS doesn't support, such as a project calendar, Gantt charts, and resource views.
- [PowerPoint Storyboarding](#): Illustrate user stories and requirements by using PowerPoint. The Team Foundation plug-in installs to your existing PowerPoint client.

## Task-specific clients

The following clients support specific tasks, such as managing testing efforts, providing feedback, or modifying work items:

- [Azure Test Plans](#): Manage your test efforts, create and run manual tests, and create and track bugs that are found during test efforts.
- [Test & Feedback extension \(previously called the Exploratory Testing extension\)](#): This extension provides a lightweight plug-in to a web browser. Stakeholders can respond to feedback requests for user stories and features created in Azure DevOps. This extension is free to Stakeholders.
- [Microsoft Feedback Client](#): Your Stakeholders can use this client to record feedback for your application as video, audio, or type-written comments. This client is installed with all versions of Visual Studio, or it can be [installed from the free download](#). All feedback is stored in the work item data store and requires Stakeholders to have permissions.

## Browser-based web tools

### Web portal

The collaboration tools supported through the web portal are summarized under [Essential services](#). New features are deployed every three weeks for Azure DevOps Services, and quarterly for Azure DevOps Server. For release notes, see [Azure DevOps Services Features Timeline](#).

You can use the following browsers to access the web portal:

VERSION	EDGE	INTERNET EXPLORER	SAFARI (MAC)	FIREFOX	CHROME
Azure DevOps Services Azure DevOps Server 2020.1	Most recent	Not supported	14.1 and later	Most recent	Most recent
Azure DevOps Server 2020 Azure DevOps Server 2019 TFS 2018 TFS 2017	Most recent	11 and later	14.1 and later	Most recent	Most recent
TFS 2015	Most recent	9 and later	5 and later	Most recent	Most recent
TFS 2013		9 and later	5 and later	Most recent	Most recent

Microsoft Edge, Firefox, and Chrome automatically update themselves, so Azure DevOps supports the most recent version.

For more information, see [Web portal navigation](#).

### Browser-based extensions

Several extensions are built and maintained by the Azure DevOps Services product team:

- [Code search](#): Increase cross-team collaboration and code sharing. Enables developers to quickly locate relevant information within the code base of all projects that are hosted within an organization or collection. You can discover implementation examples, browsing definitions, and error text.
- [Work item search](#): To quickly find relevant work items, search across all work item fields over all projects in an organization. Do full-text searches across all fields to efficiently locate relevant work items. Use inline search filters, on any work item field, to quickly narrow down a list of work items.

Find more extensions in Azure DevOps [Organization settings > Extensions > Browse marketplace](#). See also, [Overview of extensions for Azure Boards](#).

## Command-line tools

You can do many code development and administrative tasks by using the following command-line tools:

- [az devops commands](#)
- [Git commands](#)
- [TFVC commands](#)
- [TCM commands](#)
- [Manage permissions with command line tool \(az devops security\)](#)
- [witadmin \(work item tracking\)](#)
- [Git commands](#)
- [TFVC commands](#)

- [TCM commands](#)
- [witadmin \(work item tracking\)](#)
- [TFSCfg](#)
- [TFSDeleteProject](#)
- [TFSSecurity](#)
- [TFSServiceControl](#)

## Integrated tool support for third-party applications

The following tools provide support for monitoring and interacting with Azure DevOps from a third-party application.

- **Azure Boards:**
  - [Use the Azure Boards app with Slack to manage work items](#)
  - [Use the Azure Boards app in Microsoft Teams](#)
- **Azure Repos:**
  - [Azure Repos with Slack](#)
  - [Azure Repos with Microsoft Teams](#)
- **Azure Pipelines:**
  - [Use Azure Pipelines with Microsoft Teams](#)
  - [Azure Pipelines with Slack](#)
  - [Integrate with ServiceNow change management](#)
  - [Continuously deploy from a Jenkins build](#)

## Marketplace extensions

Visual Studio and Azure DevOps provide a wealth of features and functionality. They also provide a means to extend and share that functionality.

Extensions are simple add-ons that you can use to customize and extend your DevOps and work tracking experiences. They're written with standard technologies—HTML, JavaScript, and CSS. You can develop your own extensions by using your preferred dev tools.

You build extensions by using our RESTful API library. Publish your extensions to the Azure DevOps Marketplace. You can privately maintain or share them with millions of developers who use Visual Studio and Azure DevOps.

To learn more, visit the [Azure DevOps Marketplace](#) and see [Overview of extensions](#).

## REST APIs

The Azure DevOps APIs are based on REST, OAuth, JSON, and service hooks—all standard web technologies broadly supported in the industry.

REST APIs are provided to support building extensions to Azure DevOps. To learn more, see [REST API overview](#).

## Related articles

- [A tour of services](#)
- [Software development roles](#)
- [Pricing](#)
- [Azure DevOps data protection overview](#)

# Software development roles supported by Azure DevOps

12/13/2022 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

If you're a sole developer or work in a small setting, you track issues, plan features, code, test, build, and deploy.

If you work in a large setting, you might be more focused on a specific set of tasks that aligns with specific roles. These specific roles could be software development, product and scrum management, or DevOps.

The following article describes the features and tasks available to you, based on your role.

## Contributor roles

Team members are contributors who have access to the following areas and more:

- code base
- work item tracking
- Agile tools
- build pipelines
- test tools

If you need to lock down specific areas to a select set of contributors, see [permission management](#).

### Software developers

Developers use Visual Studio or other [tools](#) to develop their applications. They then check in their changes to a Git or Team Foundation Version Control (TFVC) repository hosted in Azure DevOps. From the web portal or a supported IDE, they can view repositories, check history, and more.

To get started with using Git, see one of the following resources:

- [Share your code with Git and Visual Studio](#)
- [Share your code in Git by using Eclipse](#)
- [Share your code in Git by using Xcode](#)
- [Share your code in Git by using IntelliJ](#)
- [Get started with using Git and Azure DevOps Services](#)

To get started with using TFVC, see one of the following resources:

- [Develop and share your code in TFVC by using Visual Studio](#)
- [Share your code in TFVC by using Eclipse](#)
- [Share your code in TFVC by using Xcode](#)

### Product owners

Product owners typically plan the feature set to deliver, set priorities, and track the status of work, code defects, and customer issues. The suite of web-based Agile tools in Azure DevOps provides product owners with the views and features that they need to do these tasks. All work gets captured within a work item. Each work item represents a specific type such as a user story, task, or bug.

- Use the product backlog to quickly define and prioritize user stories, features, and other work items

- Use the sprint backlog and task board to implement Scrum practices
- Use the Kanban board to work with Kanban methods
- Use queries to list and update work items, create status and trend charts, and post charts to dashboards
- Use dashboards to share information, status, and trends with your team or organization

For more information about getting started, see [About Azure Boards and Agile tools](#).

You can integrate Microsoft Excel with Azure DevOps to plan and track your work. For more information, see [Bulk modify by using Excel](#).

## Scrum masters

Scrum masters help to facilitate scrum to the larger team by ensuring the scrum framework gets followed. They're committed to the practices, but stay flexible and open to opportunities for the team to improve their workflow. Scrum masters utilize the same features as [product owners](#).

## DevOps: builders, testers, and release managers

An advantage of working with Azure DevOps is the suite of tools and integrated functionality that support build, testing, and deploying software applications. See the following general DevOps-associated tasks that Azure DevOps supports.

- Define builds
- Unit test your code
- Run tests with your builds
- Perform exploratory tests
- Define, manage, track, and approve releases
- Deploy applications to Azure, a virtual machine, Docker containers, and more

To get started, see the overviews in [Azure Pipelines](#) and [Azure Test Plans](#).

## Stakeholders

With Stakeholder access, anyone in your organization can check project status and provide feedback. Stakeholders can track project priorities and provide direction, feature ideas, and business alignment to a team. Stakeholders also contribute to plans by adding and modifying work items. They can't, however, contribute to the code base or exercise test tools.

Stakeholder access essentially provides free access to a limited set of feature to project sponsors and supporters. To learn more, see [Work as a Stakeholder](#).

# Administrator roles

A distinct advantage to working in Azure DevOps Services is the reduced overhead of server maintenance. But there are several administrative tasks required to support a collaborative, integrated software development environment.

The main tasks are grouped as follows by membership in a security group or role.

## Team administrators

Responsible for configuring team settings, which include:

- Backlog and board settings
- Team areas and iterations (sprints)
- Team members
- Team dashboards
- Team work item templates

- Team alerts

To get started, see [Manage teams and configure team tools](#).

### **Project administrators**

Responsible for configuring project-level resources, including:

- [Area paths](#) and [iteration paths](#)
  - [Project permissions](#) and [repository security](#)
  - [Build agents, pools, and service connections](#)
  - [Test](#) and [release](#) retention policies
- 
- [Area paths](#) and [iteration paths](#)
  - [Project permissions](#) and [repository security](#)
  - [Customizing work tracking objects](#)
  - [Build agents, pools, and service connections](#)
  - [Test](#) and [release](#) retention policies

### **Organization owners and Project Collection Administrators**

Organization owners are automatically members of the Project Collection Administrators group. Responsible for configuring organization-level resources, including the following tasks:

- Manage billing
- Add and manage projects
- Manage collection-level permissions
- Customize work tracking processes
- Install and manage extensions

To get started, see [Manage organizations](#) and [Settings](#).

### **Project Collection Administrators**

Responsible for configuring collection-level resources. These tasks include:

- Add and manage projects
- Manage collection-level permissions
- Install and manage extensions

To get started, see [Settings](#).

### **Azure DevOps Server administrators**

Responsible for installing, upgrading, and maintaining an on-premises Azure DevOps Server deployment, including the:

- Install Azure DevOps Server
- Update servers running Azure DevOps Server
- Manage database backups
- Manage server administrative settings and permissions
- Build retention policies
- Add and manage project collections

To get started, see [Server Administration \(Azure DevOps Server\)](#).

## Related articles

- A tour of services
- Plan your organizational structure in Azure DevOps

# Troubleshoot connecting to a project

12/13/2022 • 5 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

## Troubleshoot connectivity

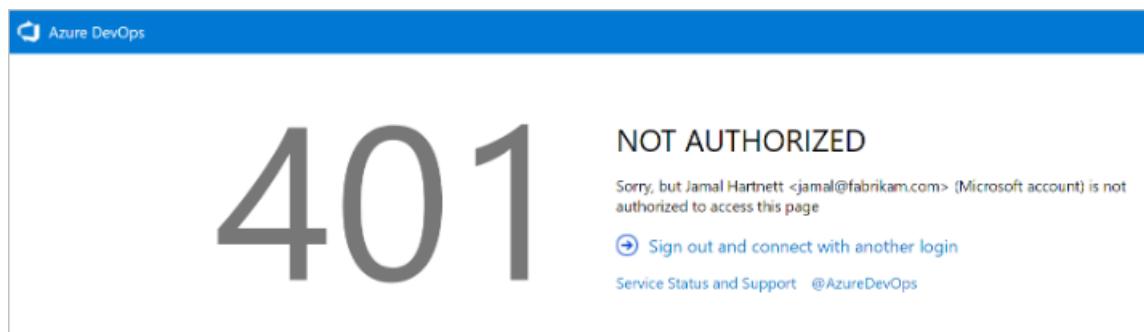
As a first step in resolving connectivity issues with Azure DevOps, complete the following steps:

1. Sign out of your browser. To do so, select the [Visual Studio sign out](#) link.
2. Delete the cookies in your browser. To delete cookies in most browsers, select Ctrl+Shift+Del.
3. Open Internet Explorer and delete the browser cookies. The Visual Studio IDE uses Internet Explorer cookies.
4. Close all browsers and close the Visual Studio IDE.
5. Use a private browser session to retry the connection. If the issue is with the Visual Studio IDE, remove the connection, and then readd it.

## Troubleshoot signing in

Two types of identities can sign in: Microsoft accounts and Azure Active Directory (Azure AD) accounts. Depending on your account, you might experience one of the following errors.

401 - Not Authorized



The most common error page is the *401 Not Authorized* error, which occurs when your identity doesn't have permissions to enter an organization. See the following common reasons for the error:

- Your identity isn't a member of the organization.
- Your identity has an invalid or missing license assignment.
- Your identity doesn't have enough memberships to access the resource. For example, membership to the Reader/Contributors group.
- Your identity is a B2B guest in the tenant, and the invitation hasn't been accepted.

If you think you're a member of the organization, but are blocked by this error page, [contact Support](#).

### Scenario 1

Your work or school Azure AD account doesn't have access, but your personal Microsoft account does.

401 - Work or school, or Personal account

# 401

## NOT AUTHORIZED

[jamal@fabrikam.com](#) has multiple accounts associated with it.

Your work or school account does not have access to [dev.azure.com/Fabrikam](#), but your personal account does have access.

[Sign in with your personal account](#)

[Sign out and connect with another login](#)

[Service Status and Support](#) | [@AzureDevOps](#)

A highly specific 401 error case. In this case, both a personal Microsoft account and a work or school account (Azure AD) that have the same sign-in address exist. You've signed in with your work or school account, but your personal account is the identity with access to the organization.

### Mitigation

In some cases, you might not know you have two identities with the same sign-in address. The work or school Azure AD account might have been created by an administrator when you were added to Office365 or Azure AD.

To sign out of your current work or school Azure AD account, select **Sign in with your personal MSA account**, and then sign in by using your personal Microsoft account. After authentication, you should have access to the organization.

- If you can't access to the organization, make sure that your Azure Active Directory still exists and that your work or school account is in the Azure AD tenant.

#### TIP

To avoid seeing this prompt, you can rename your Microsoft account. Then, only one identity, your work or school account, or Azure AD account, uses your sign-in address.

### Scenario 2

Your personal Microsoft account doesn't have access, but your Azure AD account does. This scenario is an opposite version of the 401 error page. In this case, the personal account (Microsoft account identity) doesn't have access to the organization and the work or school account (Azure AD identity) does. The same guidance from Scenario 1 applies, but in reverse.

401 - Work or school, or Personal account

# 401

## NOT AUTHORIZED

[jamal@fabrikam.com](#) has multiple accounts associated with it.

Your personal account does not have access to [dev.azure.com/Fabrikam](#), but your work or school account does have access.

[Sign in with your work or school account](#)

[Sign out and connect with another login](#)

[Service Status and Support](#) | [@AzureDevOps](#)

### Mitigation

When you get redirected back to the original sign-in page, we recommend that you clear all cookies, and then reattempt to sign in. If that doesn't fix the issue, [contact Support](#).

# Troubleshoot Azure DevOps Server connectivity

Here's a list of the most frequently reported connection problems and what to do about them. Complete the list in the order indicated.

1. Verify that you have the required permissions.

If the errors that you receive indicate read-only or blocked actions, you might not have permissions to act on the data.

2. Verify that your computer is connected to the network and that it can access network resources.

3. Verify that Azure DevOps Server hasn't been taken offline. Talk with your Azure DevOps Server administrator.

4. Check whether your project has been moved to another project collection in Azure DevOps Server. If it has been moved, you must create a connection to the new server name.

For additional troubleshooting tips, see [TF31002: Unable to connect to this Azure DevOps Server](#).

## Switch organizations

When you use two or more organizations that are linked to Azure AD, the sign-out function might not work as expected. For example, you can't switch between different organizations to connect to multiple organizations that are linked to directory tenants.

When this problem occurs, a blank screen flashes several times. Then, one of the following error messages appears after you connect to or add a new connection in the **Connect to Azure DevOps Server** dialog box:

TF31003: Either you have not entered the necessary credentials, or your user account does not have permission to connect to the Azure DevOps Server

TF31002: Unable to connect to this Azure DevOps Server

To resolve this issue, apply Visual Studio 2013.2 or install a later version from the [Visual Studio download website](#).

Another solution is to delete your browser cookies. For more information, see the support article [You can't switch between different organizations in Visual Studio Codespaces](#).

## Connect to Azure DevOps Server with Secure Sockets Layer

If you connect to an Azure DevOps Server instance that has Secure Sockets Layer (SSL) configured, install a certificate and clear the client cache. For details, see [Set up HTTPS with Secure Sockets Layer \(SSL\) for Azure DevOps Server - Configuring client computers](#).

## Clear the cache on client computers

When the on-premises Azure DevOps Server configuration changes, such as when you move or split a project collection, clear the cache.

1. Sign in to your client computer for Azure DevOps Server by using the credentials of the user whose cache you want to clear.
2. Close any open instances of Visual Studio.
3. Open a browser and go to one of the following folders, depending on the operating system your computer runs on:

- **Windows 10** *Drive:\Users< i >UserName\AppData\Local\Microsoft\Team Foundation\6.0\Cache*
  - **Windows 8** *Drive:\Users< i >UserName\AppData\Local\Microsoft\Team Foundation\4.0\Cache*
  - **Windows 7 or Windows Vista** *Drive:\Users< i >UserName\AppData\Local\Microsoft\Team Foundation\2.0\Cache*
4. Delete the contents of the Cache directory, including all subfolders.

# TF31002: Unable to connect

12/13/2022 • 5 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You might receive this error when you try to connect to Azure DevOps Services or an on-premises Azure DevOps Server from Visual Studio.

## You receive this error when you try to connect to Azure DevOps Services

PROBLEM	RESOLUTION
You don't have an active account or license.	Check with your administrator that you're a member of the account and have an active, valid license. See <a href="#">Assign licenses to users</a> for details.
Your Azure DevOps Services organization is connected to the Azure Active Directory.	When your Azure DevOps Services organization is connected to a directory that is associated with a Microsoft 365 or Microsoft Azure subscription, only members in the directory can access the account.  Check with your directory administrator to have them <a href="#">create an organizational account for you or add your account to the directory as external member</a> .
You can't switch between different organizational accounts.	If you work with several organizations that connect to different directories, such as accounts created from the Microsoft Azure Portal, the sign-out function might not work as expected. For example, you can't switch between different organizational accounts to connect to multiple accounts that are linked to directory tenants.  When this problem occurs, you see a flashing blank sign in dialog box several times. Then, you receive either TF31002 or TF31003 error after you connect to or add a new connection in "Connect to Team Foundation Server" dialog box.  To resolve this problem, apply the most recent <a href="#">Visual Studio update</a> .  To learn more, see <a href="#">You can't switch between different organizational accounts in Visual Studio Online</a> .
You want to sign in to Azure DevOps Services from Visual Studio using different credentials.	See <a href="#">Connect to projects, Sign in with different credentials</a> .

## When you try to connect to an on-premises Azure DevOps Server from your client computer

If you determine that you're receiving this error from one computer but not others, or others aren't receiving this error, then check the problem resolutions that are outlined below.

PROBLEM	RESOLUTION
Your password has expired.	Verify that you entered your user ID and password correctly, and that your password hasn't expired.
You've entered an incorrect server URL.	Verify that you've entered the server URL correctly including the server name, port number, and protocol (http/https). See <a href="#">Connect to projects</a> to learn more.
The TFS configuration has changed.	If the configuration for the on-premises Azure DevOps Server has changed, you must create a new connection. You might also need to <a href="#">clear the client cache</a> .
You work remotely and need to connect to a TFS Proxy server to check in files to Team Foundation version control.	<a href="#">Configure Visual Studio to connect to TFS Proxy</a> .
You're connecting to a later version of TFS than your Visual Studio client version.	Your version of Visual Studio or Team Explorer might be incompatible with Team Foundation Server. You might need to install one or more GDR packs. See <a href="#">Requirements and compatibility</a> for details.
Your firewall is blocking TFS services.	See <a href="#">Allow a program to communicate through Windows Firewall</a> .
Visual Studio stops responding when you run a query in Visual Studio.	Your computer might be configured to bypass the proxy server. Verify the configuration of the BypassProxyOnLocal setting on your computer. For more information, see <a href="#">BypassProxyOnLocal Configuration</a> .

## Several users can't connect to an on-premises Azure DevOps Server

If the problem occurs on more than one computer, contact your administrator to confirm whether the server is operational and available on the network.

As an administrator, check the event logs for the application-tier server to try to pinpoint the problem. Also, you can use the following table to determine whether the server is misconfigured. In the table, problems that are more likely to occur appear first. Try the resolutions in the order in which they appear, which increases the chance that you can solve the problem quickly.

PROBLEM	RESOLUTION
The <i>TFSService</i> account password has expired or is incorrect.	Many services for Team Foundation Server will stop running when the service account for Team Foundation has expired. For more information, see <a href="#">Change the service account or password for Team Foundation Server</a> .
The application-tier server for Team Foundation is unavailable.	Verify whether each required service is running. If a required service isn't running, you must restart it. If necessary, set it to start automatically. For more information, see <a href="#">Stop and start services, application pools, and websites</a> .
The network is unavailable.	Verify whether your network is operational.
A website identity for Team Foundation is configured incorrectly.	Verify or correct the server binding assignments that are made to websites for Team Foundation.

PROBLEM	RESOLUTION
Access to a website for Team Foundation has been restricted.	Verify or correct restrictions that are made to those websites that are based on IP addresses and domain names.
The firewall or ports are configured incorrectly.	Verify or correct port binding assignments for websites and port assignments for the firewall. First, you should open the administration console for Team Foundation, display the Application Tier page, and review the URL assignments. If necessary, you can click <b>Change URL</b> to modify the URL of a website. Next, you should verify the port assignments for Internet Information Services (IIS) and the ports that are allowed through the firewall. For more information, see <a href="#">Review Server Status and Settings</a> and <a href="#">Verify or Correct Port Assignments</a> .
Trust relationships between domains aren't configured correctly.	If a group of users can't access Team Foundation Server, you might have trust issues between domains.
When users connect to different versions of TFS from Visual Studio, for example, they connect to TFS 2012 and then TFS 2008, they can get the TF31002 error.	<p>This error can occur because the GUIDs for the TFS 2012 collection are the same as TFS 2008. The local client cache gets confused because it tries to maintain the same GUID-based local cache for both the 2008 server and the new Project Collection in 2012.</p> <p>To fix, run the <b>TFSCconfig ChangeServerID</b> command. See <a href="#">TFSCconfig ChangeServerID command</a>.</p>

# Troubleshoot access and permission issues

12/13/2022 • 10 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Azure DevOps security and permission structure is extensive, so you may find yourself needing to investigate why you or a project member doesn't have the access to a project, service, or feature that they expect to have. Find step-by-step guidance to understand and address problems a project member may be having in connecting to a project or accessing an Azure DevOps service or feature.

Before using this guide, we recommend that you're familiar with the following content:

- [Get started with permissions, access, and security groups](#)
- [Default permissions and access quick reference](#).
- [Quick reference index to Azure DevOps security](#)

## TIP

When you're creating an Azure DevOps security group, label it in a way that is easy to discern if it's created to limit access.

Permissions get set at one of the following levels:

- object level
- project level
- organization or project collection level
- security role
- team administrator role

## Common access and permission issues

See the following most common reasons a project member can't access a project, service, or feature:

ISSUE	TROUBLESHOOTING ACTION
Their access level doesn't support access to the service or feature.	To discover if this is the cause, you want to <a href="#">determine the user's access level and subscription status</a> .
Their membership within a security group doesn't support access to a feature or they have been explicitly denied permission to a feature.	To discover if this is the cause, <a href="#">trace a permission</a> .
The user has been recently granted permission, however a refresh is required for their client to recognize the changes.	Have the user <a href="#">refresh or re-evaluate their permissions</a> .
The user's trying to exercise a feature granted only to a team administrator for a specific team, however they haven't been granted that role.	To add them to the role, see <a href="#">Add, remove team administrator</a> .
The user hasn't enabled a preview feature.	Have the user open the Preview features and determine the on/off status for the specific feature. For more information, see <a href="#">Manage preview features</a> .

ISSUE	TROUBLESHOOTING ACTION
Project member has been added to a limited scope security group, such as the Project-Spaced Users group.	To discover if this is a cause, <a href="#">look up the user's security group memberships</a> .

## Less common access and permission issues

Less common reasons for limited access are when one of the following events has occurred:

ISSUE	TROUBLESHOOTING ACTION
A project administrator disabled a service. In this case, no one has access to the disabled service.	To determine whether a service is disabled, see <a href="#">Turn an Azure DevOps service on or off</a> .
A Project Collection Administrator disabled a preview feature, which disables it for all project members in the organization.	See <a href="#">Manage preview features</a> .
Group rules governing the user's access level or project membership are restricting access.	See <a href="#">Determine a user's access level and subscription status</a> .
Custom rules have been defined to a work item type's workflow.	see <a href="#">Rules applied to a work item type that restrict select operation</a> .

## Determine a user's access level and subscription status

You can assign users or groups of users to one of the following access levels:

- Stakeholder
- Basic
- Basic + Test Plans
- Visual Studio subscription

For more information about access level restriction in Azure DevOps, see [Supported access levels](#).

To use Azure DevOps features, users must be added to a security group with the appropriate permissions. Users also need access to the web portal. Limitations to select features get based on the access level and security group to which a user is assigned.

Users can lose access for the following reasons:

REASON FOR LOSS OF ACCESS	TROUBLESHOOTING ACTION
The user's Visual Studio subscription has expired.	Meanwhile, this user can <a href="#">work as a Stakeholder</a> , or you can give the user Basic access until the user renews their subscription. After the user signs in, Azure DevOps restores access automatically.
The Azure subscription used for billing is no longer active.	All purchases made with this subscription are affected, including Visual Studio subscriptions. To fix this issue, visit the <a href="#">Azure account portal</a> .
The Azure subscription used for billing was removed from your organization.	Learn more about <a href="#">linking your organization</a>

Otherwise, on the first day of the calendar month, users who haven't signed in to your organization for the

longest time lose access first. If your organization has users who don't need access anymore, [remove them from your organization](#).

For more information about permissions, see [Permissions and groups](#) and the [Permissions lookup guide](#).

## Trace a permission

Use permission tracing to determine why a user's permissions aren't allowing them access to a specific feature or function. Learn how a user or an administrator can investigate the inheritance of permissions. To trace a permission from the web portal, open the permission or security page for the corresponding level. For more information, see [Request an increase in permission levels](#).

If a user's having permissions issues and you use default security groups or custom groups for permissions, you can investigate where those permissions are coming from by using our permissions tracing. Permissions issues could be because of delayed changes. It can take up to 1 hour for Azure AD group memberships or permissions changes to propagate throughout Azure DevOps. If a user's having issues that don't resolve immediately, wait a day to see if they resolve. For more information about user and access management, see [Manage users and access in Azure DevOps](#).

If a user's having permissions issues and you use default security groups or custom groups for permissions, you can investigate where those permissions are coming from by using our permissions tracing. Permissions issues could be because the user doesn't have the necessary access level.

Users can receive their effective permissions either directly or via groups.

Complete the following steps so administrators can understand where exactly those permissions are coming from and adjust them, as needed.

1. Select **Project settings > Permissions > Users**, and then select the user.

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' project. The left sidebar lists various project settings like Overview, Boards, Repos, Pipelines, Artifacts, and Compliance. A red box labeled '1' highlights the 'Project settings' button at the bottom of the sidebar. The main content area is titled 'Project Settings' for 'FabrikamFiber'. A red box labeled '2' highlights the 'Permissions' link under the 'General' section. The right panel is titled 'Permissions' with a red box labeled '3' and shows a list of users with their names and email addresses. The 'Users' tab is selected.

Name	Role
Customer service Build Service (fabrikamfiber)	Customer service Build Service (fabrikamfiber)
Jamal Hartnett	fabrikamfiber4@hotmail.com
Project Collection Build Service (fabrikamfiber)	Project Collection Build Service (fabrikamfiber)
Management team Build Service (fabrikamfiber)	Management team Build Service (fabrikamfiber)
Fabrikam Test Build Service (fabrikamfiberorg)	Fabrikam Test Build Service (fabrikamfiberorg)
FabrikamFiber Build Service (fabrikamfiberorg)	FabrikamFiber Build Service (fabrikamfiberorg)

You should now have a user-specific view that shows what permissions they have.

2. To trace why a user does or doesn't have any of the listed permissions, select the information icon next to the permission in question.

JH Jamal Hartnett

Permissions Member of

**General**

Delete team project	Allow (inherited)	(i)
Edit project-level information	Allow (inherited)	(i) 
Manage project properties	Allow (inherited)	(i)
Rename team project	Allow (inherited)	(i)
Suppress notifications for work item updates	Allow (inherited)	(i)
Update project visibility	Allow (inherited)	(i)
View project-level information	Allow (inherited)	(i)

The permission value is being inherited through your direct or indirect membership in these groups:  
[FabrikamFiber]\Project Administrators

**Boards**

Bypass rules on work item updates	Allow (inherited)	(i)
Change process of team project.	Allow (inherited)	(i)
Create tag definition	Allow (inherited)	(i)
Delete and restore work items	Allow (inherited)	(i)
Move work items out of this project	Allow (inherited)	(i)
Permanently delete work items	Allow (inherited)	(i)

**Analytics**

Delete shared Analytics views	Allow (inherited)	(i)
Edit shared Analytics views	Allow (inherited)	(i)
View analytics	Allow (inherited)	(i)

The resulting trace lets you know how they're inheriting the listed permission. You can then adjust the user's permissions by adjusting the permissions that are provided to the groups that they're in.

1. Select **Project settings > Security**, and then enter the user name into the filter box.

The screenshot shows the Azure DevOps Settings - Security page for the Fabrikam Fiber project. The left sidebar lists various project management sections like Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area is titled 'Project Settings' and contains a list of security-related options: General, Overview, Teams, Security (which is highlighted with a red box and number 2), Notifications, Service hooks, Dashboards, Boards, Project configuration, Team configuration, GitHub connections, Pipelines, Agent pools, Retention, Release retention, Service connections, Repos, Repositories, Policies, Test, and Retention. In the top right corner, there is a 'Create group' button with a red box and number 3, and below it is a 'Filter users and groups' search bar. A red box also highlights the 'Filter users and groups' search bar.

You should now have a user-specific view that shows what permissions they have.

2. Trace why a user does or doesn't have any of the listed permissions. Hover over the permission, and then choose **Why**.

**Permissions** Member of

Bypass rules on work item updates	Not set
Change process of team project.	Not set
Create tag definition	Allow (inherited)
Create test runs	Allow (inherited)
Delete and restore work items	Allow (inherited)
Delete shared Analytics views	Allow (inherited)
Delete team project	Not set
Delete test runs	Allow (inherited)
Edit project-level information	Not set
Edit shared Analytics views	Allow (inherited)
Manage project properties	Not set
Manage test configurations	Allow (inherited)
Manage test environments	Allow (inherited)
Move work items out of this project	Not set
Permanently delete work items	Not set
Rename team project	Not set
SUPPRESS notifications for work item updates	Not set
View analytics	Allow (inherited)
View project-level information	Allow (inherited) <a href="#">Why?</a>
View test runs	Allow (inherited)

[Clear explicit permissions](#)

[Save changes](#)

[Undo changes](#)

The resulting trace lets you know how they're inheriting the listed permission. You can then adjust the user's permissions by adjusting the permissions that are provided to the groups they're in.

**TRACE INHERITANCE RESULTS**

Permission	View project-level information
Effective value	Allow (Inherited)
Identity	Christie Church

**GROUP MEMBER INHERITANCE**

[Fabrikam Fiber]\Contributors	Allow
[Fabrikam Fiber]\Fabrikam Fiber Team	Allow

**Close**

1. Go to the Security page for the project that the user is having access problems.
2. Enter their name into the box in the upper left-hand corner.

FabrikamFiber / Fabrika... Security

Christie Church

FabrikamFiber > FabrikamFiber-tfvc Team

Permissions Members Member of

The default project team.

Bypass rules on work item updates	Allow (inherited)
Create tag definition	Allow (inherited)
Create test runs	Allow (inherited)
Delete and restore work items	Allow (inherited)
Delete team project	Allow (inherited)

You should have a user-specific view that shows what permissions they have.

3. Trace why a user does or doesn't have any of the listed permissions. Hover over the permission, and then choose **Why**.

The screenshot shows the Azure DevOps Security page for the 'Christie Church' group under the 'FabrikamFiber > Christie Church' hierarchy. The 'Permissions' tab is selected, displaying a list of permissions with their current status. A red box highlights the 'Why?' link next to the 'View project-level information' permission.

Permission	Status
Bypass rules on work item updates	Allow (inherited)
Create tag definition	Allow (inherited)
Create test runs	Allow (inherited)
Delete and restore work items	Allow (inherited)
Delete team project	Allow (inherited)
Delete test runs	Allow (inherited)
Edit project-level information	Allow (inherited)
Manage project properties	Allow (inherited)
Manage test configurations	Allow (inherited)
Manage test environments	Allow (inherited)
Move work items out of this project	Allow (inherited)
Permanently delete work items	Allow (inherited)
Rename team project	Allow (inherited)
Suppress notifications for work item updates	Allow (inherited)
Update project visibility	Allow (inherited)
View analytics	Allow (inherited)
View project-level information	Allow (inherited) <a href="#">Why?</a>
View test runs	Allow (inherited)

[Clear explicit permissions](#)

[Save changes](#) [Undo changes](#)

The resulting trace lets you know how they're inheriting the listed permission. You can then adjust the user's permissions by adjusting those permissions provided to the groups they're in.

For more information, see [Grant or restrict access to select features and functions](#) or [Request an increase in permission levels](#).

## Refresh or reevaluate permissions

See the following scenario where refreshing or reevaluating permissions may be necessary.

### Problem

Users get added to an Azure DevOps or Azure AD group. This action grants inherited access to an organization or project. But, they don't get access immediately. Users must either wait or sign out, close their browser, and then sign back in to get their permissions refreshed.

Users get added to an Azure DevOps group. This action grants inherited access to an organization or project. But, they don't get access immediately. Users must either wait or sign out, close their browser, and then sign back in to get their permissions refreshed.

### Solution

Within [User settings](#), on the [Permissions](#) page, you can select **Re-evaluate permissions**. This function re-evaluates your group memberships and permissions, and then any recent changes take effect immediately.

The screenshot shows the 'User settings' page in Azure DevOps. On the left, there's a sidebar with options like Account, Profile, Time and Locale, Permissions (which is highlighted with a red box), Preferences, Notifications, and Theme. On the right, under the 'Permissions' heading, there's a button labeled 'Re-evaluate permissions' with a red box around it. Below the button, there's a note: 'This tool will re-evaluate your group memberships and permissions; any recent changes will take effect immediately. Normally, changes to group memberships and permissions can take up to one hour to reflect in Azure DevOps.'

## Rules applied to a work item type that restrict select operations

Before you customize a process, we recommend that you review [Configure and customize Azure Boards](#), which provides guidance on how to customize Azure Boards to meet your business needs.

For more information about work item type rules that apply toward restricting operations, see:

- [Apply rules to workflow states \(Inheritance process\)](#)
- [Sample rule scenarios](#)
- [Define area paths and assign to a team](#)

## Hide organization settings from users

If a user's limited to seeing only their projects, or from seeing the organization settings, the following information may explain why. To restrict users from accessing organization settings, you can enable the **Limit user visibility and collaboration to specific projects** preview feature.

Examples of restricted users include Stakeholders, Azure Active Directory (Azure AD) guest users, or members of a security group. Once enabled, any user or group added to the Project-Spaced Users group gets restricted from accessing the Organization Settings pages, except for Overview and Projects. They're restricted to accessing only those projects to which they've been added.

Examples of restricted users include Stakeholders, or members of a security group. Once enabled, any user or group added to the Project-Spaced Users group gets restricted from accessing the Organization Settings pages, except for Overview and Projects. They're restricted to accessing only those projects to which they've been added.

For more information about hiding organization settings from users, see [Manage your organization, Limit user visibility for projects and more](#).

## View, add, and manage permissions with CLI

You can view, add, and manage permissions at a more granular level with the `az devops security permission` commands. For more information, see [Manage permissions with command line tool](#).

### Use tools to fix permission

You can use the following tools to fix a user's permission issue.

- **TFSSecurity.exe** - TFSSecurity is a command-line tool that can be used to view and update and delete permissions or groups.

Example usage:

```
tfssecurity /a+ Identity "81e4e4b5-bde0-4f2c-a7a5-4d25c2e8a81f\" Read "Project Collection Valid Users"
ALLOW /collection:{collectionUrl}
tfssecurity /a- Identity "3c7a0a47-27b4-4def-8d42-aab9b405fc8a\" Write n: "[Project1]\Contributors"
DENY /collection:{collectionUrl}
```

- Use the public sproc

Example usage: Use `prc_pSetAccessControlEntry` or `prc_pRemoveAccessControlEntries` to add or remove ACEs directly from the security tables if TFSSecurity doesn't work for you.

For more information, see [Use TFSSecurity to manage groups and permissions for Azure DevOps](#).

## Group rules with lesser permissions

Group rule types get ranked in the following order: Subscriber > Basic + Test Plans > Basic > Stakeholder. Users always get the best access level between all the group rules, including Visual Studio (VS) subscription.

See the following examples, showing how subscriber detection factors into group rules.

### Example 1: Group rule gives me more access

If I have a VS Pro subscription and I'm in a group rule that gives me Basic + Test Plans – what happens?

Expected: I get Basic + Test Plans because what the group rule gives me is greater than my subscription. Group rule assignment always provides the greater access, rather than limiting access.

### Example 2: Group rule gives me the same access

I have a Visual Studio Test Pro subscription and I'm in a group rule that gives me Basic + Test Plans – what happens?

Expected: I get detected as a Visual Studio Test Pro subscriber, because the access is the same as the group rule. I'm already paying for the Visual Studio Test Pro, so I don't want to pay again.

## Work with GitHub

See the following troubleshooting information for when you're trying to deploy code in Azure DevOps with GitHub.

### Problem

You can't bring the rest of your team into the organization and project, despite adding them as organization and project members. They receive emails but when signing in they receive an error 401.

### Solution

You're likely signed into Azure DevOps with an incorrect identity. Complete the following steps.

1. Close all browsers, including browsers that aren't running Azure DevOps.
2. Open a private or incognito browsing session.
3. Go to the following URL: <https://aka.ms/vssignout>.

A message displays that says, "Sign out in progress." After you sign out, you're redirected to [dev.azure.microsoft.com](http://dev.azure.microsoft.com).

4. Sign in to [Azure DevOps](#) again. Select your other identity.

## Other areas where permissions might be applied

- [Area path permissions](#)

- [Work item tags](#)
- [Moved work items out of a project](#)
- [Deleted work items](#)
- [Quick guide to default permissions and access for Azure Boards](#)
- [Custom rules](#)
- [Sample custom rule scenarios](#)
- [Custom backlogs and boards](#)
- [Custom controls](#)

## Related articles

- [Manage permissions with the command line tool](#)
- [Change individual or group permissions](#)
- [Security best practices](#)
- [Security and permission management tools](#)
- [Add users to an administrator role](#)

# Allowed IP addresses and domain URLs

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

If your organization's secured with a firewall or proxy server, you must add certain internet protocol (IP) addresses and domain uniform resource locators (URLs) to the **allowlist**. Adding these IPs and URLs to the allowlist helps to ensure that you have the best experience with Azure DevOps. You know that you need to update your allowlist if you can't access Azure DevOps on your network. See the following sections in this article:

- [Domain URLs to allow](#)
- [IP addresses and range restrictions](#)

## TIP

So that Visual Studio and Azure Services work well with no network issues, you should open select ports and protocols. For more information, see [Install and use Visual Studio behind a firewall or proxy server](#), [Use Visual Studio and Azure Services](#).

## Domain URLs to allow

Network connection issues could occur because of your security appliances, which may be blocking connections  
- Visual Studio uses TLS 1.2 and above. When you're using [NuGet](#) or connecting from Visual Studio 2015 and later, update the security appliances to support TLS 1.2 and above for the following connections.

To ensure your organization works with any existing firewall or IP restrictions, ensure that `dev.azure.com` and `*.dev.azure.com` are open.

The following section includes the most common domain URLs to support sign in and licensing connections.

```
https://dev.azure.com
https://*.dev.azure.com
https://aex.dev.azure.com
https://aexprode1.vsaex.visualstudio.com
https://*vstmrblob.vsassets.io
https://amp.azure.net
https://app.vssps.dev.azure.com
https://app.vssps.visualstudio.com
https://*.vsblob.visualstudio.com
https://*.vssps.visualstudio.com
https://*.vstmr.visualstudio.com
https://azure.microsoft.com
https://go.microsoft.com
https://graph.microsoft.com
https://login.microsoftonline.com
https://management.azure.com
https://management.core.windows.net
https://microsoft.com
https://microsoftonline.com
https://static2.sharepointonline.com
https://visualstudio.com
https://vsrm.dev.azure.com
https://vstsagentpackage.azureedge.net
https://*.windows.net
https://{{organization_name}}.visualstudio.com
https://{{organization_name}}.vsrm.visualstudio.com
https://{{organization_name}}.vstmr.visualstudio.com
https://{{organization_name}}.pkgs.visualstudio.com
https://{{organization_name}}.vssps.visualstudio.com
```

Azure DevOps uses content delivery network (CDN) to serve static content. The following URLs are part of that.

```
https://cdn.vsassets.io
https://*.vsassets.io
https://*gallerycdn.vsassets.io
https://aadcdn.msauth.net
https://aadcdn.msftauth.net
https://amcdn.msftauth.net
https://azurecomcdn.azureedge.net
```

The following endpoints are used to authenticate Azure DevOps organizations using a Microsoft Account (MSA). These are only needed for Azure DevOps organizations backed by Microsoft Accounts (MSA).

Azure DevOps organizations backed an Azure Active Directory tenant does not need the following URLs.

```
https://live.com
https://login.live.com
```

The following URLs are required if you are migrating from Azure DevOps server to the cloud service using our data migration tool.

```
https://dataimport.dev.azure.com
```

## Various domain URL descriptions

- <https://login.microsoftonline.com>: authentication and sign-in related
- <https://app.vssps.visualstudio.com>: authentication and sign-in related
- [https://\\*.vssps.visualstudio.com](https://*.vssps.visualstudio.com): authentication and sign-in related
- [https://\\*gallerycdn.vsassets.io](https://*gallerycdn.vsassets.io): hosts Azure DevOps extensions
- [https://\\*vstmrblob.vsassets.io](https://*vstmrblob.vsassets.io): hosts Azure DevOps TCM log data
- <https://cdn.vsassets.io>: hosts Azure DevOps Content Delivery Networks (CDNs) content
- <https://static2.sharepointonline.com>:  
hosts some resources that Azure DevOps uses in "office fabric" UI kit for fonts, and so on

- <https://vsrm.dev.azure.com>: hosts releases
- <https://vstsagentpackage.azureedge.net>: required to setup self-hosted agent in machines within your network
- <https://amp.azure.net>: needed for deploying to Azure app service
- <https://go.microsoft.com>: access go links

We recommend you open port `443` to all traffic on these IP addresses and domains. We also recommend you open port `22` to a smaller subset of targeted IP addresses.

#### NOTE

Azure DevOps uses Content Delivery Networks (CDNs) to serve static content. Users in **China** should also add the following domain URLs to an allowlist:

```
https://*.vsassetscdn.azure.cn  
https://*.gallerycdn.azure.cn
```

## More domain URLs

### Azure Artifacts

Ensure the following domain URLs are allowed for Azure Artifacts:

```
https://*.blob.core.windows.net  
https://*.visualstudio.com
```

Also allow all IP addresses in the "name": "Storage.{region}" section of the following file (updated weekly): [Azure IP ranges and Service Tags - Public Cloud](#). {region} is the same Azure Geography as your organization.

### NuGet connections

Ensure the following domain URLs are allowed for NuGet connections:

```
https://azurewebsites.net  
https://nuget.org
```

#### NOTE

Privately owned NuGet server URLs might not be included in the previous list. You can check the NuGet servers you're using by opening `%APPData%\Nuget\NuGet.Config`.

### SSH connections

If you need to connect to Git repositories on Azure DevOps with SSH, allow requests to port 22 for the following hosts:

```
ssh.dev.azure.com  
vs-ssh.visualstudio.com
```

Also allow IP addresses in the "name": "AzureDevOps" section of [this downloadable file](#) (updated weekly) named: **Azure IP ranges and Service Tags - Public Cloud**

### Azure Pipelines Microsoft-hosted agents

If you use Microsoft-hosted agent to run your jobs and you need the information about what IP addresses are

used, see [Microsoft-hosted agents IP ranges](#). See all [Azure virtual machine scale set agents](#).

For more information about hosted Windows, Linux and macOS agents, see [Microsoft-hosted agent IP ranges](#).

## Azure Pipelines Self-hosted agents

If you're running a firewall and your code is in Azure Repos, see [Self-hosted Linux agents FAQs](#), [Self-hosted macOS agents FAQs](#) or [Self-hosted Windows agents FAQs](#). This article has information about which domain URLs and IP addresses your private agent needs to communicate with.

# IP addresses and range restrictions

## Outbound connections

*Outbound connections* originate from inside your organization and target Azure DevOps or other dependent sites. Examples of such connections include:

- Browsers connecting to Azure DevOps website as users go to and use features of Azure DevOps
- Azure Pipelines agents installed on your organization's network connecting to Azure DevOps to poll for pending jobs
- CI events sent from a source code repository that's hosted within your organization's network to Azure DevOps

Ensure the following IP addresses are allowed for outbound connection, so your organization works with any existing firewall or IP restrictions. The endpoint data in the following chart lists requirements for connectivity from a machine in your organization to Azure DevOps Services.

- [IP V4 ranges](#)
- [IP V6 ranges](#)

13.107.6.0/24
13.107.9.0/24
13.107.42.0/24
13.107.43.0/24

If you're currently allowing the `13.107.6.183` and `13.107.9.183` IP addresses, leave them in place, as you don't need to remove them.

### NOTE

[Azure Service Tags](#) aren't supported for *outbound* connection.

## Inbound connections

*Inbound connections* originate from Azure DevOps and target resources within your organization's network. Examples of such connections include:

- Azure DevOps Services connecting to endpoints for [Service Hooks](#)
- Azure DevOps Services connecting to customer-controlled SQL Azure VMs for [Data Import](#)
- Azure Pipelines connecting to on-premises source code repositories such as [GitHub Enterprise](#) or [Bitbucket Server](#)
- Azure DevOps Services [Audit Streaming](#) connecting to on-premises or cloud-based Splunk

Ensure the following IP addresses are allowed for inbound connection, so your organization works with any existing firewall or IP restrictions. The endpoint data in the following chart lists requirements for connectivity

from Azure DevOps Services to your on-premises or other cloud services.

REGION	IP V4 RANGES
Australia East	20.37.194.0/24
Australia South East	20.42.226.0/24
Brazil South	191.235.226.0/24
Central Canada	52.228.82.0/24
Asia Pacific (Singapore)	20.195.68.0/24
South India	20.41.194.0/24
Central United States	20.37.158.0/23
West Central United States	52.150.138.0/24
East United States	20.42.5.0/24
East 2 United States	20.41.6.0/23
North United States	40.80.187.0/24
South United States	40.119.10.0/24
West United States	40.82.252.0/24
West 2 United States	20.42.134.0/23
Western Europe	40.74.28.0/23
United Kingdom South	51.104.26.0/24

Azure Service Tags are supported for *inbound* connection. Instead of allowing the previously listed IP ranges, you may use the [AzureDevOps](#) service tag for Azure Firewall and Network Security Group (NSG) or on-premises firewall via a JSON file download.

#### NOTE

The Service Tag or previously mentioned inbound IP addresses don't apply to Microsoft Hosted agents. Customers are still required to allow the [entire geography for the Microsoft Hosted agents](#). If allowing the entire geography is a concern, we recommend using the [Azure Virtual Machine Scale Set agents](#). The Scale Set agents are a form of self-hosted agents that can be auto-scaled to meet your demands.

Hosted macOS agents are hosted in GitHub's macOS cloud. IP ranges can be retrieved using the [GitHub metadata API](#) using the instructions provided [here](#).

#### Other IP addresses

Most of the following IP addresses pertain to Microsoft 365 Common and Office Online.

```
40.82.190.38
52.108.0.0/14
52.237.19.6
52.238.106.116/32
52.244.37.168/32
52.244.203.72/32
52.244.207.172/32
52.244.223.198/32
52.247.150.191/32
```

For more information, see [Worldwide endpoints](#) and [Adding IP address rules](#).

### Azure DevOps ExpressRoute connections

If your organization uses ExpressRoute, ensure the following IP addresses are allowed for both outbound and inbound connections.

- [IP V4 ranges](#)
- [IP V6 ranges](#)

```
13.107.6.175/32
13.107.6.176/32
13.107.6.183/32
13.107.9.175/32
13.107.9.176/32
13.107.9.183/32
13.107.42.18/32
13.107.42.19/32
13.107.42.20/32
13.107.43.18/32
13.107.43.19/32
13.107.43.20/32
```

For more information about Azure DevOps and ExpressRoute, see [ExpressRoute for Azure DevOps](#).

## Azure DevOps import service

During the import process, we highly recommend that you restrict access to your virtual machine (VM) to only IP addresses from Azure DevOps. To restrict access, allow only connections from the set of Azure DevOps IP addresses, which were involved in the collection database import process. For information about identifying the correct IP addresses, see [\(Optional\) Restrict access to Azure DevOps Services IPs only](#).

## Related articles

- [Available service tags](#)
- [Microsoft-hosted agents IP address ranges](#)
- [Self-hosted Windows agents FAQs](#)
- [Configure Azure Storage firewalls and virtual networks](#)
- [Install and use Visual Studio behind a firewall or proxy server](#)

# Get support and provide feedback

12/13/2022 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Share your feedback and ideas with us, or join our communities. We're always working to improve Azure DevOps, and we want you to be part of the process!

Do you need to do any of the following?

- **Get advice** Visit StackOverflow for [Azure DevOps Services](#) or [Azure DevOps Server](#).
- **Report a bug** Submit it through our Developer Community for [Azure DevOps Services](#) or [Azure DevOps Server](#).
- **Suggest a feature or a fix** Submit your idea or issue through our Developer Community for [Azure DevOps Services](#) or [Azure DevOps Server](#).
- **Find out what's new in Azure DevOps** Check out the [current Azure DevOps Release Notes](#). These notes are updated every three weeks.
- **Chat with our virtual support agent** Get help with common issues, troubleshooting steps, or create a request to change the region your Azure DevOps instance is hosted in using our [virtual support agent](#).

## Get live help

We offer a [live chat](#) (English only) support option. Choose from **Technical Support**, **Sales Support**, **Visual Studio (For your Company)**, and **Account, Subscription, and Billing Support**. Select your country/region from the dropdown menu, and then select **Live Chat (English)**.

## Documentation feedback

All articles on Microsoft Learn have a ratings tool in the lower right-hand corner of the page. It asks "Is this content helpful?" Answer **Yes** or **No** depending on your experience.

Add more detailed feedback by selecting the **Tell us more** link after selecting **Yes** or **No**. Check an appropriate box and enter what we can do to improve the content for you! Although we can't reply back, we collect and review this feedback regularly, and use your sentiments in our content planning.

## Tips for effective feedback

If you just want to vent about the product or the documentation, that's okay. It helps us a lot to know when you're happy or unhappy with an experience. For the most impact, though, provide details so we can better understand what we're doing right or wrong.

- Provide a little context. What problem were you trying to solve? At what point did it go wrong?
- What's your role? We don't need personal or professional details. Are you a dev? A manager? A business owner? When we understand our audience, we can come up with better solutions for you and other customers doing similar work.
- What version of the product were you using? What other products were you using with it?

The best feedback we get is clear and precise. For example:

- Product feedback: "I'm a project manager for a small start-up. I'm using Azure DevOps. I'm trying to create work item templates through the UI, but my changes don't seem to persist. It's not clear what I'm doing

wrong."

- Doc feedback: "I'm a dev in a large organization that works on Java apps. I tried to use Maven with your build system in Azure DevOps Server 2017 Update 1 (15.112.26307.0), but I couldn't get the configuration shown in the documentation to work."

The more details, the better!

## Related articles

- [Azure DevOps features timeline](#)
- [Report a problem with Visual Studio](#)

# Look up your Azure DevOps platform and version

12/13/2022 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can tell what platform you use by opening the **About** page for Azure DevOps Services or Azure DevOps Server.

## Azure DevOps Services

Enter the following URL for your organization, specifying the organization name.

`https://dev.azure.com/YourOrganizationName/_home/About`

A page similar to the following example opens showing the version number.

The screenshot shows the 'About Azure DevOps Service' page. At the top, there's a logo for 'Azure DevOps'. Below it, the title 'About Azure DevOps Service' is centered. Underneath the title, the text 'Version Dev19.M206.1 (AzureDevOps\_M206\_20220801.2)' is displayed. At the bottom, there's a copyright notice: '© Microsoft Corporation. All rights reserved.'

## Azure DevOps Server

Open the **About** page from the profile menu as shown in the following image.

The screenshot shows a browser window for 'devops-2022 - Microsoft Azure'. The address bar shows the URL 'Not secure | devops-2022/DefaultCollection'. The main content area displays the 'DefaultCollection' dashboard for the project 'FabrikamFiber'. On the right side, a user profile is shown with the name 'Jamal Hartnett' and email 'fabrikamfiber4@hotmail.com'. A dropdown menu is open from the user profile icon. The menu items are: 'My profile', 'Security', 'Notification settings', 'Theme', 'Help' (which is highlighted with a red box), 'Sign in as...', and 'Sign out'. A red arrow points from the 'About' link at the bottom of the page to the 'Help' menu item in the profile dropdown.

The corresponding browser URL is:

`https://ServerName/CollectionName/_home/About`

A page similar to the following image opens showing the version number.



## About Azure DevOps Server

Version Azure DevOps Server 2022 RC1 (AzureDevOpsServer\_20220720.1)

© Microsoft Corporation. All rights reserved.

## Version numbers

ON-PREMISES RELEASE	VERSION	ABOUT HELP PAGE VERSION	BUILD NUMBER
Azure DevOps Server 2022	RC1	(AzureDevOpsServer_20220720.1)	19.205.32728.1
Azure DevOps Server 2020	Update 1.1	Version Azure DevOps Server 2020 Update 1.2	18.181.31230.2
Azure DevOps Server 2020	RTW	Version Dev 18.M170.9	18.170.30830.2
Azure DevOps Server 2019	Update 1	Version Dev17.M153.6	17.153.29522.3

## Related articles

- [Azure DevOps features timeline](#)
- [Report a problem with Visual Studio](#)

# Navigate in Visual Studio Team Explorer

12/13/2022 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You use Team Explorer to coordinate your code efforts with other team members to develop a software project. In addition, you can manage work and that is assigned to you, your team, or your projects. Team Explorer is a plug-in that installs with Visual Studio and Team Explorer Everywhere is a plug-in that installs with Eclipse. Developers can effectively collaborate using Team Explorer connected to projects hosted on Azure DevOps Services or an on-premises Azure DevOps Server (previously named Team Foundation Server (TFS)).

## TIP

You can install the latest version of Visual Studio clients from the [Visual Studio downloads page](#).

Additional options for connecting to Azure DevOps Services or TFS include:

- [Team Explorer Everywhere](#)
- [Azure DevOps Plugin for Android Studio](#)
- [Azure DevOps Plugin for IntelliJ](#)
- [Visual Studio Code](#)

For information about compatibility among client and server versions, see [Requirements and compatibility](#).

If you don't need Visual Studio, but want to connect to a project in Azure DevOps, you can install the free [Visual Studio Community](#).

## Prerequisites

- You must have a project in Azure DevOps. If you need to add a project, see [Create a project](#).
- You must be a member of the project you connect to. To get added, see [Add users to a project or team](#).

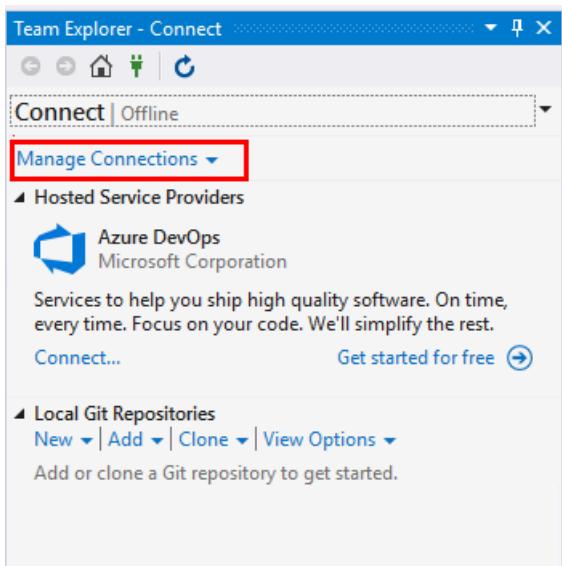
## Connect to a project or repository

Team Explorer connects Visual Studio to projects in Azure DevOps. You can manage source code, work items, and builds. The operations available to you depend on which source control option—Git or Team Foundation version control (TFVC)—was selected to manage source code when the project was created.

## TIP

If you open Visual Studio and the Team Explorer pane doesn't appear, choose the [View>Team Explorer](#) menu option from the tool bar.

From the [Connect](#) page, you can select the projects you want to connect to and quickly switch connection to a different project and or repository. For details, see [Connect to a project](#).



The Git and TFVC repos support different pages and functions. For a comparison of the two version control systems, see [Choosing the right version control for your project](#).

## Git version control and repository

The following images show the pages available when you connect to a Git repository from Team Explorer.

### NOTE

Visual Studio 2019 version 16.8 and later versions provide a new Git menu for managing the Git workflow with less context switching than Team Explorer. Procedures provided in this article under the Visual Studio tab provide information for using the Git experience as well as Team Explorer. To learn more, see [Side-by-side comparison of Git and Team Explorer](#).

VISUAL STUDIO 2019	VISUAL STUDIO 2017	VISUAL STUDIO 2015
A screenshot of the 'Team Explorer - Home' window in Visual Studio 2019. It shows a connection to 'Fabrikam Fiber' via Azure DevOps. The main pane displays a list of actions: Changes, Branches, Pull Requests, Sync, Tags, Work Items, Builds, and Settings. The 'Changes' item is currently selected.	A screenshot of the 'Team Explorer - Home' window in Visual Studio 2017. It shows a connection to 'Fabrikam Git'. The main pane displays a list of actions: Changes, Branches, Pull Requests, Sync, Tags, Work Items, Builds, and Settings. The 'Changes' item is currently selected.	A screenshot of the 'Team Explorer - Home' window in Visual Studio 2015. It shows a connection to 'Fabrikam Fiber'. The main pane displays a list of actions: Changes, Branches, Pull Requests, Sync, Tags, Work Items, Builds, and Settings. The 'Changes' item is currently selected.

To learn more about each page, see the following articles.

## Home and Builds

### Git version control

## Work items

---

### Home

- [Web portal](#)
- [Task Board](#)

### Builds

- [Create build pipelines](#)
- [View and manage builds](#)
- [Manage the build queue](#)
- [Install Continuous Delivery \(CD\) Tools for Visual Studio](#)
- [Configure and execute Continuous Delivery \(CD\) for your app](#)
- [Create a new repo](#)
- [Clone an existing repo](#)
- **Changes:** [Save work with commits](#)
- **Branches:** [Create work in branches](#)
- **Pull Requests:** [Review code with pull requests"](#)
- **Sync:** [Update code with fetch and pull](#)
- **Tags:** [Work with Git tags](#)
- [Git preferences](#)
- [Git command reference](#)

### Default experience (Visual Studio 2019 and later versions)

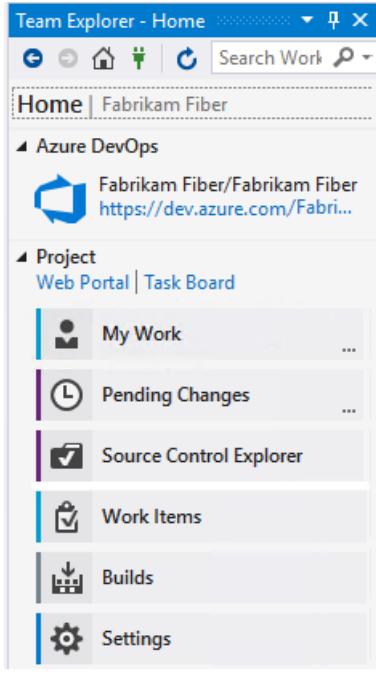
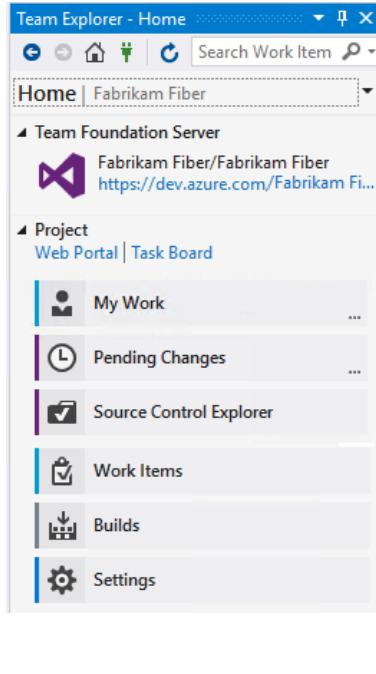
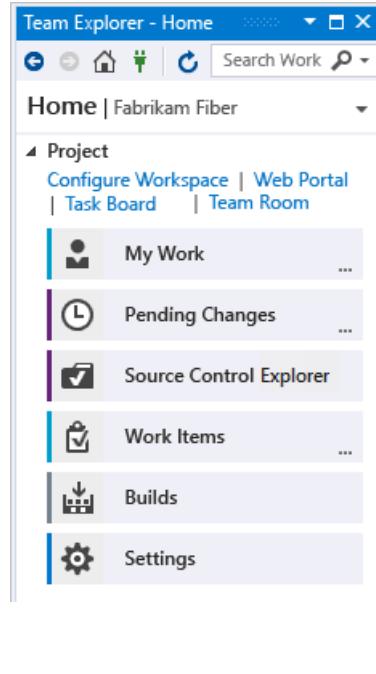
- [View and add work items](#)
- [Set the Work Items experience in Visual Studio](#)

### Legacy experience (All Visual Studio versions)

- [Add work items](#)
- [Query editor](#)
- [Query folders](#)
- [Query permissions](#)
- [Open query in Excel](#)
- [Email query results using Outlook](#)
- [Create reports from query in Excel](#)

## Team Foundation version control

The following images show the pages available when you connect to a TFVC repository from Team Explorer.

VISUAL STUDIO 2019	VISUAL STUDIO 2017	VISUAL STUDIO 2015
		

To learn more about each page, see the following articles.

## Home and Builds

### TFVC

### Work items

#### Home

- [Web portal](#)
- [Task Board](#)

#### Builds

- [Create build pipelines](#)
- [View and manage builds](#)
- [Manage the build queue](#)
- [Install Continuous Delivery \(CD\) Tools for Visual Studio](#)
- [Configure and execute Continuous Delivery \(CD\) for your app](#)
  
- [Configure workspace](#)
- [Suspend/resume work, Code review](#)
- **Pending Changes:** [Manage pending changes](#), [Find shelvesets](#), [Resolve conflicts](#)
- **Source Control Explorer:** [Add/view files and folders](#)
- [Add Check-In Policies](#)
- [Version control commands](#)

#### Default experience (Visual Studio 2019 and later versions)

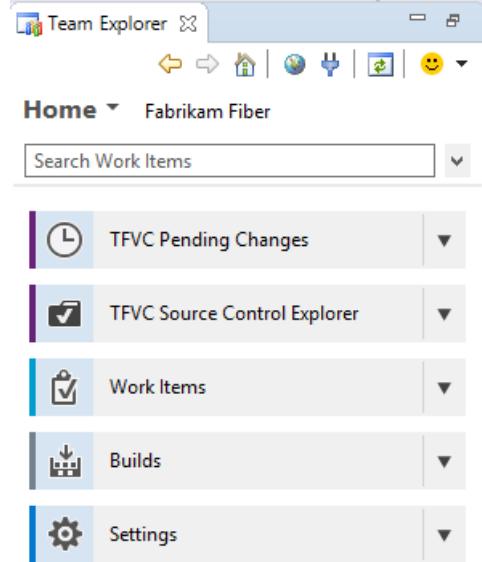
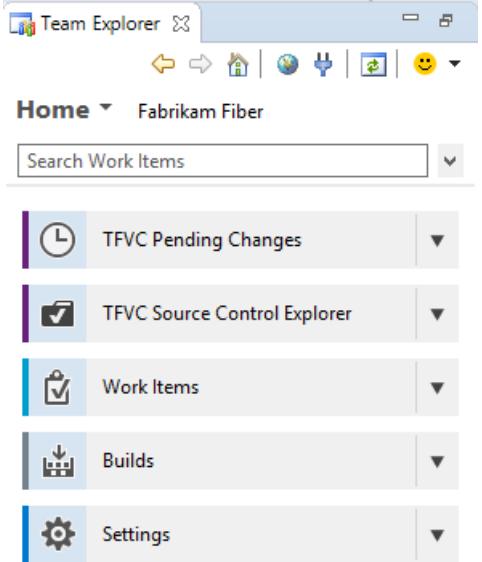
- [View and add work items](#)
- [Set the Work Items experience in Visual Studio](#)

#### Legacy experience (All Visual Studio versions)

- [Add work items](#)
- [Query editor](#)
- [Query folders](#)
- [Query permissions](#)
- [Open query in Excel](#)
- [Email query results using Outlook](#)
- [Create reports from query in Excel](#)

## Team Explorer plug-in for Eclipse

If you work in Eclipse or on a non-Windows platform, you can [install the Team Explorer plug-in for Eclipse](#). Once installed, you can share your Eclipse projects by adding them to Azure DevOps Services or TFS using [Git](#) or [TFVC](#).

HOME PAGE WITH GIT (ECLIPSE)	HOME PAGE WITH TFVC (ECLIPSE)
	

To learn more about each page, see the following articles.

### Home and Builds

#### Version control

#### Work items

### Home

- [Web portal](#)

### Builds

- [Create build pipelines](#)
- [View and manage builds](#)
- [Manage the build queue](#)
- [Install Continuous Delivery \(CD\) Tools for Visual Studio](#)
- [Configure and execute Continuous Delivery \(CD\) for your app](#)

### Git repo

- [Share your code](#)
- [Git preferences](#)
- [Git command reference](#)

#### TFVC repo

- [Share your code](#)
- [Pending changes](#)
- [Source Control Explorer](#)
- [Add Check-In Policies](#)
- [Version control commands](#)
- [Add work items](#)
- [Query editor](#)
- [Query folders](#)
- [Query permissions](#)

## Reports

### NOTE

Some pages, such as **Reports**, only appear when an on-premises TFS is configured with the required resources, such as SQL Server Reporting Services.

The **Reports** page opens the [Reporting Services report site](#). This page appears only when your project has been configured with SQL Server Analysis Services and Reporting Services. Also, the option to [Create Report in Microsoft Excel](#) appears only when reporting has been configured for the project.

If your project is missing one or more pages, you may be able to [add functionality to your on premises TFS deployment](#).

## Settings

From the **Settings** page, you can configure administrative features for either a project or project collection. To learn more about each page, see the following articles. Most of the links open to a web portal administration page. Not all settings are available from the Team Explorer plug-in for Eclipse.

### Project

- [Security, Group Membership](#)
- [Security, Source Control \(TFVC\)](#)
- [Work Item Areas](#)
- [Work Item Iterations](#)
- [Portal Settings](#)
- [Project Alerts](#)

### Project Collection

- [Security, Group Membership](#)
- [Source Control \(TFVC\)](#)
- [Process Template Manager](#)
- [Other](#)
- [Git Global Settings](#)

- [Git Repository Settings](#)

To learn more about settings, see [About team, project, and organizational-level settings](#).

## Refresh Team Explorer or Team Explorer Everywhere

If data doesn't appear as expected, the first thing to try is to refresh your client. Refreshing your client updates the local cache with changes that were made in another client or in TFS. To refresh Team Explorer, do one of the following actions:

- To refresh a page that you are currently viewing, choose Refresh in the menu bar (or choose F5).
- To refresh the project you currently have selected, choose Home, and then choose Refresh (or choose F5).
- To refresh the set of teams defined for the project that you currently have selected, choose Connect, and then choose Refresh (or choose F5).

To avoid potential errors, you should refresh your client application under the following circumstances:

- Process changes are made.
- Work item type definitions are added, removed, renamed, or updated.
- Area or iteration paths are added, removed, renamed, or updated.
- Users are added to or removed from security groups, or permissions are updated.
- A team member adds a new shared query or changes the name of a shared query.
- A build pipeline is added or deleted.
- A team or project is added or deleted.

### Resolve images that don't display in Team Explorer

If an inline image isn't displayed in a work item form that you view in Visual Studio Team Explorer, but the image is displayed in the web portal, your credentials might have expired. You can resolve this by completing the following steps:

1. In Visual Studio, select **View > Other Windows > Web Browser** (or use the shortcut **Ctrl+Alt+R**).
2. In the web browser, locate your organization.
3. Sign in with your credentials.
4. Refresh your work item in Team Explorer.

## Related articles

- [Troubleshoot connection](#)
- [Create a project](#)

### Additional tools provided with TFS Power Tools

By installing [TFS Power Tools](#), you gain access to these additional tools through the Team Explorer plug-in for Visual Studio:

- Process Template Editor
- Additional check-in policies for Team Foundation Version Control
- Team Explorer enhancements including Team Members
- Team Foundation Power Tool Command Line
- Test Attachment Cleaner
- Work Item Templates

Additional requirements may apply.



# Service and rate limits for Azure DevOps Services

12/13/2022 • 2 minutes to read • [Edit Online](#)

## Azure DevOps Services

Azure DevOps Services, like many Software-as-a-Service solutions, uses multi-tenancy to reduce costs and to enhance scalability and performance. This leaves users vulnerable to performance issues and even outages when other users of their shared resources have spikes in their consumption. To combat these problems, Azure DevOps Services limits the resources individuals can consume and the number of requests they can make to certain commands. When these limits are exceeded, subsequent requests may be either delayed or blocked.

This article specifies certain limits placed on the use and configuration of Azure DevOps services. For more information, see [Rate limits](#) and [Work tracking, process, and project limits](#).

## Work items

- A long text field can contain 1M characters.
- You can't assign more than 100 tags to a work item.
- You can't add more than 1,000 links to a work item.
- You can't add more than 100 attachments to a work item.
- You can't add an attachment size larger than 60 MB to a work item.
- You can have up to 1,000 tasks on a task board
- You can have up to 10,000 work items on a backlog
- You are limited to 5,000 teams in a project
- You can't create more than 150,000 tag definitions per project

## Area and iteration paths

CONFIGURATION OBJECT	LIMIT
Projects	1000 per organization for Azure DevOps Services No prescribed limit for on-premises (See also <a href="#">Work tracking, process, and project limits</a> )
Teams	5,000 per organization
Work item tags	150,000 tag definitions per organization
Area Paths	10,000 per organization
Area Path Depth	14
Area Paths per team	300
Iteration Paths	10,000 per organization
Iteration Path Depth	14
Iteration Paths per team	300

## Queries

- The execution time limit for queries is 30 seconds. See [Optimization best practices](#) to improve query performance.
- Query results are limited to 20,000
- Queries are limited in length to 32,000 characters

## Process customization

When customizing the work item types (WITs) defined in the Inheritance or Hosted XML process models, be aware of the limits placed on objects defined in [Work tracking, process, and project limits](#).

## Dashboards

A limit is placed on 500 dashboards per project.

## Wiki

Wikis defined for a project are limited to 1 GB per git repository.

**TIP**

To derive the size of a wiki/git repository, download the repo to your local computer, unzip the file, and then open the [Properties](#) for the corresponding folder.

## Data import

- Limited to 300 projects per organization
- See [data import documentation](#) for details

## Related articles

- [Rate limits](#)
- [Work tracking, process, and project limits](#)
- [Configure and customize Azure Boards](#)
- [Usage monitoring](#)

# Get started with Azure DevOps CLI

12/13/2022 • 2 minutes to read • [Edit Online](#)

## Azure DevOps Services

With the Azure DevOps extension for Azure Command Line Interface (CLI), you can manage many Azure DevOps Services from the command line. CLI commands enable you to streamline your tasks with faster and flexible interactive canvas, bypassing user interface workflows.

### NOTE

The Azure DevOps Command Line Interface (CLI) is only available for use with Azure DevOps Services. The Azure DevOps extension for the Azure CLI does not support any version of Azure DevOps Server.

To start using the Azure DevOps extension for Azure CLI, perform the following steps:

1. Install Azure CLI: Follow the instructions provided in [Install the Azure CLI](#) to set up your Azure CLI environment. At a minimum, your Azure CLI version must be 2.10.1. You can use `az --version` to validate.
2. Add the Azure DevOps extension:

```
az extension add --name azure-devops
```

You can use `az extension list` or `az extension show --name azure-devops` to confirm the installation.

3. Sign in: Run `az login` to sign in. Note that we support only interactive or log in using user name and password with `az login`. To sign in using a Personal Access Token (PAT), see [Sign in via Azure DevOps Personal Access Token \(PAT\)](#).
4. Configure defaults: We recommend you set the default configuration for your organization and project. Otherwise, you can set these within the individual commands themselves.

```
az devops configure --defaults organization=https://dev.azure.com/contoso project=ContosoWebApp
```

## Command usage

Adding the Azure DevOps Extension adds `devops`, `pipelines`, `artifacts`, `boards`, and `repos` groups. For usage and help content for any command, enter the `-h` parameter, for example:

```
$ az devops -h
```

```

Group
  az devops : Manage Azure DevOps organization level operations.
    Related Groups
      az pipelines: Manage Azure Pipelines
      az boards: Manage Azure Boards
      az repos: Manage Azure Repos
      az artifacts: Manage Azure Artifacts.

Subgroups:
  admin       : Manage administration operations.
  extension   : Manage extensions.
  project     : Manage team projects.
  security    : Manage security related operations.
  service-endpoint : Manage service endpoints/service connections.
  team        : Manage teams.
  user         : Manage users.
  wiki        : Manage wikis.

Commands:
  configure    : Configure the Azure DevOps CLI or view your configuration.
  feedback     : Displays information on how to provide feedback to the Azure DevOps CLI team.
  invoke       : This command will invoke request for any DevOps area and resource. Please use
                 only json output as the response of this command is not fixed. Helpful docs -
                 https://learn.microsoft.com/rest/api/azure/devops/.
  login        : Set the credential (PAT) to use for a particular organization.
  logout       : Clear the credential for all or a particular organization.

```

## Open items in browser

You can use `--open` switch to open any artifact in Azure DevOps portal in your default browser.

For example :

```
az pipelines build show --id 1 --open
```

This command shows the details of build with `id 1` on the command-line and also opens it in the default browser.

## Related articles

- [Sign in via Azure DevOps Personal Access Token \(PAT\)](#)
- [Output formats](#)
- [Index to az devops examples](#)
- [Azure DevOps CLI Extension GitHub Repo](#)

# Cross-service integration and collaboration overview

12/13/2022 • 16 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

One of the major strengths of Azure DevOps is the integration it supports across its core services. Azure DevOps supports multiple integration points across each of the major services—Azure Boards, Azure Repos, Azure Pipelines, and Azure Test Plans.

Review this article to understand how to use various features to support collaboration and traceability for all your devops tasks.

## Collaboration across Azure DevOps

Collaborating within and across teams is supported with many of the features summarized in the following table.

### Feature

### Description

**@mentions** (add to discussions and comments)

You can @mention a team member or an entire team within a work item form discussion or the comment section of a commit, pull request, or changeset. For details, see [Use @mentions in work items and pull requests](#).

**#ID** (link to a work item)

To support end-to-end traceability, you can link to work items from commits, pull requests, and changesets. For details, see [Link to work items from other objects](#).

### Teams

Each team gets access to a suite of Agile tools and team assets. These tools let teams work autonomously and collaborate with other teams across the enterprise. Each team can configure and customize each tool to support how they work. For quick navigation, they can favorite repositories, pipelines, and test plans. To learn more, see:

- [About teams and Agile tools](#)
- [Set personal or team favorites](#)
- [Unsubscribe from default notification](#)
- [Manage team, group, and Global notifications](#).

### Set up alerts

Configure or opt out of personal, team, project, or organization-level alerts. Subscribe to email alerts when changes occur to work items, code reviews, pull requests, source control files, builds and more. To learn more, see:

- [About notifications](#)
- [Manage personal notifications](#)
- [Unsubscribe from default notification](#)

- Manage team, group, and Global notifications.

---

Share summaries by email

- Email a list of work items
- Email query items
- Send release summaries by email

---

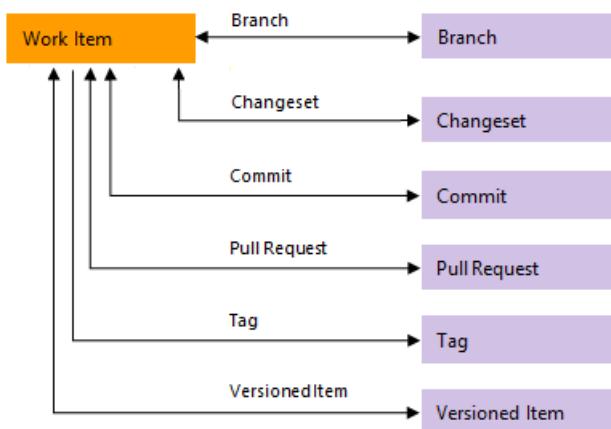
Wiki

Embed Azure Boards query results in Wiki.

---

## Azure Boards - Azure Repos

The following table summarizes the integration points between Azure Boards and Azure Repos. Through various link types, you can track code changes—commits and pull requests for Git, and changesets and versioned items for Team Foundation Version Control (TFVC)—that support development of user stories and features. The link types used to construct these links include *Branch*, *Commit*, *Pull Request*, and *Tag* for Git repositories, and *Changeset*, and *Versioned Item* for TFVC repositories. To learn more, see [Link to work items from other objects](#), [View list of linked objects](#).



### Feature

#### Description

---

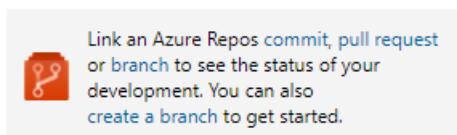
Drive Git development from work item(s)

You can initiate a Git branch or link to Git commits or pull requests and drive your Git development cycle for a work item from within the work item form.

#### Development

---

[+ Add link](#)



For details, see [Drive Git development from a work item](#).

---

Automatically link and transition work items with Git commits

You can enable or disable the following options for a single Git repository:

- Automatically create links for work items mentioned in a commit comment
- Allow mentions in commit comments to close work items
- Remember user preferences for completing work items with pull requests.

For details, see [Configure branch policies to support integration](#).

#### Check for linked work items in a Git branch

Encourage traceability by checking for linked work items on pull requests. For details, see [Configure branch policies to support integration](#).

#### Auto complete work items with pull requests

When you link a work item to a pull request (PR), you have the option to automatically complete those work items when you successfully complete the PR. The system defaults to your selection for future PRs. For details, see [Auto complete of work items with pull requests](#).

#### View list of code objects a single work item is linked to

You can link work items to code changes, builds, and releases—providing an audit trail of how a feature has been developed

## Azure Boards - Azure Pipelines

The following table summarizes the integration points between Azure Boards and Azure Pipelines. Several features provide support for end-to-end traceability as user stories and features move through the development cycle. As with Azure Repos, you can link work items to pipeline objects with the following link types: *Build*, *Integrated in build*, and *Integrated in release*.



#### Feature

#### Description

Manually link work items to builds.

Link work items to builds in the same or other project within the organization or collection.

Link work items to builds in the same project within the organization or collection.

Set integration option to automatically create *Integrated in build* links to work items linked to a branch, commit, or pull request associated with a pipeline.

Required to populate the **Development** control with *Integrated in build* links. The work items or commits that are part of a release are computed from the versions of artifacts. For example, each build in Azure Pipelines is associated with a set of work items and commits. For details, see [Configure pipelines to support integration](#).

## Development

---

 Add link

 Integrated in build [VSO.CI-Test\\_VSO.CI-Test\\_20210603.1](#)

6/3/2021,  partially succeeded

 Integrated in build [VSO.CI-FCSv1\\_VSO.CI-FCSv1\\_20210601.1](#)

6/2/2021,  partially succeeded

 [Status Service] Update IcM mapping metadata to include new...

Created 6/1/2021,  Completed

 [Status Service] Add Southeast Asia region to VSTSRegions.csl

Created 3/23/2021,  Completed

 8c84925c Merged PR 616007: [Status Service] Update IcM ma...

Created 6/1/2021

 d0de4229 Merged PR 616007: [Status Service] Update IcM ma...

Created 6/1/2021

[Show more](#) (6 of 8)

Not shown: Commit (2)

---

Set option and branch to automatically create *Integrated in build* and *Integrated in release stage* links to work items linked to a branch, commit, or pull request associated with a Classic or YAML pipeline.

Required to populate the work item form **Development** control with *Integrated in build* links and the **Deployment** control with *Integrated in release stage* links when running a Classic or YAML pipeline. For details, see [Configure pipelines to support integration](#).

---

Set integration option to automatically create *Integrated in release stage* links to work items linked to a branch, commit, or pull request associated with a release.

Required to populate **Deployment** control in work item form with *Integrated in release stage* links. For details, see [Release pipelines, How do I integrate and report release status?](#)

---

View list of work items linked to a Classic release pipeline

Lists all work items linked to a build or release.

---

View and open list of work items linked to a Classic or YAML pipeline.

Lists all work items linked to a release since the previous selected release. Can sort the list by each column.

---

View list of build or release objects a single work item is linked to

You can link work items to builds and releases—providing an audit trail of how a feature has been built and deployed. To learn more, see [Link to work items from other objects, View list of linked objects](#).

---

Query for external links.

You can query for work items that contain external links. For details, see [Query by link or attachment count](#)

---

View and quickly navigate to release stages a work item is linked to.

The work item form **Deployment** control lists set of stages work item is associated with. You can expand a stage to view status of select runs and quickly open each stage or run. For details, see [Link and view work items to builds and deployments](#).

## Deployment

### Testing

 My Release  
1h ago

### Staging

 My Release  
1h ago

### Production

 My Release  
59m ago

-  #4 → Production - East Coast
-  #4 → Production - West Coast
-  #4 → Production - Midwest

Create a work item on failure, optionally set values for a work item field (Classic)

Automatically create a work item and set fields when a build fails. For details, see [Build options](#).

Create a work item on failure (Classic or YAML), optionally set values for a work item field (Classic)

Automatically create a work item and set fields when a build fails. For details, see [Build options](#) for Classic pipelines, and [Customize pipelines, Create work item on failure](#).

Query Work Items task. Ensure the number of matching work items returned from a query is within a threshold.

Use this task to ensure the number of matching items returned by a work item query is within the configured thresholds. For details, see [Query Work Items task, Control deployments with gates and approvals](#).

## Azure Repos - Azure Pipelines

Azure Pipelines provides support for building code stored in Azure Repos, either a Git or Team Foundation Version Control (TFVC) repository. Other repositories that Azure Pipelines supports are listed in [Supported source repositories](#).

The following table summarizes the integration features between Azure Repos and Azure Pipelines.

### Feature

### Description

Report deployment status

Indicates the status of a deployment on the [Files](#), [Commits](#), and [Branches](#) pages for Git repositories. This feature improves the traceability from code commit to deployment. You can configure the release environments to report deployment status. For details, see [Release pipelines, How do I integrate and report release status?..](#)

Release status badge

Post the status of your most recent pipeline build in your repository. To learn how, see [Create your first pipeline](#), [Add a status badge to your repository](#).

## Code coverage

Publish and review code coverage results that indicate the proportion of your project's code that is actually being tested. To learn more, see [Publish Code Coverage Results task](#) and [Review code coverage results](#).

# Azure Boards - Azure Repos - Azure Test Plans

Several collaboration scenarios are supported through Azure Boards work item types. As with other work item types, you can use [managed queries](#) and the [Azure DevOps search function](#) to find and list work items.

### NOTE

Several of these work item types—such as Feedback Request, Code Review Request, Shared Steps, and Shared Parameters—are designed to be created through a specific tool or form. They aren't meant to be created manually. Therefore, they are added to the Hidden Types category. Work item types that are added to the Hidden Types category don't appear in the menus used to add work items.

Also, for the Inherited process model, you can only customize the following work item types: Test Plan, Test Suite, Test Case.

## Scenario

### Work item type

### Description

Request code review

### Code Review Request

Tracks information entered into the TFVC New Code Review form. To learn more, see [Get your code reviewed with Visual Studio](#).

Provide code review

### Code Review Response

Tracks review comments provided by code reviewers in response to a code review request. To learn more, see [Respond to the code review request](#).

Request feedback

### Feedback Request

Tracks information entered into a request feedback form. There are two forms that you can use to initiate a feedback request.

- [Request stakeholder feedback](#)
- [Get feedback](#).

Provide feedback

### Feedback Review

Enables stakeholders to provide feedback based on request for feedback or by volunteering feedback using the Microsoft Test & Feedback marketplace extension. To learn more, see the following articles:

- [Provide feedback](#)
- [Voluntarily provide stakeholder feedback](#)
- [Give feedback.](#)

---

Manual testing

## Test Plan

Groups one or more test suites and individual test cases together. Test plans include static test suites, requirement-based suites, and query-based suites. To get started, see [Create test plans and test suites](#).

---

Manual testing

## Test Suite

Groups one or more test cases into separate testing scenarios within a single test plan. Grouping test cases makes it easier to see which scenarios are complete. To learn more, see [Create test plans and test suites](#).

---

Manual testing

## Test Case

Defines steps used to validate individual parts of your code to ensure your code works correctly, has no errors, and meets business and customer requirements. You can add individual test cases to a test plan without creating a test suite. More than one test suite or test plan can refer to a test case. You can effectively reuse test cases without having to copy or clone them for each suite or plan. To learn more, see [Create manual test cases](#).

---

Manual testing

## Shared Steps

Enables sharing steps across several test cases. For details, see [Share steps between test cases](#).

---

Manual testing

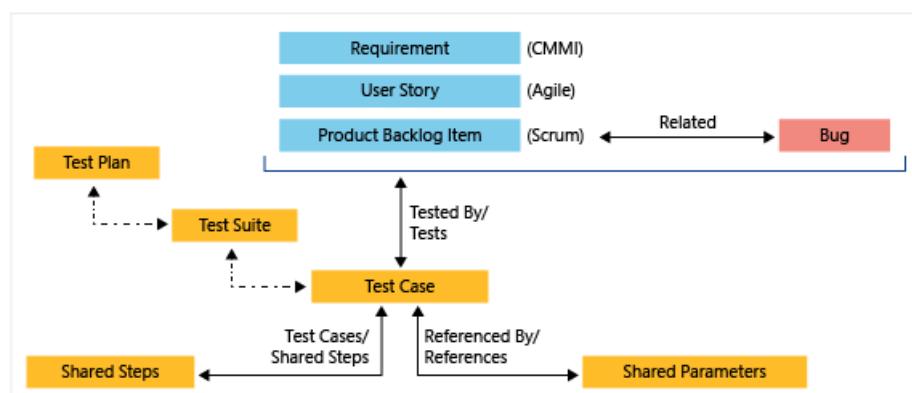
## Shared Parameters

Enables repeating the same test cases with different data. For more information, see [Repeat a test with different data](#).

---

## Test work item types

Work item types that support the test experience are linked together using the link types shown in the following image. These include *Tested By/Tests*, *Test Cases/Shared Steps*, and *Reference By/References*.



From the web portal, you can view which test cases are defined for a test suite, and which test suites are defined for a test plan. However, these objects aren't connected to each other through specific link types.

## Bug tracking

When tracking bugs using the Bug work item type, note the following supported integrations.

### Scenario

#### Description

Create a bug from a testing tool

You can add a bug from Test Runner or the Test & Feedback extension. To learn more, see [Define, capture, triage, and manage bugs](#).

Create inline tests linked to bugs or user stories

When your team tracks bugs as requirements, you can use the Kanban board to add tests to verify bug fixes or user stories. To learn more, see [Add, run, and update inline tests](#).

Track build information with bugs

The Bug work item form contains System Info, Found in Build, and Integrated in Build that support tracking code defects found and resolved within pipeline builds. To learn more, see [Query based on build and test integration fields](#).

## Azure Pipelines - Azure Test Plans

Azure Test Plans is fully integrated with Azure Pipelines to support testing within continuous integration/continuous deployment (CI/CD). Test plans and test cases can be associated with build or release pipelines. Pipeline tasks can be added to pipeline definitions to capture and publish test results. Test results can be reviewed via built in progress reports and pipeline test reports. The following table summarizes the integration points between Azure Pipelines and Azure Test Plans.

### Feature

#### Description

Test plans setting

With test plan settings, you configure the Test Run settings to associate build or release pipelines and Test Outcome settings. To learn more, see [Run automated tests from test plans](#)

Pipeline test-enable tasks

Specify test-enable tasks within a pipeline definition. Azure Pipelines provides several tasks, including those listed below, that support a comprehensive test reporting and analytics experience.

- [Publish Test Results task](#): Use to publish test results to Azure Pipelines.
- [Visual Studio Test task](#): Use to run unit and functional tests (Selenium, Appium, Coded UI test, and more) using the Visual Studio Test Runner.
- [.NET Core CLI task](#): Use to build, test, package, or publish a dotnet application.  
For additional tasks, see [Publish Test Results task](#)

Run automated tests in build pipelines

Associate test plans with a build pipeline so that they run with each build. To learn more, see [Run automated](#)

tests from test plans.

---

Associate automated tests with test cases

See [Associate automated tests with test cases](#)

---

Set retention policy for automated test results associated with builds

You can set the test retention policy for automated builds from the **Pipelines > Retention** page. See [Set test retention policies](#)

---

Requirements traceability

The Requirements quality widget supports tracking quality continuously from a build or release pipeline. The widget shows the mapping between a requirement and latest test results executed against that requirement. It provides insights into requirements traceability. To learn more, see [Requirements traceability](#).

---

Test results trend

The Test results trend configurable widget displays the trend of test results for the selected build or release pipeline. The widget helps you visualize the test trends over a period of time, thereby surfacing patterns about test failures, test duration etc. To learn more, see [Configure the Test Results Trend \(Advanced\) widget](#)

---

Deployment status

The Deployment status configurable widget shows a combined view of the deployment status and test pass rate across multiple environments for a recent set of builds. You configure the widget by specifying a build pipeline, branch, and linked release pipelines. To view the test summary across multiple environments in a release, the widget provides a matrix view of each environment and corresponding test pass rate. See [Associate automated tests with test cases](#)

---

View test results in builds and releases

Both build and release summaries provide details of test execution. Review these summaries to assess pipeline quality, review traceability, and troubleshoot failures. Choose **Test summary** to view the details in the **Tests** tab. To learn more, see [Review test results, Tests tab](#).

---

Test analytics for builds

Each build summary includes an **Analytics** tab that hosts the Test analytics report. To learn more, see [Test Analytics](#)

---

## Dashboards, reporting, and Analytics

Dashboards provide an easy way to monitor progress and status. Using widgets, teams can add configurable widgets to support their goals. To learn more, see [About dashboards, charts, reports, & widgets](#).

The Analytics service is the reporting platform for Azure DevOps, replacing the previous platform based on SQL Server Reporting Services. Built for reporting, Analytics is optimized for fast read-access and server-based aggregations. The Analytics service provides:

- Analytics widgets that you can add to your dashboards
- In-context Analytics reports available from select Azure DevOps pages
- Rollup bars and counts for Azure Boards backlogs
- Custom reports you can create using Power BI
- Custom reports you can create using OData queries

- Support to develop and add your custom Analytics widgets you can add to dashboards

To learn more, see [What is the Analytics service?](#)

## Dashboards and reporting

Dashboards provide an easy way to monitor progress and status. Using widgets, teams can add configurable widgets to support their goals. To learn more, see [About dashboards, charts, reports, & widgets](#).

SQL Server reports provide additional monitoring capabilities. To learn more, see [Reporting Services reports](#).

Built-in widgets you can add to your dashboard are listed below. They are organized under the service they support. You may find additional widgets from the [Azure DevOps Marketplace](#).

Widgets are annotated as follows:

- **Analytics**: Widget derives data from [Analytics data](#)
- **Build**: Widget derives data for a selected build pipeline
- **Project**: indicates you can select the project and team when configuring the widget
- **Release**: Widget derives data for a selected release pipeline
- **Team**: Widget is scoped to a single team
- **Teams**: Widget is scoped to one or more teams
- **User**: Widget is scoped to the logged in user account
  
- **Build**: Widget derives data for a selected build pipeline
- **Release**: Widget derives data for a selected release pipeline
- **Team**: Widget is scoped to a single team
- **User**: Widget is scoped to the logged in user account

---

### Boards

- [Assigned to me](#) (User)
- [Burndown chart](#) (Analytics, Project, Teams)
- [Burnup chart](#) (Analytics, Project, Teams)
- [Chart for work items](#)
- [Cumulative flow diagram](#) (Team)
- [Cycle time \(Analytics\)](#) (Analytics, Team)
- [Lead time \(Analytics\)](#) (Analytics, Team)
- [New Work item](#)
- [Query results](#)
- [Query tile](#)
- [Sprint burndown](#) (Analytics, Team)
- [Sprint burndown - Legacy](#) (Team)
- [Sprint capacity](#) (Team)
- [Sprint overview](#) (Team)
- [Velocity](#) (Analytics, Team)
- [Work links](#)

### Boards

- [Assigned to me](#) (User)
- [Burndown chart](#) (Analytics)
- [Burnup chart](#) (Analytics)

- [Chart for work items](#)
- [Cumulative flow diagram](#)
- [Cycle time \(Analytics\) \(Analytics\)](#)
- [Lead time \(Analytics\) \(Analytics\)](#)
- [New Work item](#)
- [Query results](#)
- [Query tile](#)
- [Sprint burndown](#)
- [Sprint capacity](#)
- [Sprint overview](#)
- [Velocity \(Analytics\)](#)
- [Work links](#)

## Work

- [Assigned to me \(User\)](#)
- [Chart for work items](#)
- [New Work item](#)
- [Query results](#)
- [Query tile](#)
- [Sprint burndown](#)
- [Sprint capacity](#)
- [Sprint overview](#)
- [Work links](#)

---

## Code

- [Code tile \(Repository, Branch, Folder\)](#)
- [Pull request \(Team, User\)](#)

## Code

- [Code tile \(Repository, Branch, Folder\)](#)
- [Pull request \(Team\)](#)

## Pipelines

- [Build history \(Build pipeline\)](#)
- [Deployment status \(Build pipeline\)](#)
- [Release pipeline overview \(Release pipeline\)](#)
- [Requirements quality \(Query, Build or Release pipeline\)](#)

---

## Test Plans

- [Chart for test plans](#)
- [Test results trend \(Advanced\) \(Analytics, Build or Release pipeline\)](#)
- [Test results trend \(Build or Release pipeline\)](#)

---

## Information and links

- [Embedded web page](#)
- [Markdown](#)

- [Other links](#)
- [Team members \(Team\)](#)
- [Visual Studio Shortcuts](#)
- [Welcome](#)

## Build & Release

- [Build history \(Build pipeline\)](#)
  - [Deployment status \(Build pipeline\)](#)
  - [Release pipeline overview \(Release pipeline\)](#)
  - [Requirements quality \(Query, Build or Release pipeline\)](#)
- 

## Test

- [Chart for test plans](#)
  - [Test results trend \(Build or Release pipeline\)](#)
- 

## Information and links

- [Embedded web page](#)
  - [Markdown](#)
  - [Other links](#)
  - [Team members \(Team\)](#)
  - [Visual Studio Shortcuts](#)
  - [Welcome](#)
- 

# Data available from Analytics

Analytics provides the reporting platform for Azure DevOps. Analytics is generally available for Azure DevOps Service and Azure DevOps Server 2020. It is in preview for Azure DevOps Server 2019.

You can access the following data from Analytics.

## Service

### Data availability

- Azure DevOps Services**
  - Azure DevOps Server 2020**
  - Azure DevOps Server 2019**
- 

## Boards

### [Widgets](#)

### [In-context reports](#)

### [OData Power BI](#)





---

## Repos

None

---

## Pipelines

[Test Analytics](#)

[Pipeline Analytics](#)

[OData Preview](#)



---

## Test Plans

[Progress Report](#)



---

## Artifacts

None

---

## Automation and Azure DevOps connectors

Several connectors are supported by Microsoft products to support automation or integration with other applications and services. You can learn more from the following resources.

- [Power Automate, Azure DevOps](#)
- [Power Automate templates for Azure DevOps](#)
- [Microsoft Power Automate documentation](#)

## Related articles

- [End-to-end traceability](#)
- [Data model for Analytics](#)

# Azure DevOps and GitHub integration overview

12/13/2022 • 6 minutes to read • [Edit Online](#)

## Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019

Azure Boards and Azure Pipelines provide several integration points with GitHub and GitHub Enterprise.

### Sign-in with GitHub credentials

Azure DevOps simplifies deployment from your repository with seamless access to the Azure portal and Azure DevOps using your GitHub account credentials.

Feature	Description
Invite GitHub collaborators into Azure DevOps	Provides support for inviting GitHub account users to collaborate within an Azure DevOps project. For details, see <a href="#">Invite GitHub collaborators into Azure DevOps (Release Notes)</a> .
Sign into Azure DevOps using your GitHub credentials	Allows users to sign in using their GitHub credentials and link their GitHub account to a Microsoft account. For details, see <a href="#">Signing into Azure DevOps using your GitHub credentials (Release Notes)</a> .
Connect to a GitHub repository from Visual Studio	Provides a user interface to support cloning GitHub repositories, pushing and pulling commits, and more. For details, see <a href="#">Side-by-side comparison of Git and Team Explorer</a> .

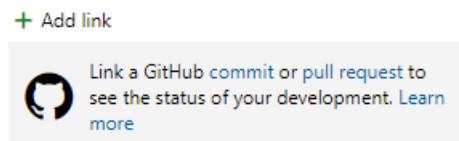
### Azure Boards and GitHub integration

By connecting Azure Boards with GitHub repositories, you enable linking between GitHub commits, pull requests, and issues to work items. You can use GitHub for software development while using Azure Boards to plan and track your work. To get started, see [Azure Boards-GitHub integration](#).

Feature	Description
Connect Azure Boards project to GitHub repos	Supports establishing connection of one or more GitHub repositories to an Azure Boards project. For details, see <a href="#">Azure Boards-GitHub integration</a> .
Connect Azure Boards project to repositories hosted in a GitHub Enterprise Server instance	Supports establishing connection of one or more GitHub repositories hosted in a GitHub Enterprise Server. For details, see <a href="#">Azure Boards-GitHub integration</a> .
Link work items to GitHub commits, pull requests, and issues. Quickly view and open linked objects from the Kanban board.	

Supports linking GitHub commits, pull requests, and issues to Azure Boards work items. Mentioned work items in GitHub comments are configured as hyperlinks to support quick navigation to Azure Boards work items.

## Development



For details, see [Link GitHub commits, pull requests, and issues to work items](#).

Add status badges of Azure Boards to a GitHub repository README file.

Supports adding Markdown syntax to a GitHub repo README.md file to display the status of a Kanban board.

For details, see [Configure status badges to add to GitHub README files](#).



Work items linked to GitHub commit in Release Summary

Review list of all work items linked to GitHub commits in the Release summary page. This helps teams track and retrieve more information about the commits that have been deployed to an environment.

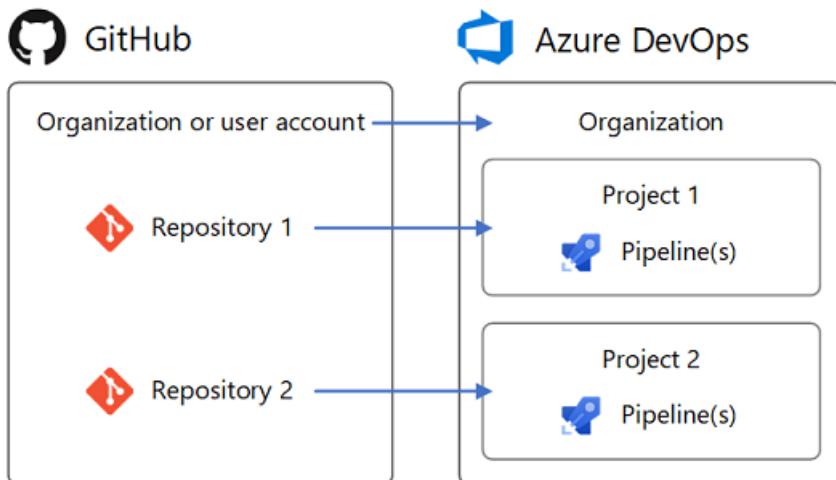
Sync GitHub Issues to Azure Boards Work Items

Using the [GitHub Action, GitHub Issues to Azure DevOps](#) you can sync your GitHub Issues to your Azure Boards. For details, see [Sync GitHub Issues to Azure DevOps Work Items \(Release Notes\)](#).

## Azure Pipelines and GitHub integration

You can use Azure Pipelines to automatically build, test, package, release, and deploy your GitHub repository code. To get started, see [Build GitHub repositories](#).

You can map your GitHub repositories to one or more projects in Azure DevOps.



### Feature

#### Description

GitHub repository and pull request builds

Automatically build pull requests from repository forks to ensure changes successfully build and tests pass

before they are merged. For details, see [Build GitHub repositories](#).

---

## GitHub repository and pull request builds

- Automatically build your GitHub pull requests. After the build is done, status is reported back with a comment in your GitHub pull request.
  - Manually run a pipeline or test suite triggered by a GitHub pull request comment.
  - Configure draft PR validation for GitHub repository. Supports adding `drafts` to the `pr` trigger YAML syntax for GitHub draft pull requests. You can choose if you want your draft PRs to queue a build. The default option is true (a build will be queued) like it currently is for GitHub PRs.
  - Rebuild GitHub pull request builds upon failure. Provides support for queueing a failed build.
  - Configure draft PR validation for GitHub repositories
  - Automatically build pull requests from repository forks to ensure changes successfully build and tests pass before they are merged. For details, see [Build GitHub repositories](#).
- 

## GitHub Enterprise builds

Supports continuous integration (CI) builds for GitHub Enterprise repositories. For details, see [Build GitHub repositories, CI triggers](#).

---

## GitHub Enterprise builds

- Supports continuous integration (CI) builds for GitHub Enterprise repositories.
  - Create a pipeline to build code contained within a GitHub Enterprise repository using the the build pipeline wizard. For details, see [Build GitHub repositories, CI triggers](#).
- 

## GitHub service connections

The pipeline wizard automatically creates and reuses a service connection for the repository you choose. If you wish to manually choose a connection other than the one that is automatically selected, follow the **Choose connection** hyperlink. For details, see [Build GitHub repositories](#).

---

## GitHub-specific tasks and utilities

Supported:

- [Download GitHub Release task](#)
  - [GitHub Release task](#)
  - [Open source Azure Pipeline tasks](#)
- 

## Manage GitHub releases

- Inline GitHub connection as a release artifact source.
  - Automate GitHub releases using the **GitHub Release** task.  
For details, see:
    - [CI triggers](#)
    - [Download GitHub Release task](#)
- 

## Manage GitHub releases

- Inline GitHub connection as a release artifact source.
- Automate GitHub releases using the **GitHub Release** task.
- Link your GitHub releases as an artifact source in release pipelines. This function lets you consume the GitHub release as part of your deployments.

For details, see:

- [CI triggers](#)
  - [Download GitHub Release task](#)
  - [GitHub Release task](#)
- 

Filter GitHub branches for GitHub, GitHub Enterprise, or external Git artifacts

When releasing from GitHub, GitHub Enterprise, or external Git repositories, you can configure the specific branches to release. For example, you may want to deploy only builds coming from a specific branch to production. For details, see [Release triggers](#), [Continuous deployment triggers](#).

---

GitHub Actions to trigger a pipeline run

automate your software development workflows from within GitHub. You can deploy workflows in the same place where you store code and collaborate on pull requests and issues. For details, see [Quickstart: Trigger an Azure Pipelines run from GitHub Actions](#).

---

Use build tags to trace GitHub sources

Use build tags to trace GitHub sources to builds. While choosing a GitHub repository in a build definition, you can select the types of builds you want to tag, along with the tag format. For details, see [Build GitHub repositories](#), [Label sources](#).

---

Use build tags to trace GitHub sources or trigger GitHub releases

Use build tags to trace GitHub sources to builds. While choosing a GitHub repository in a build definition, you can select the types of builds you want to tag, along with the tag format.

- Use build tags to trace GitHub sources to builds. While choosing a GitHub repository in a build definition, you can select the types of builds you want to tag, along with the tag format.
- Specify a tag pattern to determine when to trigger a GitHub release. By specifying a tag regular expression, you can control when a GitHub release is created based on the triggering commit.

For details, see [Build GitHub repositories](#), [Label sources](#).

---

GitHub packages support in YAML pipelines

In your YAML pipeline, specify a package type (NuGet or npm) that you want to consume from GitHub. For details, see [Resources: packages](#).

---

Status checks, tracking, and traceability

- **GitHub Checks:** Display status for each pipeline job: Run a pipeline or test suite to validate a GitHub pull request from the comments section of the GitHub pull request.
- **GitHub Checks** allows for sending detailed information about the pipeline status, test, code coverage, and errors. Status is posted to GitHub Checks for each job in the pipeline.
- **Status badges:** Supports adding Markdown syntax to a GitHub repo README.md file to display the pipeline status.
- GitHub artifacts show associated commits deployed in a release. To enhance traceability, you can see all the commits that were deployed to an environment for GitHub repositories, as a part of a specific release.
- Track GitHub commits and associated issues in releases. Lists commits made in GitHub repos and the associated GitHub issues that are being deployed with a release. For details, see [Track GitHub commits and associated issues in releases \(Release Notes\)](#).

For details, see:

- [Create your first pipeline](#), Add a status badge to your repository.
  - [GitHub Checks API](#)
  - [Display status for each pipeline job in GitHub Checks \(Release Notes\)](#).
- 

## Related articles

- [Azure Boards-GitHub integration](#)
- [Build GitHub repositories](#)
- [Git experience in Visual Studio](#)

# Deploy to Azure

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to test and build your code and ship it to any target. While you do not have to use Azure services with Pipelines, Pipelines can help you take advantage of Azure. You can use Pipelines to integrate your CI/CD process with most Azure services.

To learn more about selecting an Azure service for hosting your application code, see [Choose an Azure compute service for your application](#).

If you're just getting started, we recommend you review and get started with the following resources.

## Azure service

### Integration points

#### [Azure DevOps Projects/Azure DevOps Starter](#)

Start using Azure Pipelines to automate the setup of a CI/CD of your application to Azure. Choose where to deploy your application such as Virtual Machines, Azure App Service, Azure Kubernetes Services (AKS), Azure SQL Database, or Azure Service Fabric.

To learn more, see [Overview of DevOps Starter](#).

## Azure portal

The Azure portal is a web-based, unified console from which you can build, manage, and monitor everything from simple web apps to complex cloud deployments. Also, you can create custom dashboards for an organized view of resources and configure accessibility options. If you have an Azure DevOps Services organization, you have access to the Azure portal.

[Sign in to your Azure portal](#).

## DevOps solutions on Azure

Use end-to-end solutions on Azure to implement DevOps practices throughout application planning, development, delivery, and operations. Apply the right combination of DevOps technologies, culture, and processes to enable continual software delivery and better value for customers. Get started with the following Learn modules:

- [Deploy applications with Azure DevOps](#)
- [Build applications with Azure DevOps](#)
- [Deploy and maintain cloud-native apps with GitHub actions and Azure Pipelines](#)
- [Load test Azure web apps by using Azure DevOps](#)

Follow the links provided in the following table to learn more about the Azure services that support continuous integration (CI) and continuous delivery (CD) using Azure Pipelines. For a complete list of Azure pipeline tasks, see [Build and release tasks](#).

## Azure service

### Integration points

## Azure App Service

An HTTP-based service for hosting web applications, REST APIs, and mobile back ends; the Azure App Service employs Azure Pipelines to deliver CI/CD. To learn more, see:

- [App Service overview](#)
  - [Deploy an Azure Web App](#)
  - [Use CI/CD to deploy a Python web app to Azure App Service on Linux](#)
  - [Continuously deploy from a Jenkins build](#)
  - [Azure App Service Deploy task](#)
  - [Azure App Service Manage task](#)
  - [Azure App Service Settings task](#)
- 

## Azure App Configuration

Service to centrally manage application settings and feature flags. To learn more, see the following articles:

- [Push settings to App Configuration with Azure Pipelines](#)
  - [Pull settings to App Configuration with Azure Pipelines.](#)
- 

## Azure Blob Storage [Azure Storage](#)

Store and access unstructured data at scale using Azure Pipelines and Azure Blob Storage.

---

## Azure Static Web Apps

Use Azure Static Web Apps to automatically build and deploy a full stack web app to Azure from a code repository.

- [Tutorial: Publish Azure Static Web Apps with Azure DevOps](#)
- 

## Azure Container Registry

Build, store, secure, scan, replicate, and manage container images and artifacts. For example, build and publish a private Docker registry service. To learn more, see [Build and push Docker images to Azure Container Registry](#).

---

## Azure Databases

[Azure SQL Database](#)

[Azure Database for MySQL](#)

[Azure Cosmos DB](#)

Use Azure Pipelines to deploy to Azure SQL Database, Azure Database for MySQL, or Azure Cosmos DB. To learn more, see the following articles:

- [Deploy to Azure SQL Database](#)
  - [Azure SQL Database Deployment task](#)
  - [Azure Database for MySQL Deployment task](#)
  - [Quickstart: Deploy to Azure MySQL](#)
  - [Set up a CI/CD pipeline with the Azure Cosmos DB Emulator build task in Azure DevOps](#)
- 

## Azure Databricks

[Azure Data Factory](#) [Azure Machine Learning](#)

Configure a pipeline to integrate with a fully managed, serverless data integration service and unlock insights from all your data. Create an Azure Pipeline that builds and deploys a machine learning model as a web service and to automate the machine learning lifecycle. To learn more, see the following resources:

- Build a data pipeline by using Azure Data Factory, DevOps, and machine learning; Configure Azure Databricks and Azure Data Factory
  - DevOps for Azure Databricks
  - Prepare data, train, deploy, and monitor machine learning models with Azure Pipelines.
- 

## Azure DevTest Labs

Quickly provision development and test stages using reusable templates. To learn more, see [Manage a virtual machine in Azure DevTest Labs](#).

---

## Azure Functions

Provides a fully managed Platform as a service (PaaS) to implement serverless architecture. To learn more, see:

- [Deploy an Azure Function](#)
  - [Azure Function App task](#)
  - [Azure Function App for Containers task](#)
- 

## Azure Government

Use Azure Pipelines to set up CI/CD of your web app running in Azure Government. To learn more, see [Deploy an app in Azure Government with Azure Pipelines](#).

---

## Azure IoT Edge

Use Azure Pipelines to manage services built on Azure IoT Hub. To learn more, see [Continuous integration and continuous deployment to Azure IoT Edge devices](#) and [Create a CI/CD pipeline for IoT Edge with Azure DevOps Starter](#).

---

## Azure Key Vault

Use Azure Pipelines to manage services for storing secret data. To learn more, see [Use Azure Key Vault secrets in Azure Pipelines](#) and [Azure Key Vault task](#).

---

## Azure Kubernetes Services (AKS)

Deploy and manage containerized applications with a fully managed Kubernetes service. To learn more, see [Build and deploy to Azure Kubernetes Service](#).

---

## Azure Monitor

Configure alerts on available metrics for an Azure resource. Observe the configured Azure monitor rules for active alerts in a release pipeline. Define pre or post-deployment gates based on Query Azure Monitor Alerts. For details, see the following articles:

- [Define approvals and checks, Query Azure Monitor Alerts](#)
  - [Release deployment control using gates](#)
  - [Azure Monitor Alerts task](#)
  - [Query Azure Monitor Alerts task](#).
- 

## Azure Policy

Manage and prevent IT issues by using policy definitions that enforce rules and effects for your resources. To learn how, see [Check policy compliance with gates](#).

---

## Azure Resource Manager (ARM)

## ARM Templates

Use ARM templates to define the infrastructure and dependencies and streamline authentication to deploy your app using Azure Pipelines. Specifically, you can:

- Create an ARM service connection using automated security
- Create an ARM service connection with an existing service principal
- Create an ARM service connection to a VM with a managed service identity
- Connect to an Azure Government Cloud
- Connect to Azure Stack

To learn more, see [Connect to Microsoft Azure](#).

---

## Azure Service Bus

In a release pipeline, send a message to an Azure Service Bus using a service connection. To learn more, see [Publish To Azure Service Bus task](#) and [Manage service connections, Azure Service Bus service connection](#).

---

## Azure Service Fabric

Distributed systems platform that can run in many environments, including Azure or on-premises. To learn more, see the following articles: [Tutorial: Deploy an application with CI/CD to a Service Fabric cluster](#) and [Service Fabric Application Deployment task](#).

---

## Azure Stack

Build, deploy, and run hybrid and edge computing apps consistently across your ecosystems. To learn more, see [Deploy to Azure Stack Hub App Service using Azure Pipelines](#).

---

## Azure Virtual Machines

### Azure Virtual Machine Scale Sets

Simplify continuous delivery to Azure VMs using Azure Pipelines. To learn more, see these articles:

- [Build an Azure virtual machine using an Azure RM template](#)
  - [Deploy to Azure VMs using deployment groups in Azure Pipelines](#)
  - [Tutorial: Deploy a Java app to a virtual machine scale set](#)
- 

## Azure WebApps

Use publish profile to deploy Azure WebApps for Windows from the Deployment Center. To learn more, see the following articles:

- [Deploy an Azure Web App](#)
  - [Deploy an Azure Web App Container](#)
  - [Azure App Service Deploy task](#)
  - [Azure App Service Manage task](#)
  - [Azure App Service Settings task](#)
-

# Web portal navigation in Azure DevOps

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The web portal for Azure DevOps is organized around a set of services as well as administrative pages and several task specific features such as the search box. The service labels differ depending on whether you work from Azure DevOps Services or Azure DevOps on premises and its version.

## IMPORTANT

### Version

- Azure DevOps Services
- Azure DevOps Services
- Azure DevOps Server 2022
- Azure DevOps Server 2020
- Azure DevOps Server 2019
- TFS 2018

Previous versions

To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.

To learn which on-premises version you are using, see [Look up your Azure DevOps platform and version](#)

Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

The web portal for Azure DevOps Server is organized around a set of services—such as, **Overview**, **Boards**, **Repos**, **Pipelines**, **Test Plans**, and **Artifacts**—as well as administrative pages and several task-specific features such as the search box. Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

The web portal for Team Foundation Server is organized around a set of applications—such as, **Dashboards**, **Code**, **Work**, **Build and Release**—as well as administrative pages and several task-specific features such as the search box. Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

Here's what you need to know to get up and running using the web portal.

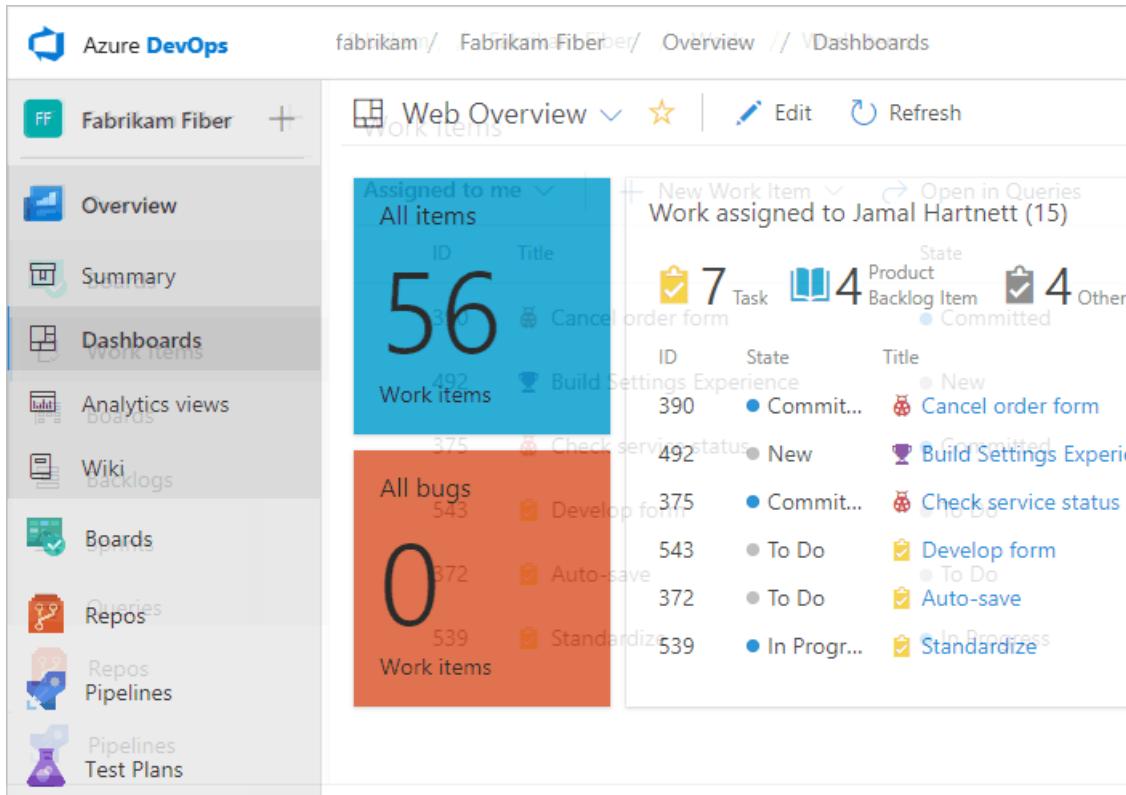
- **Open a service, page, or settings:** use to switch to a different service or functional area
- **Add an artifact or team:** use to quickly add a work item, Git repo, build or release pipelines, or a new team

- **Open another project or repo:** use to switch to a different project or access work items and pull requests defined in different projects, or items you've favorited
- **Open team artifacts, use breadcrumbs, selectors and directories:** use to navigate within a service, to open other artifacts, or return to a root function
- **Work with favorites:** favorite artifacts to support quick navigation
- **Search box:** use to find code, work items, or wiki content
- **Your profile menu:** use to set personal preferences, notifications, and enable preview features
- **Settings:** use to add teams, manage security, and configure other project and organization level resources.
  
- **Open a service, page, or settings:** use to switch to a different service or functional area
- **Add an artifact or team:** use to quickly add a work item, Git repo, build or release pipelines, or a new team
- **Open another project or repo, or switch to a different team:** use to switch to a different project or browse teams
- **Work across projects:** use to quickly open work assigned to you, your active pull requests, or items you've favorited
- **Open team artifacts, use breadcrumbs & selectors:** use to navigate within a service, to open other artifacts or return to a root function
- **Work with favorites:** favorite artifacts to support quick navigation
- **Search box:** use to find code, work items, or wiki content
- **Your profile menu:** use to set personal preferences, notifications, and enable preview features
- **Settings:** use to add teams, manage security, and configure other project and organization-level resources.

#### NOTE

Only those services that are enabled will appear in the user interface. For example, if **Boards** is disabled, then **Boards** or **Work** and all pages associated with that service won't appear. To enable or disable a service, see [Turn an Azure DevOps service on or off](#).

You select services —such as **Boards**, **Repos**, and **Pipelines**— from the sidebar and pages within those services.



The screenshot shows the Azure DevOps Web Overview dashboard for the 'Fabrikam Fiber' project. The left sidebar lists various services: Boards, Queries, Repos, Pipelines, and Test Plans. The main area displays two cards: 'Assigned to me' (56 work items) and 'All bugs' (0 work items). A summary bar on the right shows 7 tasks, 4 backlog items, and 4 other items. The top navigation bar includes links for 'Overview', 'Work items', 'Edit', and 'Refresh'.

Category	Count	Type	State	Title
Task	7	Task	New	Cancel order form
Backlog Item	4	Product Backlog Item	Committed	Build Settings Experience
Other	4	Other	Committed	Check service status
			Commit...	Develop form
			To Do	Auto-save
			To Do	In Progress
			In Progress	Standardize

You select a service—such as **Code**, **Work**, and **Build and Release**—from the horizontal bar and pages within those services.

The screenshot shows the Azure DevOps Web Overview page for the 'Fabrikam Fiber / Fabrika...' project. The top navigation bar includes 'Dashboards', 'Code', 'Work', 'Build and release', 'Test', and 'Wiki'. Below the navigation is a toolbar with 'Web Overview' (dropdown), a star icon, 'Edit', and 'Refresh' buttons. On the left, there are two summary cards: 'All items' (56 Work items) and 'All bugs' (0 Work items). The main content area displays a list titled 'Work assigned to Jamal Hartnett (15)' with the following details:

Icon	Count	Type
Task	7	Product Backlog Item
Product Backlog Item	4	Other

ID	State	Title
390	● Commit...	Cancel order form
492	● New	Build Settings Experience
375	● Commit...	Check service status
543	● To Do	Develop form
372	● To Do	Auto-save
539	● In Progr...	Standardize

Now that you have an understanding of how the user interface is structured, it's time to get started using it. As you can see, there are a lot of features and functionality.

If all you need is a code repository and bug tracking solution, then start with [Get started with Git](#) and [Manage bugs](#).

To start planning and tracking work, see [About Agile tools](#).

## Connect to the web portal, user accounts, and licensing

You connect to the web portal through a supported web browser—such as the latest versions of Edge, Chrome, Safari, or Firefox. Only users who have been [added to a project](#) can connect. This is typically done by the organization owner.

Five account users are free as are Visual Studio subscribers and stakeholders. After that, you need to [pay for more users](#). Find out more about licensing from [Azure DevOps pricing](#).

Limited access is available to an unlimited number of stakeholders for free. For details, see [Work as a Stakeholder](#).

You connect to the web portal through a supported web browser—such as the latest versions of Edge, Chrome, Safari, or Firefox. Only users who have been [added to a project](#) can connect. This is typically done by a member of the Project Administrators group.

Limited access is available to an unlimited number of stakeholders for free. For details, see [Work as a Stakeholder](#). Most regular contributors must have a TFS client access license (CAL). All Visual Studio subscriptions include a TFS CAL. Find out more about licensing from [TFS pricing](#).

## Refresh the web portal

If data doesn't appear as expected, the first thing to try is to refresh your web browser. Refreshing your client

updates the local cache with changes that were made in another client or the server. To refresh the page or object you're currently viewing, refresh the page or choose the  Refresh icon if available.

To avoid potential errors, you should refresh your client application under the following circumstances:

- Process changes are made
- Work item type definitions are added, removed, renamed, or updated
- Area or iteration paths are added, removed, renamed, or updated
- Users are added to or removed from security groups or permissions are updated
- A team member adds a new shared query or changes the name of a shared query
- A build definition is added or deleted
- A team or project is added or deleted

## Differences between the web portal and Visual Studio

Although you can access source code, work items, and builds from both clients, some task specific tools are only supported in the web browser or an IDE but not in both. Supported tasks differ depending on whether you connect to a Git or TFVC repository from Team Explorer.

---

### Web portal

### Visual Studio

---

- [Product backlog](#), [Portfolio backlogs](#), [Sprint backlogs](#), [Taskboards](#), [Capacity planning](#)
- [Kanban boards](#)
- [Dashboards](#), [Widgets](#), [Charts](#)
- [Request feedback](#)
- [Web-based Test Management](#)
- [Administration pages to administer accounts, team projects, and teams](#)
- Git: [Changes](#), [Branches](#), [Pull Requests](#), [Sync](#), [Work Items](#), [Builds](#)
- TFVC: [My Work](#), [Pending Changes](#) | [Source Control Explorer](#), [Work Items](#) | [Builds](#)
- Greater integration with work items and Office integration clients. You can open a work item or query result in an office supported client.

---

#### NOTE

Visual Studio 2019 version 16.8 and later versions provide a new Git menu for managing the Git workflow with less context switching than Team Explorer. Procedures provided in this article under the Visual Studio tab provide information for using the Git experience as well as Team Explorer. To learn more, see [Side-by-side comparison of Git and Team Explorer](#).

## Resources

- [Manage projects](#)
- [Project & Organizational Settings](#)

# Open a service, page, or settings

12/13/2022 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The web portal for Azure DevOps provides support for software development teams to collaborate through the planning, development, and release cycles. You can manage source code, plan and track work, define builds, run tests, and manage releases.

This article shows you how to navigate to functional and administrative tasks available from the web portal. There are three levels of administrative tasks: team, project, and organization.

If you don't have a project yet, [create one](#). If you don't have access to the project, [get invited to the team](#).

This article shows you how to navigate to functional and administrative tasks available from the web portal. There are four levels of administrative tasks: team, project, collection, and server.

If you don't have a project yet, [create one](#). If you don't have access to the project, [get invited to the team](#).

## Open a service or functional task page

Services support getting work done—managing code, planning and tracking work, defining and managing pipelines, creating and running tests, and so on.

### NOTE

Only those services that are enabled will appear in the user interface. For example, if **Boards** is disabled, then **Boards** or **Work** and all pages associated with that service won't appear. To enable or disable a service, see [Turn an Azure DevOps service on or off](#).

You open a service by choosing the service from the sidebar and then selecting from the available pages.

For example, here we select **Boards > Backlogs**.



Order	Work item Type	Title
1	Product Backlog	> Hello World Web Site
2	Bug	> Slow response on information form
3	Product Backlog	> Change initial view
4	Product Backlog	> Interim save on long form
5	Bug	> Canadian addresses don't display correctly
6	Product Backlog	> Hello World Web Site
7	Product Backlog	> GSP locator interface
8	Product Backlog	> Request support

Within the page you may select a specific view or artifact, such as a team backlog or choose another page.

You open a service by choosing it from the horizontal blue bar. Then, select from the available pages.

For example, here we select **Work>Work Items**.

Fabrikam

Work

Work Items\*

## Open team settings

Select configurations are made to teams through the team settings pages. For an overview of all team settings, see [About user, team, project, and organization-level settings](#).

1. Choose **Project Settings**.

Summary - Overview   X

← → C https://dev.azure.com/fabrikam/FabrikamFiber

Azure DevOps   Fabrikam / FabrikamFiber / Overview / Summary

FabrikamFiber   Private   Invite

Overview   Summary   Dashboards   Wiki   Boards   Repos   Pipelines   Test Plans   Artifacts

Project settings

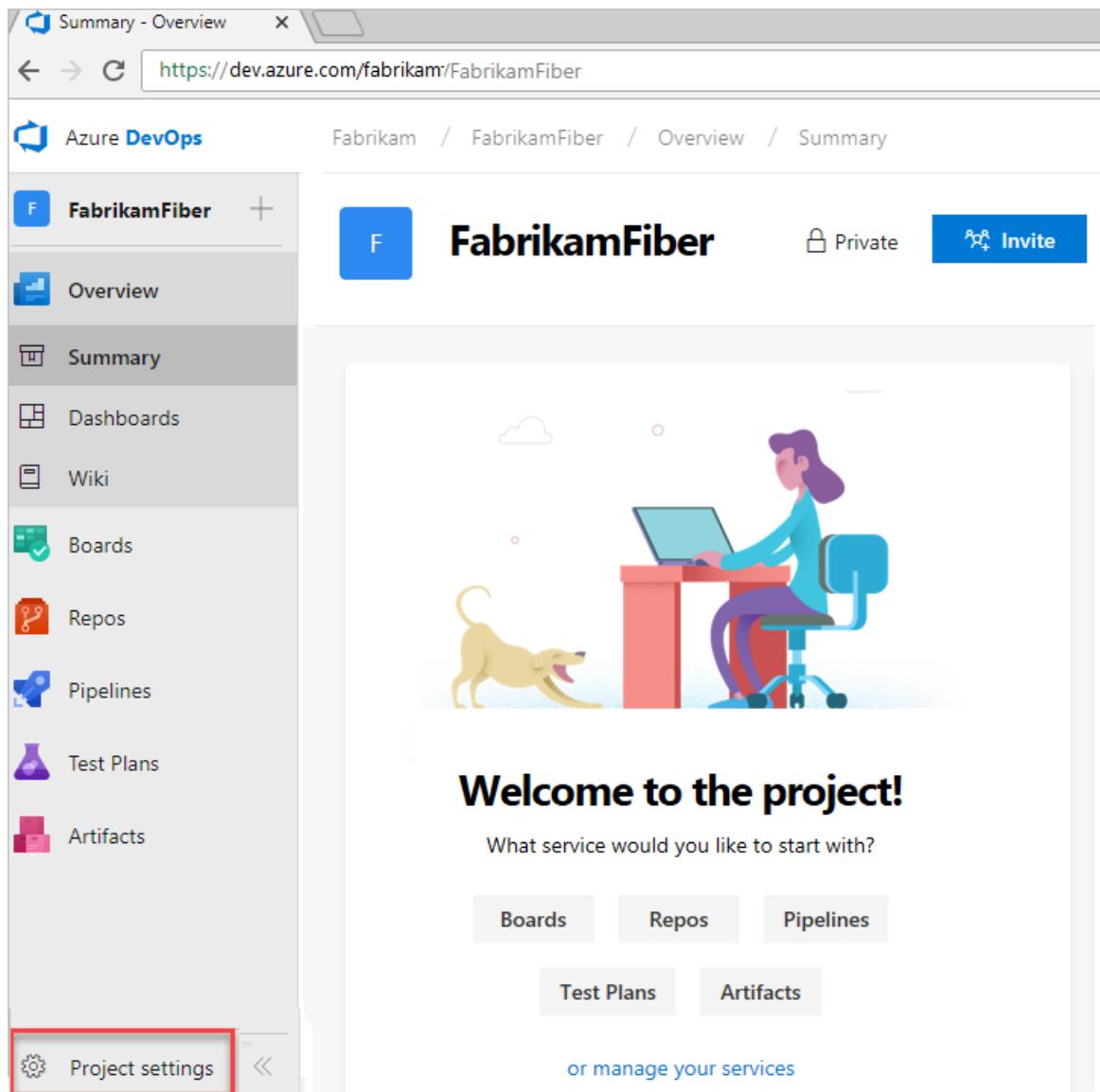


Welcome to the project!

What service would you like to start with?

Boards   Repos   Pipelines   Test Plans   Artifacts

or manage your services



2. Expand Boards and choose Team configuration.

**General**

Overview

Services

Teams

Security

Notifications

Service hooks

Dashboards

**Boards**

Project configuration

**Team configuration**

> Build and release

> Code

**Backlogs**

See only the backlogs your team manages.

**Backlog navigation levels**

- Epics
- Features
- Backlog items

**Working days**

Capacity and burndown are based on the days your team works.

**Select days**

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

3. Choose one of the pages **General**, **Iterations**, **Areas**, or **Templates** to configure settings for the team.  
To learn more, see [Manage teams](#).

4. If you need to switch to a different team, use the team selector within the breadcrumbs.

/ Settings / Teams / **Fabrikam Fiber Team** ▾

**Team Profile**

Name: Fabrikam Fiber Team

Description: The default project team.

Display Name	Username Or Scope
Jamal Hartnett	fabrikamfiber4@hotmail.com

+ Add... |

**Fabrikam Fiber Team (Fabrikam F...)**

Customer Service (Fabrikam Fiber)

5. To add a team administrator, add team members, or change the team profile, choose **Teams** from the vertical sidebar, and then choose the name of the team you want to configure.

You open team settings from the top navigation bar. Select the team you want and then choose the gear icon. To learn more about switching your team focus, see [Switch project, repository, team](#).

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. At the top, there's a navigation bar with links for Overview, Work (which is selected), Security, Version Control, Policies, Agent Queues, Notifications, and Service Hooks. Below this is a secondary navigation bar with General, Iterations, Areas, and Templates. The main content area is titled 'Work' and contains tabs for General, Iterations, Areas, and Templates. A large red box highlights the 'General' tab. To the right of the tabs, there's a gear icon.

1. Choose one of the pages **General**, **Iterations**, **Areas**, or **Templates** to configure settings for the team.  
To learn more, see [Manage teams](#).

2. To add a team administrator, add team members, or change the team profile, choose **Overview**.

3.

## Open project settings

Administrators configure resources for a project and manage project-level permissions from the **Project settings** pages. Tasks performed in this context can impact the project and team functions. For an overview of all project settings, see [Project administrator role and managing projects](#).

1. Choose **Project Settings**.

2. From there, you can choose a page from the list. Settings are organized based on the service they

support. Expand or collapse the major sections such as **Boards**, **Build and release**, **Code**, **Test**, and **Extensions** to select from the list.

Project Settings

General

- Overview
- Services
- Teams
- Security
- Notifications
- Service hooks
- Dashboards

> Boards

> Build and release

> Code

> Test

> Extensions

Project details

Changes made here will affect all members and URLs associated with this project.

FF

Name

Fabrikam Fiber [Rename](#)

Description

Web, voice, and phone apps

Visibility

Private [Edit](#)

Process

MyScrum

From a user context, open **Project settings** by choosing the gear icon.

Open any admin page by choosing its name. Choose or hover over the gear icon to access other administrative options. Note that you can choose any of the user-context areas—**Dashboards**, **Code**, **Work**—to return to the user context.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. At the top, there's a navigation bar with 'Dashboards', 'Code', 'Work', 'Build and release', and a gear icon. Below that is a secondary navigation bar with 'Fabrikam Fiber', 'Files' (which is selected), 'Commits', 'Pushes', 'Branches', 'Tags', and 'Pull'. On the left, there's a sidebar for 'Fabrikam Fiber' with files like 'page-1.md', 'page-2.md', 'page-3.md', and 'README.md'. The main area shows a table with columns for 'Name', 'Last change', and 'Type'. A dropdown menu is open from the gear icon, listing options like 'Version Control', 'Policies', 'Agent Queues', etc., with 'Project settings' highlighted by a red box and an arrow pointing to it.

## Open Organization settings

Organization owners and members of the Project Collection Administrators group configure resources for all projects or the entire organization, including adding users, from the Organization settings pages. This includes managing permissions at the organization-level. For an overview of all organization settings, see [Project collection administrator role and managing collections of projects](#).

## Open Collection settings

Members of the Project Collection Administrators group configure resources for all projects or the entire project collection from the Collection settings pages. This includes managing permissions at the collection-level. For an overview of all collection-level settings, see [Project collection administrator role and managing collections of projects](#).

1. Choose the Azure DevOps logo to open Projects. Then choose Admin settings.

The screenshot shows the Azure DevOps interface. On the left, there's a sidebar with organization navigation. The main area displays the 'FabrikamFiber01' organization, featuring a project named 'Fabrikam Fiber'. A 'What's new' section highlights the 'Sprint 162 release notes'. At the bottom of the sidebar, the 'Organization settings' link is highlighted with a red box.

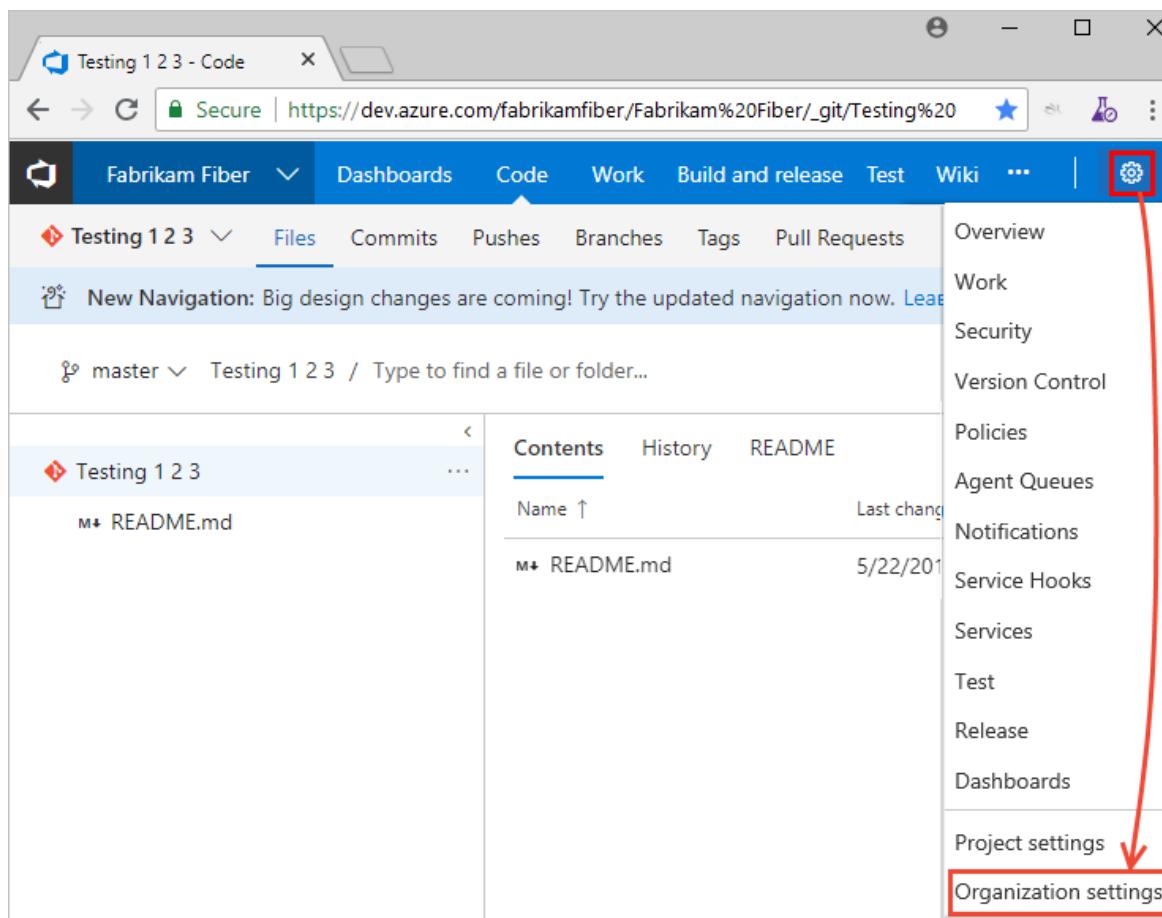
- From there, you can choose a page from the list of settings. Settings are organized based on the service they support. Expand or collapse the major sections such as **Boards** and **Build and release** to select a page from the list.

## Organization Settings > Projects

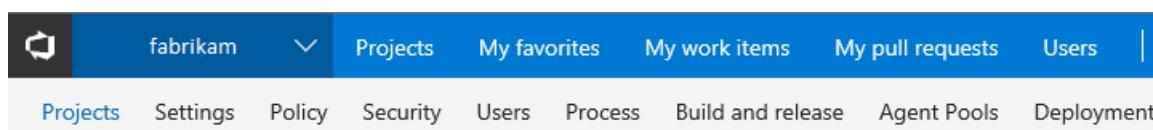
The screenshot shows the 'Organization Settings > Projects' page. The left sidebar has sections like General, Overview, Projects (which is selected and highlighted in blue), Policy, Users, Security, Notifications, Extensions, and Usage. The main area shows a table of projects:

Project name	Process	Status	Description
Agile 11	MyAgile 2	Online	New agile project
Fabrikam Fiber	MyScrum	Online	Web, voice, and phone apps
Fabrikam Test	MyAgile Test	Online	MyAgile process customizations
Test Agile Repo	MyAgile	Online	
Visual Studio Code	Agile	Online	

1. Choose the gear icon to open Organization settings or Collection settings.



2. From there, you can choose a page. Settings are organized based on the service they support.



## Open Server settings

Members of the Team Foundation Server Administrators group configure resources for the server instance from the Server settings pages.

1. From the web portal home page for a project, choose or hover over the gear icon and select Server settings.

The screenshot shows the Microsoft Team Foundation Server (TFS) interface. At the top, there's a navigation bar with links for Overview, Plan, Home, Code, Work, and more. Below the navigation is a main content area with a 'Welcome' section containing four cards: 'Manage Work' (with a 'Add work to your board' link), 'Collaborate on code' (with a 'Add code to your repository' link), 'Continuously integrate' (with a 'Automate your builds' link), and 'Visualize progress' (with a 'Learn how to add charts' link). To the right is a sidebar with a 'Work' category expanded, showing 'Backlog', 'Board', 'Task board', and 'Queries'. Below the 'Work' category, there are other sections: 'Visual Studio' (with 'Open in Visual Studio' and 'See Visual Studio' links), 'Security' (with 'Alerts', 'Requires Visual Studio', and 'Version Control'), 'Agent Queues', 'Service Hooks', 'Services', 'Test', 'Release', 'Default team settings', 'Collection settings', and 'Server settings' (which is highlighted with an orange border).

2. Choose **Access levels**, to set access levels for a member or group. For details, see [Change access levels](#).

If you don't see **Access levels**, you aren't a TFS administrator and don't have permission. [Here's how to get permissions](#).

## Related articles

- [Manage projects](#)
- [About team, project, and admin settings](#)

# Add an artifact or team artifacts

12/13/2022 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

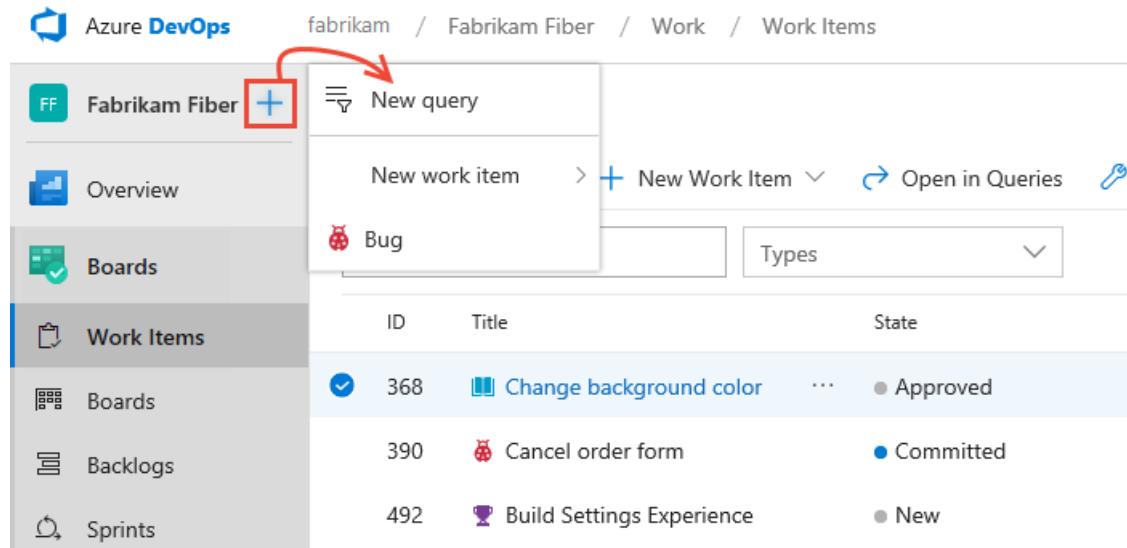
Select the service of interest to get started adding new artifacts or objects. For example, to add work items, choose **Boards** or **Work**. Some artifacts—such as a product backlog, Kanban board, portfolio backlogs—are added when you add a team.

Prior to adding an artifact, make sure that you've [selected the project and repository](#) that you want to work in.

## Add work items, queries, or other work tracking artifacts

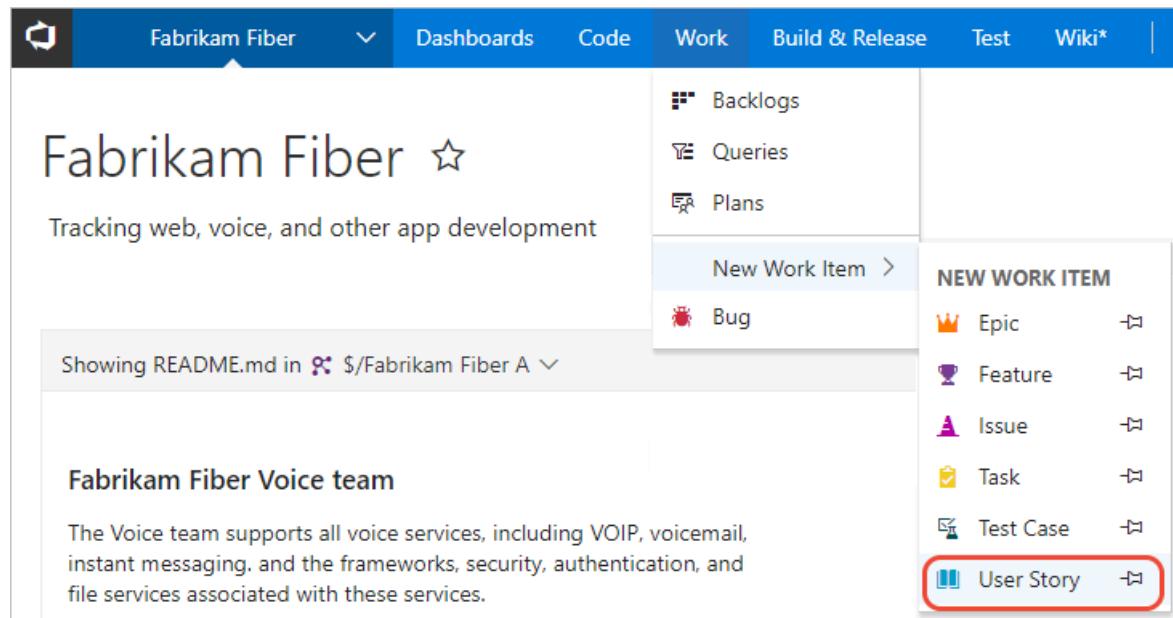
You can quickly add a query or work item when working from a **Boards** or **Work** page.

Choose a **Boards** page—such as **Work Items**, **Boards**, or **Backlogs**. Then choose the **+** plus icon and select from the menu of options.



The screenshot shows the 'Work Items' page in the Azure DevOps interface. The URL is 'fabrikam / Fabrikam Fiber / Work / Work Items'. On the left, there's a sidebar with links: Overview, Boards (selected), Work Items, Boards, Backlogs, and Sprints. The main area shows a table of work items with columns: ID, Title, and State. Three items are listed: 368 (Change background color, Approved), 390 (Cancel order form, Committed), and 492 (Build Settings Experience, New). Above the table, there's a navigation bar with 'New work item' and a 'New Work Item' dropdown. A red arrow points from the 'New query' button to the 'New Work Item' dropdown.

From a **Work** page, you can add a work item from the menu of options as shown in the following image.



The screenshot shows the 'Fabrikam Fiber' Work page. The top navigation bar includes links for Dashboards, Code, Work (selected), Build & Release, Test, and Wiki\*. The main content area displays the 'Fabrikam Fiber' project name and a brief description: 'Tracking web, voice, and other app development'. Below this, it says 'Showing README.md in \$/Fabrikam Fiber A'. A sidebar on the right lists Backlogs, Queries, Plans, and a 'New Work Item' dropdown. Under 'New Work Item', there are options: Bug (selected), Epic, Feature, Issue, Task, Test Case, and User Story (highlighted with a red box).

Or, you can open one of the pages—**Boards**, **Backlogs**, **Queries**, or **Plans**—to add an artifact specific to each of these functional pages.

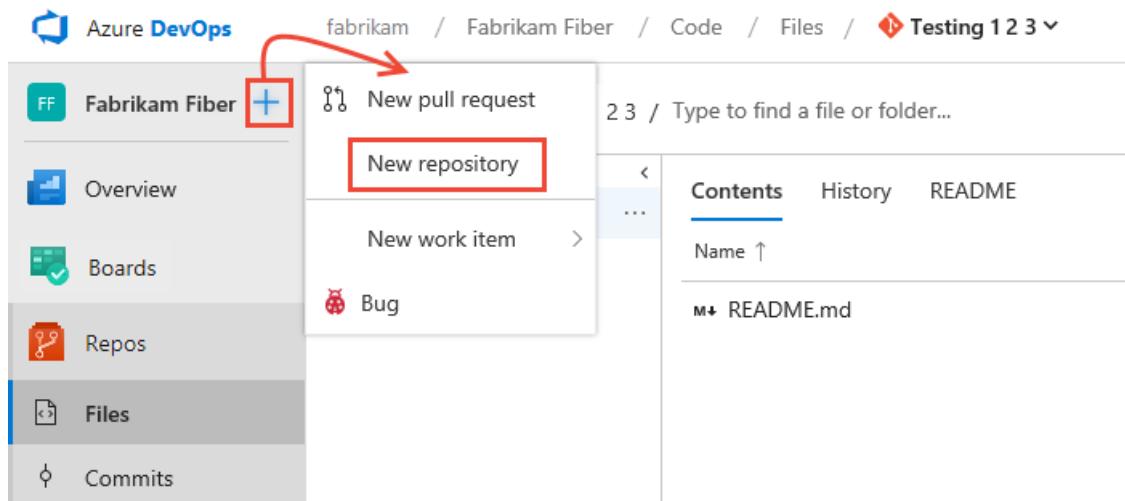
To add other work tracking artifacts, see one of the following articles:

- To add a board, backlog, or sprint backlog, first [add a team](#) which will be associated with those artifacts
- [Add a delivery plan](#)
- [Add a managed work item query](#)
- [Add work items](#).

## Add a pull request or Git repository

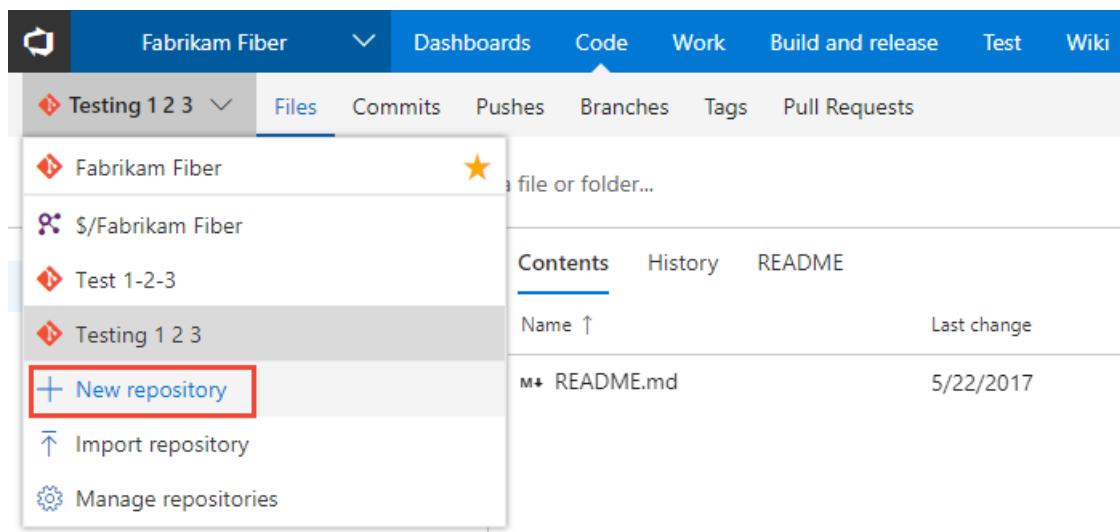
You can quickly add a pull request, Git repository, or work item using the **Add** menu when working from **Code**.

Expand the **Repos** service and choose **Files**, **Commits**, or **Pull Requests** (Git repos) or **Files**, **Changesets**, or **Shelvesets** (TFVC). Then, choose the **+** plus icon and select from the menu of options.



For details on adding a Git repository, see [Git repository](#).

From **Code**, open the context menu for the current repository and choose **+** **New repository**. For details on adding a Git repository, see [Git repository](#)



From one of the other **Code** pages, you can add files or folders, a new branch, or a new pull request.

Note that you can only add one TFVC repository per project, but an unlimited number of Git repositories. To learn more about Git artifacts, see one of the following articles:

- [Git repository](#)
- [Git branch](#)
- [Git pull request](#)
- [Add work items](#)

## Add build and release pipelines

Expand **Pipelines** and choose **Builds** or **Releases**. Then choose the **+** plus icon and select from the menu of options.

From **Build and Release**, choose **Builds**, **Releases**, or other page to add an artifact associated with that page.

To learn more about adding other pipeline related artifacts, see the following articles:

- [Deployment groups](#)
- [Task groups](#)
- [Variable groups](#)
- [Secure files](#)

## Add a team

Agile tools and dashboards are typically associated with teams. You add teams to a project. To learn more about teams, see [About teams and Agile tools](#). To add a team, see [Add a team and team members](#).

## View teams already defined

To view the set of defined teams, open **Project settings**, and choose **Overview**.

The screenshot shows the 'Project Settings > Teams' page for the 'Fabrikam Fiber' project. On the left, there's a sidebar with icons for Overview, Boards, Repos, Pipelines, Test Plans, Artifacts, and 'Project settings'. The 'Project settings' item is highlighted with a red box. In the center, under 'General', the 'Teams' option is selected and highlighted with a red box. To the right, a table lists the defined teams:

Team Name	Members	Description
Customer Service	7	
Fabrikam Fiber Team	7	
Management team	1	
Phone	1	
Voice	1	
Web	2	

To view the set of defined teams, open the admin context for the project, and choose **Overview**.

The screenshot shows the 'Fabrikam Fiber' project overview page. The 'Overview' tab is selected and highlighted with a red box. On the left, there's a 'Project profile' section with details: Name (Fabrikam Fiber), Process (Scrum | [Change process](#)), and Description (Web, voice, and phone apps). On the right, there's a 'Teams' section with a table:

Team Name	Members	Description
Customer Service	7	
<b>Fabrikam Fiber Team</b>	7	The default project team.
Management team	1	
Phone	1	
Voice	1	
Web	2	

## Add a dashboard

Dashboards are associated with a team or a project. Each team can create and configure a number of dashboards. And, any team member can create one or more project dashboards. To learn how, see [Add a dashboard](#).

Dashboards are associated with a team. Each team can create and configure a number of dashboards. To learn how, see [Add a dashboard](#).

## Add a wiki

If you don't have a wiki yet, you can add one. Once added, you can add and update pages to that wiki.

- [Create a wiki](#)
  - [Add and edit wiki pages](#)
  - [Publish a Git repository to a wiki](#)
- 
- [Create a wiki](#)
  - [Add and edit wiki pages](#)

## Related articles

- [Azure Artifacts](#)
- [Exploratory & Manual Testing](#)

# Use breadcrumbs, selectors, and directories to navigate and open artifacts

12/13/2022 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To quickly navigate to a feature or artifact—such as a dashboard, repository, product backlog, Kanban board, build pipeline—you can use breadcrumbs, selectors, and directories.

## Organization and project breadcrumbs

To navigate to the project summary page, choose the project link within the breadcrumbs. To navigate to the organization page with all projects defined for the organization, choose the organization link.

The screenshot shows the 'Work Items' page in Azure DevOps. At the top, there's a breadcrumb navigation bar: 'Organization / Project / Service / Page'. A red arrow points from the word 'Organization' to the 'fabrikam' link in the breadcrumb. Below the breadcrumb is the Azure DevOps logo and the project name 'Fabrikam Fiber'. The left sidebar has a 'Work Items' item selected, indicated by a blue border. The main area displays a table of work items with columns for ID, Title, and State. The first few rows of data are:

ID	Title	State
390	Cancel order form	Committed
492	Build Settings Experience	New
375	Check service status	Committed
543	Develop form	To Do

Horizontal navigation doesn't provide a breadcrumb structure for the organization and project levels. Instead, you can select a recent team or project from the project/team selector.

The screenshot shows the Azure DevOps homepage. The top navigation bar includes links for 'Dashboards', 'Code', 'Work', 'Build & Release', and 'Test'. Below the navigation is a 'Recent projects/teams' dropdown menu. A red box highlights the dropdown arrow icon. The menu lists several projects: 'Fabrikam Fiber Home', 'Agile 11', 'FabrikamFiber', 'Fabrikam Fiber A' (which is currently selected), 'Fabrikam Fiber PB', and 'Browse...'. At the bottom of the menu, there's a note about README.md files and a link to learn more about Markdown.

Choosing [Browse...](#) opens the projects page.

# Selectors

Selectors are used to select an artifact within the current page. Most Agile tools are defined for a team and therefore require selection of the team artifact or tool.

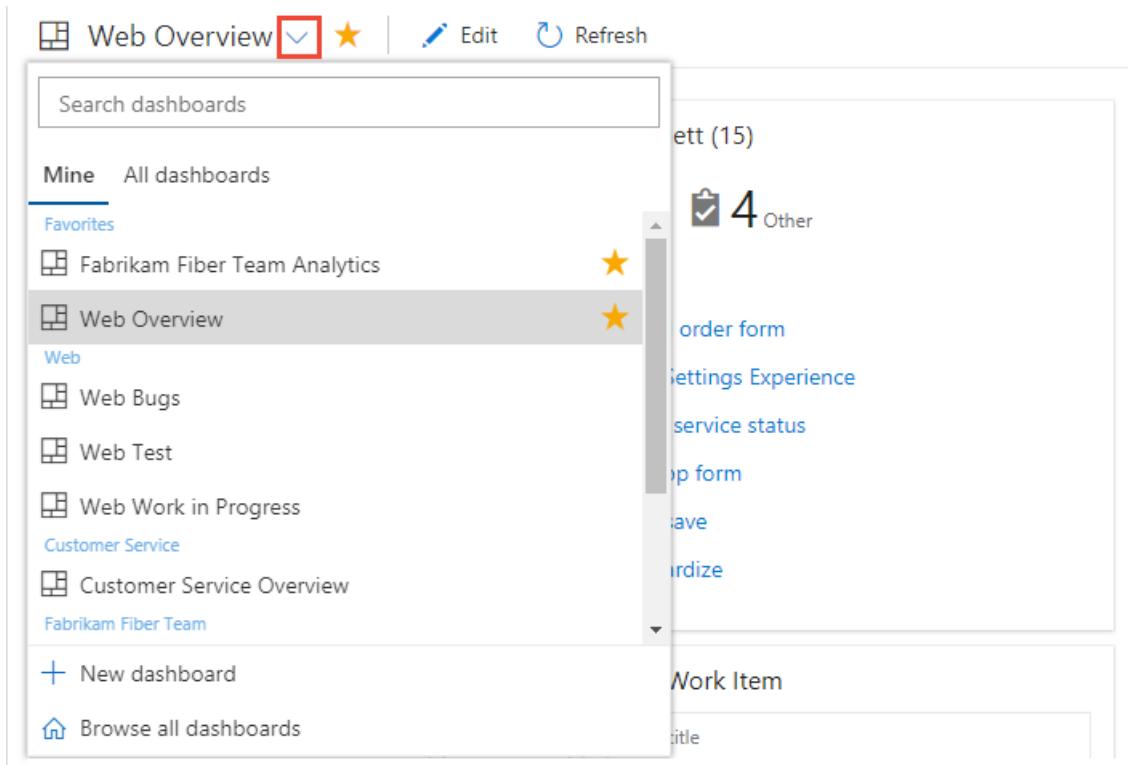
Selectors are used to select an artifact within the current page. Most Agile tools are defined for a team and therefore require selection of the team as well as the specific page.

## NOTE

When you navigate to a specific page or artifact, the system remembers your selection. You use selectors to choose a different artifact within the current page.

### Example: Dashboard selector

Within Dashboards, you open a specific dashboard from the selector.



This particular selector features these navigational elements:

- Search box for filtering dashboards based on a team name or keyword
- Two pages you can choose from:
  - Mine (dashboards you created) which are organized by team
  - All (dashboards created by everyone) which are listed alphabetically
- Dashboards you've favorited will appear at the top of the selector
- Add new dashboard feature
- Browse all dashboards - opens Dashboards>All

Within Dashboards, you select the team whose dashboards you want to view.

The screenshot shows the 'Fabrikam Fiber' application interface. At the top, there's a navigation bar with links for Dashboards, Code, Work, Build & Release, Test, and a gear icon. Below the navigation bar is a sidebar menu with sections for 'Recent projects/teams', 'Agile 11', 'FabrikamFiber', 'Fabrikam Fiber A' (which is selected), 'Fabrikam Fiber PB', 'Browse...', 'New team', and a note about a README.md file. The main content area displays a placeholder message: 'A README.md file is intended to quickly orient readers to what your project can do. Learn more about Markdown.'

Then, choose the name of the dashboard to view it.

For example, here we open the *Work in Progress* dashboard.

The screenshot shows the 'Fabrikam Fiber / Web' application interface. The top navigation bar includes links for Dashboards, Code, Work, Build and Release, and a gear icon. Below the navigation bar, a secondary navigation bar has tabs for Overview, Bugs, Work in Progress (which is highlighted with a red box), and Test. The main content area is currently empty.

### Example: Backlogs

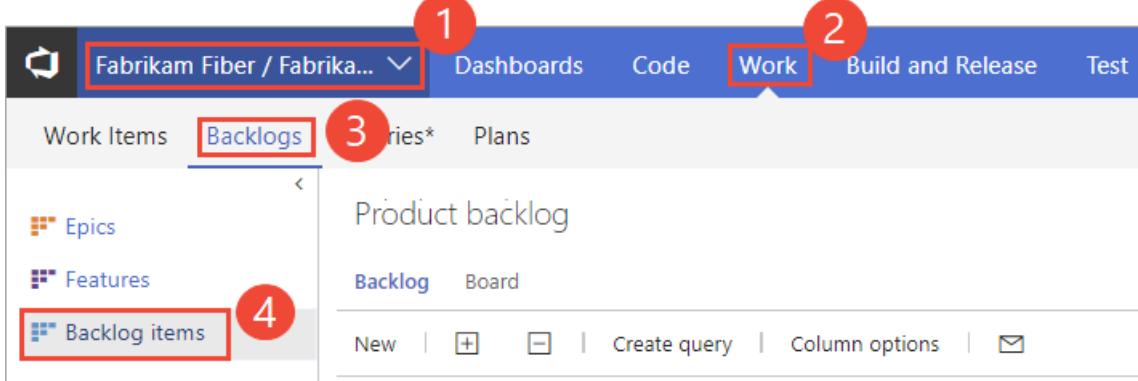
From the **Boards>Backlogs** page, you use the selector to switch to another team's backlog. Again, favorited backlogs appear towards the top of the menu. You can also filter the list based on a team name or keyword.

The screenshot shows the 'Boards>Backlogs' page. On the left, there's a sidebar with a project selector (set to 'Web') and a search bar ('Search team backlogs'). Below the search bar is a list of teams: Phone, Web (which is selected and highlighted with a red box), Customer Service, Fabrikam Fiber Team, Management team, and Voice. At the bottom of the sidebar is a link to 'Browse all team backlogs'. The main content area shows a table of backlog items for the 'Web' team. The table has columns for 'Index', 'Title', 'State', and 'Last updated'. The items listed are:

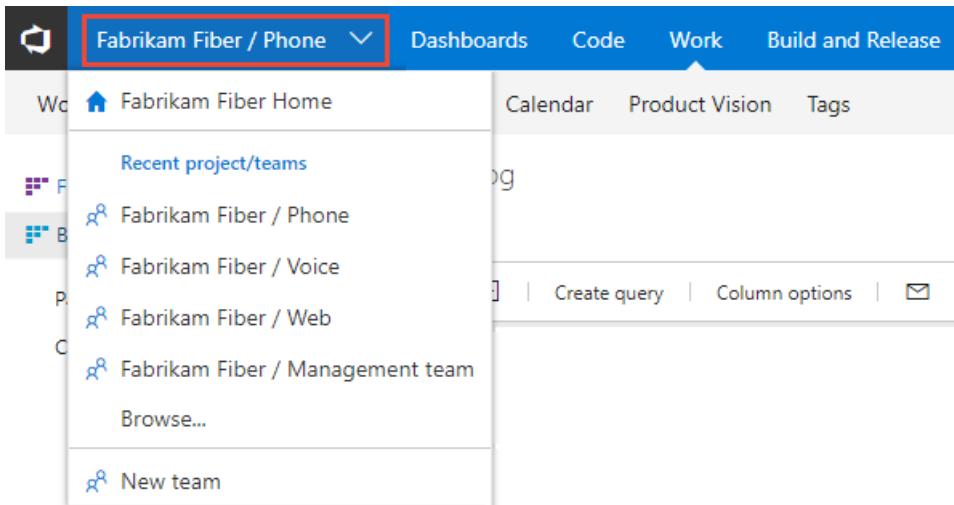
Index	Title	State	Last updated
8	Product Backl... > Request support	● Committed	2023-01-15
9	Product Backl... > Cancel order form	● Committed	2023-01-15

Or, choose **Browse all team backlogs** to open the **Backlogs>All** page.

(1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).



To choose another team, open the project/team selector and select a different team or choose the **Browse** option.



## Artifact breadcrumbs and selectors

Within select pages, breadcrumbs are provided to support navigating within the page or opening an artifact.

### Example: Queries folders and breadcrumbs

For example, when working in the **Queries** pages, you can navigate to a subfolder, folder, or page.

The screenshot shows the Azure DevOps **Queries** page. The breadcrumb navigation at the top indicates the current path: **Queries page / Queries folder / Queries subfolder / query**. A red arrow points to the "Shared Queries" link in the breadcrumb.

The left sidebar shows a navigation menu with the following items:

- Overview
- Boards** (selected)
- Work Items
- Boards
- Backlogs
- Sprints
- Queries** (selected)

The main content area displays a table of queries with columns: ID, State, Assigned To, Remaining Work, and Title. One query is highlighted with a blue background and a checkmark icon, labeled "Standardize on form".

ID	State	Assigned To	Remaining Work	Title
399	In Progress	Jamal Hartnett	6	<input checked="" type="checkbox"/> Standardize on form
539	In Progress	Jamal Hartnett	8	<input checked="" type="checkbox"/> Standardize
538	In Progress	Johnnie McLeod	8	<input checked="" type="checkbox"/> Design welcome screen
371	In Progress	Johnnie McLeod	8	<input checked="" type="checkbox"/> Auto-complete user's name in form
388	In Progress	Raisa Pokrovskaya	6	<input checked="" type="checkbox"/> Code form

Also, you can choose a query that you've favorited from the selector menu, Or, you can choose to browse all queries which returns you to the **All Queries** page.

All Queries > Shared Queries > Current Sprint > Work in Progress

Results Editor Charts | Run query +

ID	State	Assigned To	Remaining Work
399	In Progress	Jamal Hartnett	6
539	In Progress	Jamal Hartnett	8
538	In Progress	Johnnie McLeod	8
371	In Progress	Johnnie McLeod	8
388	In Progress	Raisa Pokrovskaya	6

All items in a tree query ★

Work in Progress ★

Browse all queries

Standardize

Design welcome screen

Auto-complete user's name in form

Code form

### Example: Pipeline folders and breadcrumbs

Breadcrumb-and-selector navigation elements are used within most services that support defining and organizing artifacts within folders. This includes **Pipelines** or **Build and Release** applications pages.

Build Definitions / RedTeam / RedCreator

Summary History Deleted

Details

Repository vtbastmatt Default queue Queue status Enabled Last updated by Tuesday, April 3, 2018 6:30 AM

Queued & running

No builds queued or running at the moment

Recently completed

#41 succeeded master

Choose the **Deployment** breadcrumb link to return to the *Deployment* folder.

Fabrikam Fiber

Builds Releases Releases\* Library Task groups Deployment Groups Build Tags

... > Deployment > Fabrikam Fiber-CI

## Directories

Directories provide a filterable list of all artifacts defined for a service area. Often when you navigate to an application, it will open the application's directory.

For example, here is the **Boards**>**Boards** directory.

Name	Team
Fabrikam Team boards	Fabrikam Team
My favorite boards (1)	
Fabrikam Team boards	Fabrikam Team
My team boards (8)	
Account Management boards	Account Management
Customer Profile boards	Customer Profile
Fabrikam Team boards	Fabrikam Team
Phone boards	Phone
Service Delivery boards	Service Delivery
Service Status boards	Service Status
Shopping Cart boards	Shopping Cart
TV boards	TV

It lists boards in the following order:

- Your last visited board
- Your favorited boards
- All boards of teams that you belong to
- All boards defined for the project in alphabetical order.

Choose the filter icon to filter the list as described in [Filter basics](#).

From a specific page, you can open the directory from the breadcrumbs or a selector. For example, choose **Browse all boards** from the Boards selector.

[Open from breadcrumb](#)

[Open from selector](#)

The screenshot shows the 'Browsing boards' interface. At the top, there's a search bar labeled 'Search team boards'. Below it, under 'Favorites', is a board titled 'Fabrikam Team' with a yellow star icon. Under 'My Team Boards', there are several boards: 'Account Management', 'Customer Profile', 'Fabrikam Team' (with a yellow star), 'Phone', and 'Service Delivery'. At the bottom of the list is a button labeled 'Browse all boards', which is highlighted with a red box.

fabrikam / Fabrikam Fiber / Boards / **Boards**

## Team profiles

- Open a team profile to quickly access items defined for a team. The team profile is available from the Overview > Dashboards, Boards > Boards, Boards > Backlogs, and Boards > Sprints pages.

The screenshot shows the 'Service Delivery' team profile page. At the top, there's a header with the Azure DevOps logo, the project name 'fabrikam / Fabrikam Fiber', and a 'Backlogs' tab. On the left, there's a sidebar with a 'Fabrikam Fiber' icon, a '+' sign, and an 'Overview' link. The main area shows the team name 'Service Delivery' with a yellow star icon and a 'g8' icon. Below the team name are buttons for 'New Work Item', 'View as Board', 'Column Options', and three dots. A close button 'X' is at the top right.

A panel opens that shows all items defined for the team.

The screenshot shows the expanded 'Service Delivery' team profile. At the top, there's a purple circular icon with two people, the team name 'Service Delivery', the project name 'Fabrikam Fiber', and a 'Team Settings' link. Below this, there are tabs for 'Items' (which is selected) and 'Members (7)'. A dropdown menu shows 'All Items'. The main area lists four items: 'Service Delivery Boards' (with a yellow star), 'Service Delivery Backlogs' (with a yellow star), 'Sprint 2 Sprints' (with a yellow star), and 'Overview Dashboards' (with a yellow star).

- You can filter the list to show only Dashboards, Boards, Backlogs, or Sprints by choosing from the menu.

The screenshot shows the 'Service Delivery' team settings page. At the top, there's a purple icon with two people, followed by the team name 'Service Delivery', the project name 'Fabrikam Fiber', and the link 'Team Settings'. Below this, there are two tabs: 'Items' (which is underlined in blue) and 'Members (7)'. A red box highlights the 'Items' tab. A dropdown menu is open, showing five options: 'All Items' (which is highlighted with a light blue background), 'Dashboards', 'Boards', 'Backlogs', and 'Sprints'. There is also a small 'X' button at the top right of the dropdown.

- To view the team admins and members of the team, choose **Members**.

The screenshot shows the 'Service Delivery' team settings page. At the top, there's a purple icon with two people, followed by the team name 'Service Delivery', the project name 'Fabrikam Fiber', and the link 'Team Settings'. Below this, there are two tabs: 'Items' and 'Members (7)', with 'Members (7)' being underlined in blue and highlighted with a red box. A red box also highlights the 'Members' tab. The main area shows sections for 'Admins' and 'Members'. Under 'Admins', there's one entry for 'Cristina Potra' with a profile picture and initials 'CP'. Under 'Members', there are five entries: 'Christie Church' (profile picture and initials 'CC'), 'Chuck Reinhart' (profile picture and initials 'CR'), 'Jamal Hartnett' (profile picture and initials 'JH'), 'Johnnie McLeod' (profile picture and initials 'JM'), and 'Raisa Pokrovskaya' (profile picture and initials 'RP').

- To view or change the team configuration, choose **Team Settings**.

You can then add [team members](#), [team admins](#), or navigate to [team notifications](#), or team [iterations](#) and [area paths](#).

See also [Manage and configure team tools](#).

## Related articles

- About teams and Agile tools
- Add an artifact or team
- Set favorites
- Open a service or page
- Filter basics

# Switch project, repository, team

12/13/2022 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Several features depend on the project, repository, or team that you have selected. For example, dashboards, backlogs, and board views will change depending on the project and team you select.

Also, when you add a work item, the system references the default area and iteration paths defined for the team context. Work items you add from the team dashboard (new work item widget) and queries page are assigned the team default iteration. Work items you add from a team backlog or board, are assigned the team default backlog iteration. To learn more, see [About teams and Agile tools](#).

## Prerequisites

- You must be added to a project as a member of the **Contributors** or administrator security group. To get added, [Add users to a project or team](#).

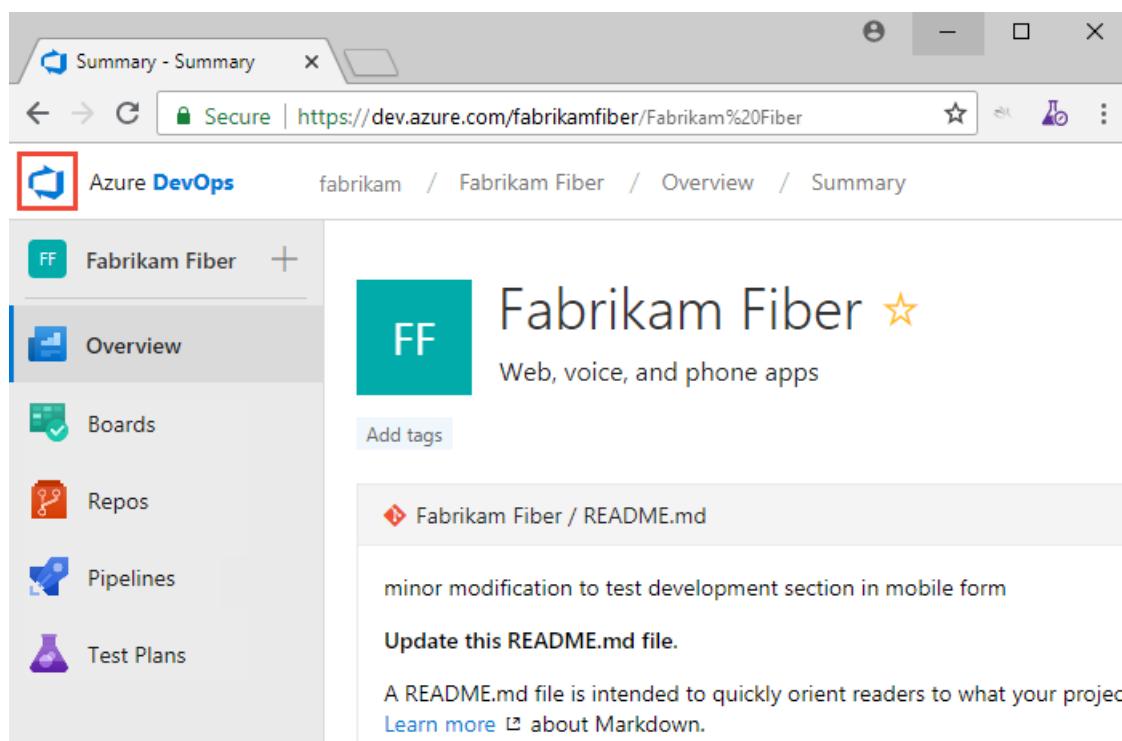
### NOTE

If the **Limit user visibility and collaboration to specific projects** preview feature is enabled for the organization, users added to the **Project-Spaced Users** group won't be able to access projects that they haven't been added to. To learn more, see [Manage your organization, Limit user visibility for projects and more](#).

## View and open a project

From the **Projects** page you can quickly navigate to a project that you have permissions to view.

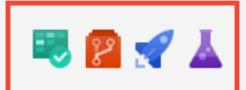
1. Choose the  Azure DevOps logo to open **Projects**.



The projects you most recently viewed are displayed, followed by a list of all projects in alphabetic order.

2. Hover over the dots and you can open the service of interest for that project.

**Projects**  [+ Create project](#)

<b>FF</b>	<b>Fabrikam Fiber</b> Web, voice, and phone apps	<b>M</b>	<b>MyFirstProject</b>
...			
<a href="#">All projects</a>			
<b>A1</b>	<b>Agile 11</b> New agile project	...	
<b>D1</b>	<b>Demo 11</b> Agile team project		
<b>FF</b>	<b>Fabrikam Fiber</b> Web, voice, and phone apps	...	
<b>M</b>	<b>MyFirstProject</b>	...	

3. You can filter the project and team list using the *Filter projects* search box. Simply type a keyword contained within the name of a project or team. Here we type **Fabrikam** to find all projects or teams with *Fabrikam* in their name.

**Projects**  [X](#) [+ Create project](#)

<b>FF</b>	<b>Fabrikam Fiber</b> Web, voice, and phone apps	...
<b>FT</b>	<b>Fabrikam Test</b> Project used to verify MyAgile process customizations	...
<b>F</b>	<b>FabrikamFiber</b> Customer-focused apps under development based on Agile process.	...

4. Choose **Create Project** to add a project. You must be an account administrator or a member of the Project Collection Administrators group to [add a project](#).

The screenshot shows the Azure DevOps 'Projects' page. On the left, there's a sidebar titled 'My accounts' with three entries: 'fabrikamfib' (dark blue), 'FabrikamFiber' (light blue), and 'fabrikam-fiber' (green). The main content area is titled 'Projects' and features a search bar 'Filter projects'. A prominent red box highlights the '+ Create project' button. Below the button are two project cards: 'Fabrikam Fiber' (Web, voice, and phone apps) and 'MyFirstProject'. Each card has a small icon (teal for Fabrikam Fiber, magenta for MyFirstProject) and a horizontal ellipsis below it.

From the **Projects** page you can quickly navigate to a project or a team that you've accessed or worked in previously. Projects and teams are listed in the order you've last accessed, with the most recent five projects accessed appearing first. All projects you've accessed are listed within the **All** section.

1. Choose the  Azure DevOps logo to open **Projects**.

The screenshot shows a browser window with the URL 'https://dev.azure.com/fabrikamfiber/Fabrikam%20Fiber'. The navigation bar at the top includes links for 'Fabrikam Fiber', 'Dashboards', 'Code', 'Work', 'Build and release', 'Test', and 'Wiki'. The 'Fabrikam Fiber' link is highlighted with a red box.

The projects you most recently viewed are displayed, followed by a list of all projects in alphabetic order.

The screenshot shows the 'Projects' page with a navigation bar where 'Projects' is selected. Below is a 'Recent' section listing projects: 'Fabrikam Fiber', 'Fabrikam Fiber / Web' (marked with a yellow star), and 'FabrikamFiber'. At the bottom is a 'Filter projects and teams' search bar and a 'New Project' button.

2. As you hover over a project or team, you can choose one of the links to go to **Home** or **Dashboards**, **Code**, **Work**, **Build and Release**, **Test**, or **Wiki** pages. Choose the  star icon to mark the project as a favorite.

The screenshot shows the 'Fabrikam\_Fiber' project selected in the navigation bar. The navigation bar includes links for 'Dashboards', 'Code', 'Work', 'Build and Release', 'Test', and 'Wiki'.

3. You can filter the project and team list using the *Filter projects and teams* search box. Simply type a keyword contained within the name of a project or team. Here we type **Fabrikam** to find all projects or teams with *Fabrikam* in their name.

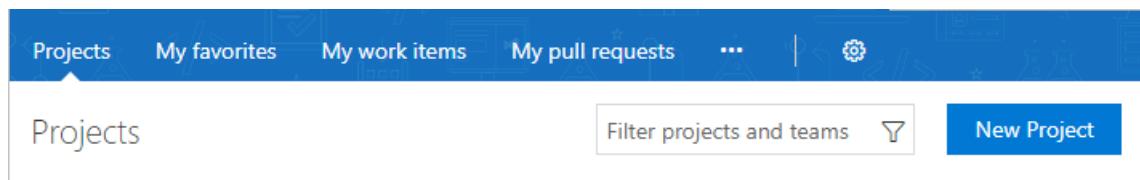
## Projects

Fabrikam 

### Results

-  Fabrikam Fiber
-  Fabrikam Fiber / Customer Service 
-  Fabrikam Fiber / Management team 
-  Fabrikam Fiber / Phone 
-  Fabrikam Fiber / Voice 
-  Fabrikam Fiber / Web 
-  Fabrikam Test
-  FabrikamFiber

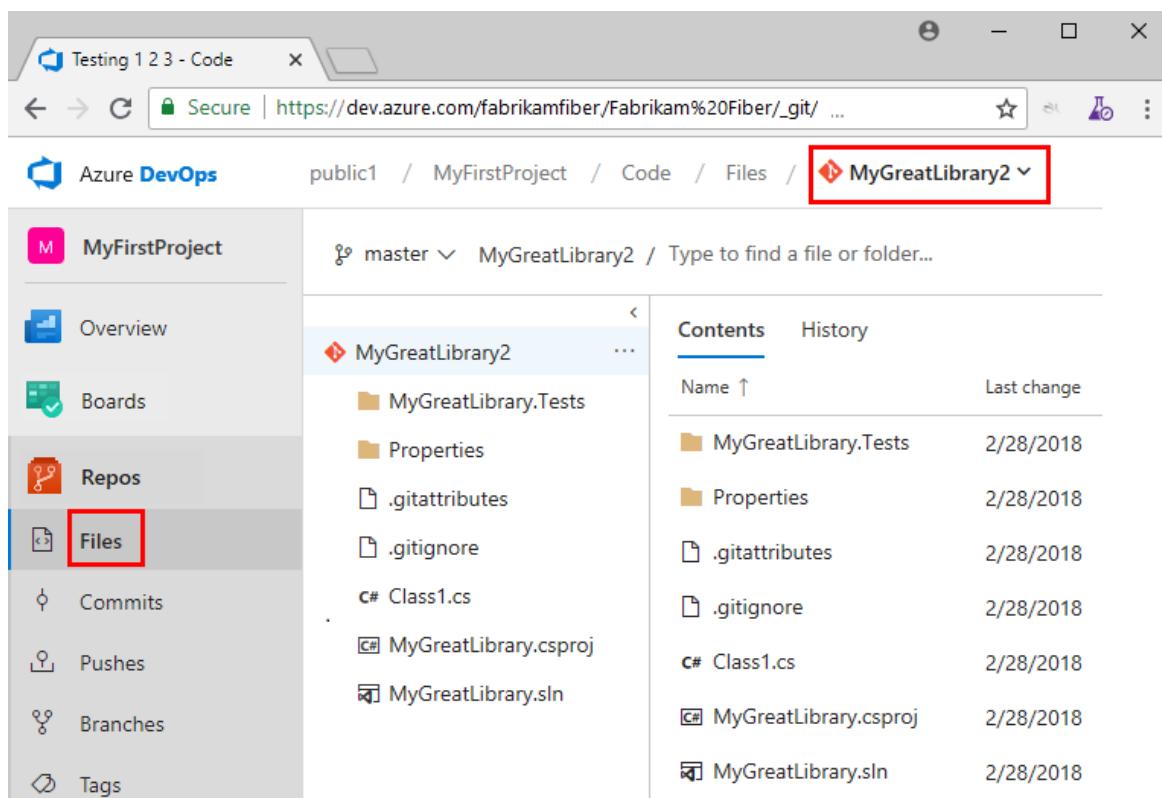
4. Choose **New Project** to add a project. You must be an account administrator or a member of the Project Collection Administrators group to [add a project](#).



The screenshot shows the Azure DevOps interface with the 'Projects' tab selected in the top navigation bar. Below the navigation, there's a search bar labeled 'Filter projects and teams' and a blue 'New Project' button. A list of projects is displayed, including 'Fabrikam Fiber', 'Fabrikam Fiber / Customer Service', 'Fabrikam Fiber / Management team', 'Fabrikam Fiber / Phone', 'Fabrikam Fiber / Voice', 'Fabrikam Fiber / Web', 'Fabrikam Test', and 'FabrikamFiber'. Each project entry has a yellow star icon to its right.

## View and open a repository

1. Choose **Repos > Files**.



The screenshot shows the Azure DevOps repository interface. On the left, a sidebar menu is open with options: 'MyFirstProject' (selected), 'Overview', 'Boards', 'Repos' (selected), 'Files' (highlighted with a red box), 'Commits', 'Pushes', 'Branches', and 'Tags'. The main content area shows a list of files under the 'MyGreatLibrary2' repository. The 'Contents' tab is selected. The file list includes:

Name	Last change
MyGreatLibrary.Tests	2/28/2018
Properties	2/28/2018
.gitattributes	2/28/2018
.gitignore	2/28/2018
Class1.cs	2/28/2018
MyGreatLibrary.csproj	2/28/2018
MyGreatLibrary.sln	2/28/2018

2. Select the repository of interest from the repository selector.

The screenshot shows the GitHub interface for a user named 'public1'. The user has selected the 'MyFirstProject' organization. In the 'Code' section, the 'Files' tab is active. A dropdown menu titled 'MyGreatLibrary2' is open, listing several repositories: 'breadth-of-trees', 'Compose', 'GVFS', 'MyGreatLibrary', 'MyGreatLibrary2', 'RedBot', 'Specs', 'VSCode', 'New repository', 'Import repository', and 'Manage repositories'. The 'MyGreatLibrary2' item is highlighted with a red box and a yellow star icon. A red arrow points from the 'MyGreatLibrary2' dropdown to the 'MyGreatLibrary2' item in the list.

1. Choose **Code**.

The screenshot shows the main project page for 'MyFirstProject'. The top navigation bar is blue with tabs for 'MyFirstProject', 'Dashboards', 'Code' (which is highlighted), 'Work', 'Build and Release', and 'Wiki'. The project name 'MyFirstProject' is displayed prominently with a large 'M' icon. Below it, the text 'No description provided' is shown. A 'TypeScript' badge is present. A file list on the right includes 'VSCode / README.md'.

2. Select the repository from the selector.

The screenshot shows the 'Files' tab of the 'MyGreatLibrary' repository. A red box highlights the repository selector at the top left, and a red circle with the number '1' is placed over it. Another red box highlights the 'MyGreatLibrary' repository in the list below, and a red circle with the number '2' is placed over it. The list includes 'breadth-of-trees', 'Compose', 'GVFS', 'MyGreatLibrary' (highlighted), 'MyGreatLibrary2', 'RedBot', 'Specs', and 'VSCode'. To the right, a table shows commit history for files like 'MyGreatLibrary.Tests', 'MyGreatLibrary.csproj', and 'MyGreatLibrary.sln'. The commits are all marked as 'succeeded'.

File	Last change	Commits
MyGreatLibrary.Tests	2/28/2018	2fb94e4 create library
MyGreatLibrary.csproj	2/28/2018	2fb94e4 create library
MyGreatLibrary.sln	2/28/2018	2fb94e4 create library

## Switch to a different team

From a user page, one under—**Boards**, **Repos**, **Pipelines**, or **Test Plans**—you can't switch to a different team, you can [only select team artifacts](#).

From a **Project Settings>Work>Team configuration** page, you select a team from the team selector breadcrumb.

Project Settings > Team configuration > **Fabrikam Fiber Team**

General Work General Iteration

Backlogs See only the backlog

Epics Features Backlogs

More teams

Working days

Capacity and burndown are based on the days your team works.

Select days

Monday Tuesday

You can switch your team focus to one that you've recently viewed from the project/team selector. If you don't see the team or project you want, choose **Browse...** or choose the Azure DevOps logo to [access the Projects page](#).

Fabrikam Fiber

Recent projects/teams

Agile 11

FabrikamFiber

Fabrikam Fiber A

Fabrikam Fiber PB

Browse...

New team

A README.md file is intended to quickly orient readers to what your project can do.

Learn more about Markdown.

## Related articles

- [Work across projects](#)
- [Add teams](#)

# Tutorial: Set personal or team favorites

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Favorite  those views that you frequently access. You can favorite all sorts of Azure DevOps features and tools—such as a project, repository, build pipeline, dashboard, backlog, board, or query. You can set favorites for yourself or your team.

As your code base, work tracking efforts, developer operations, and organization grows, you'll want to be able to quickly navigate to those view of interest to you and your team. Setting favorites allows you to do just that.

Team favorites are a quick way for members of your team to quickly access shared resources of interest. You favorite an item for yourself by choosing the  star icon. The favorited item will then show up easily from one or more directory lists. You set favorites for a team through the context menu for the definition, view, or artifact.

In this tutorial you'll learn how to view your personal favorites and to favorite or unfavorite the following views:

- Project or team
- Dashboard
- Team backlog, board, shared query, or other Azure Boards view
- Repository
- Build and release definition
- Test plans
  
- Project
- Shared query
- Repository
- Build and release definition
- Test plans

## Prerequisites

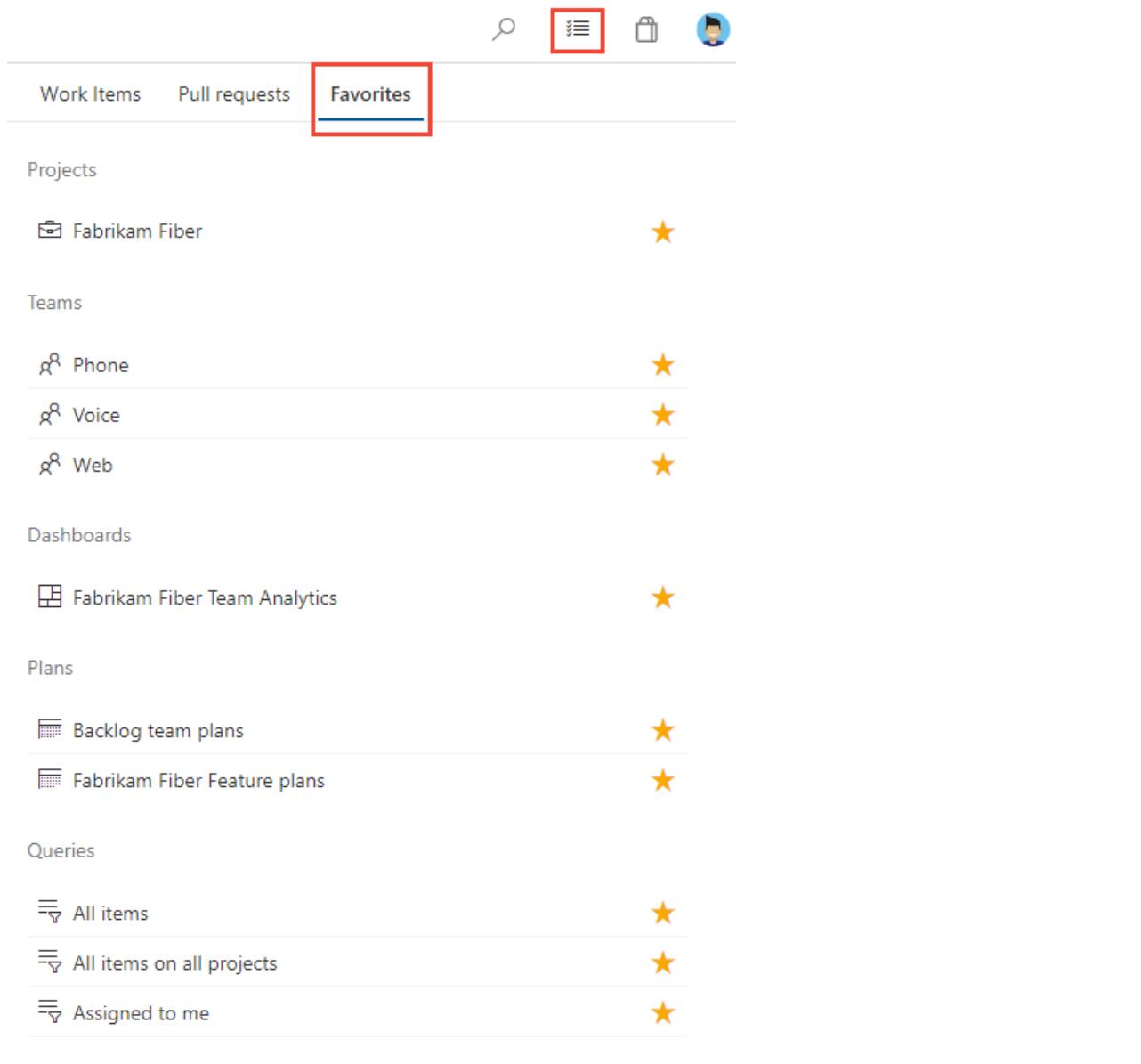
- You must connect to a project through the web portal. If you don't have a project yet, [create one](#). To connect to the web portal, see [Connect to a project](#).
- You must be a member of the **Contributors** or an administrators security group of the project. To get added, [Add users to a project or team](#).
- To favorite projects, backlogs, boards, queries, dashboards, or pipeline views, you must have **Stakeholder** access or higher.
- To favorite repositories, or delivery plans, you must have **Basic** access or higher.
- To favorite test plans, you must have **Basic + Test Plans** access level or equivalent.
  
- You must connect to a project through the web portal. If you don't have a project yet, [create one](#). To connect to the web portal, see [Connect to a project](#).
- You must be a member of the **Contributors** or an administrators security group of the project. To get added, [Add users to a project or team](#).
- To favorite projects, backlogs, boards, queries, dashboards, or pipeline views, you must have **Stakeholder** access or higher.
- To favorite repositories, or delivery plans, you must have **Basic** access or higher.

- To favorite test plans, you must have **Basic + Test Plans** access level or equivalent.

For details about the different access levels, see [About access levels](#).

## View personal favorites

Access views that you have favorited by choosing the  inbox icon, and then choosing **Favorites**.



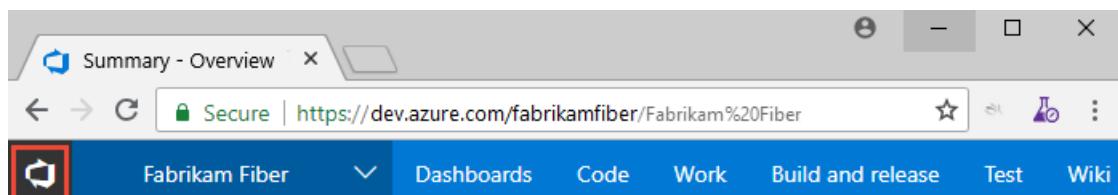
The screenshot shows the 'Favorites' view in Azure DevOps. At the top, there's a navigation bar with icons for search, inbox (highlighted with a red box), library, and profile. Below the bar, tabs for 'Work Items', 'Pull requests', and 'Favorites' are visible, with 'Favorites' being the active tab. The main area is divided into sections: 'Projects', 'Teams', 'Dashboards', 'Plans', and 'Queries'. Each section lists items with a star icon indicating they are favorited. For example, under 'Teams', there are three items: 'Phone', 'Voice', and 'Web', each with a yellow star. Under 'Plans', there are two items: 'Backlog team plans' and 'Fabrikam Fiber Feature plans', both with stars. Under 'Queries', there are three items: 'All items', 'All items on all projects', and 'Assigned to me', all with stars.

Category	Item	Status
Projects	Fabrikam Fiber	★
	Phone	★
	Voice	★
Teams	Web	★
	Fabrikam Fiber Team Analytics	★
	Backlog team plans	★
Plans	Fabrikam Fiber Feature plans	★
	All items	★
	All items on all projects	★
Queries	Assigned to me	★

### NOTE

If a service is disabled, then you can't favorite an artifact or view of that service. For example, if **Boards** is disabled, then the favorite groups—Plans, Boards, Backlogs, Analytics views, Sprints, and Queries and all Analytics widgets—are disabled. To re-enable a service, see [Turn an Azure DevOps service on or off](#).

1. Access views that you have favorited by choosing the  Azure DevOps logo to open **Projects**.



2. Choose **My Favorites** to quickly access any view or item that you've marked as a favorite.

## Favorites

Filter favorites



### Queries

Bug Triage	Fabrikam Fiber	.../Shared Queries/Current Iteration	
My Bugs	Contoso	Shared Queries	
Open User Stories	Contoso	.../Shared Queries/Current Iteration	
Product Planning	Fabrikam Fiber	Shared Queries	
Product Planning	Contoso	Shared Queries	

## Favorite a project or team

1. To favorite a project, open the project **Summary** page and choose the star icon.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. The top navigation bar includes 'Summary - Summary', a browser address bar showing 'https://dev.azure.com/fabrikamfiber/Fabrikam%20Fiber', and a breadcrumb trail 'fabrikam / Fabrikam Fiber / Overview / Summary'. On the left, a sidebar lists 'Fabrikam Fiber' (selected), 'Overview', 'Summary' (highlighted with a red box), 'Dashboards', 'Analytics views', 'Wiki', and 'Boards'. The main content area displays the project logo 'FF', the name 'Fabrikam Fiber' with a yellow star icon, and the tagline 'Web, voice, and phone apps'. Below this is a 'Fabrikam Fiber / README.md' section containing a link to 'Fabrikam Fiber / README.md', a note about minor modifications, and instructions to update the README file. At the bottom, there's a note about README files being intended to quickly orient readers to the project.

2. To favorite a team artifact, open **Boards > Boards** or **Boards > Backlogs**. Select the team you want to favorite from the team selector and choose the star icon.

The screenshot shows the 'Boards > Backlogs' page. The top navigation bar includes 'Phone' (selected), a search icon, and a refresh icon. The main content area has tabs for 'Backlog items backlog' (selected) and 'Backlog items'. Below these are filter and sort icons.

3. To favorite other team artifacts, choose the team icon, and then choose the star icon next to one of the listed artifacts.

The screenshot shows the 'Phone' team settings in Microsoft Teams. At the top, there's a purple circular icon with three people, followed by the team name 'Phone', the owner 'Fabrikam Fiber', and a 'Team Settings' link. Below this, there are two tabs: 'Items' (selected) and 'Members (1)'. A dropdown menu shows 'All Items'. The main list contains three items: 'Phone Boards', 'Phone Backlogs', and 'Phone Sprints', each with a yellow star icon.

Item	Category	Star Icon
Phone Boards	Boards	Yellow Star
Phone Backlogs	Backlogs	Yellow Star
Phone Sprints	Sprints	Yellow Star

## Favorite a project

To favorite a project, open the project **Summary** page and choose the star icon.

The screenshot shows the 'FabrikamFiber' project summary page. The title 'FabrikamFiber' has a yellow star icon to its right, which is highlighted with a red box. Below the title, it says 'Customer-focused apps under development based on Agile process.' To the right, there are sections for 'Members' (with a 'K' icon and a '+' button), 'Activity' (empty), 'Code' (empty), and 'Build & Release' (empty). A modal window is open in the center, titled 'Use continuous integration', with a sub-instruction 'Improve code quality by detecting breaking changes as soon as they happen.' It features a 'Setup Build' button and a link 'Learn more about continuous integration'. There's also a large circular icon with a download arrow.

Or, you can favorite a project from the **Projects** page by choosing the star icon next to the project.

## Favorite a dashboard

1. From **Overview > Dashboards**, open the selector and choose the **Browse all dashboards** option.

The screenshot shows the Microsoft Power BI 'Mine' page. On the left, there's a sidebar with a search bar at the top. Below it, under 'Favorites', are 'Fabrikam Team Analytics' and 'Account Management Overview'. Under 'Customer Profile', is 'Customer Profile Overview'. Under 'Fabrikam Team', is 'Fabrikam Team Analytics'. At the bottom of the sidebar is a red-bordered button labeled 'Browse all dashboards'. The main area displays two cards: a green one for 'Fabrikam Items' with the number 6 and a purple one for 'Fabrikam Fiber ...' with the number 0. Below these cards, there's a section titled 'Commits by State' with a table showing commit counts for different states.

2. The **Mine** page shows your favorited dashboards, and all dashboards of teams that you belong to. The **All** page (shown below) lists all dashboards defined for the project in alphabetical order. You can filter the list by team or by keyword.

The screenshot shows the Microsoft Power BI 'All' page. At the top, there are tabs for 'Mine' and 'All', with 'All' being selected and highlighted with a red box. There's also a '+ New dashboard' button and a search/filter icon. Below the tabs is a 'Filter dashboards' button and a 'Filter by team' dropdown menu with a search bar and a red box around it. The main area is a table listing various dashboards:

Name	Team
Analytics	Fabrikam Team
Bug status	Fabrikam Team
Bugs	Internet
Overview	Account Management
Overview	Customer Profile
Overview	Email
Overview	Fabrikam Team
Overview	Internet
Overview	Phone
Overview	Service Delivery
Overview	Service Status
Team Guidance	Fabrikam Team
Work in Progress	Internet

At the bottom right of the table, it says 'Active work items'.

**TIP**

You can change the sort order of the list by choosing the column label.

3. To favorite a dashboard, hover over the dashboard and choose the star icon.

The screenshot shows the Azure DevOps interface with the 'Analytics' dashboard selected. At the top, there is a 'Name ↑' sorting header. Below it, a dark button labeled 'Add to favorites' is highlighted with a red box. To its right is a 'Team' link. Further down, a yellow star icon with a red border is also highlighted with a red box, indicating it's the 'Add to favorites' button. The 'Fabrikam Team' name is visible next to the star icon.

Favoriting a dashboard will cause it to appear on your **Favorites** page and towards the top in the **Dashboards** selection menu.

## Favorite a team's backlog, Kanban board, or other view

You can favorite several Agile tools for a team from a **Boards** page.

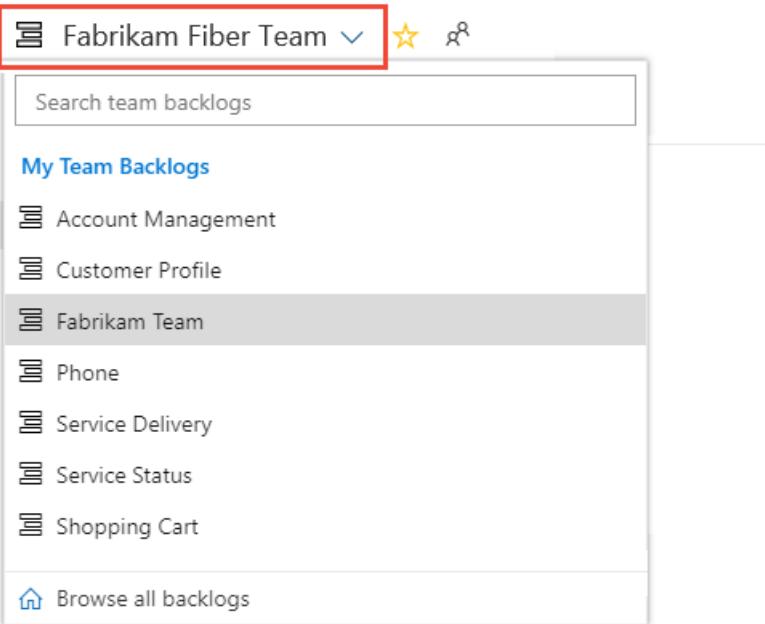
1. Choose **Boards**, and then choose the page of interest, such as **Boards**, **Backlogs**, or **Sprints**.

For example, here we choose (1) **Work** and then (2) **Backlogs**.

The screenshot shows the Azure DevOps 'Boards' page for the 'Fabrikam Fiber' project. On the left, a sidebar lists navigation options: Overview, Boards, Work Items, Boards, Backlogs (which is highlighted with a red box), Sprints, and Queries. The main area displays the 'Fabrikam Fiber Team' backlog. At the top of the backlog list is a yellow star icon with a red border, indicating it is favorited. Below the star are buttons for 'New Work Item', 'Backlog items Board', and 'Backlog items'. The backlog table has columns for Order, Assigned To, State, and Title. Seven items are listed:

Order	Assigned To	State	Title
1	Jamal Hartnett	Committed	Hello World Web Site
2	Jamal Hartnett	Committed	Slow response on informa
3	Raisa Pokrovskaya	New	Add an information form
4	Raisa Pokrovskaya	New	Change initial view
5	Christie Church	Committed	Secure sign-in
6	Johnnie McLeod	Approved	Welcome back page
7	Christie Church	Committed	Cancel order form

To choose a specific team backlog, open the selector and select a different team or choose the **Browse all team backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



2. Choose the star icon to favorite a team backlog. Favorited artifacts ( favorited icon) appear on your **Favorites** page and towards the top of the team backlog selector menu.

## Favorite a shared query

Open **Boards > Queries** and choose the **All** page. Expand a folder as needed. Choose the star icon next to the query you want to favorite.

Or, open the context menu of the query, and then select **Add to Team Favorites**, and then select from the list of teams.

### NOTE

You must be a member of at least one team for the **Add to Team Favorites** option to be visible. If not visible, ask your project administrator or team administrator to add you to a team.

## Queries

Favorites All + New query Filter by keywords ↗

Title

> My Queries

Shared Queries

Current Sprint

Blocked Tasks

K Kathryn updated 7/12/2018

Open Impediments

Test Cases

Unfinished Work

Work in Progress

Triage folder

All items

All items in a tree query

Feedback

Run query

Edit

Rename

Delete

Add to Team Favorites

Customer Service

Security...

Manage Tags

Phone

You can also set a query as a personal favorite by opening the query and choosing the star icon.



Open **Work>Queries**. Next, open the actions icon menu of the shared query you want to favorite, and then select **Add to my favorites** or **Add to team favorites**.

Fabrikam Fiber Dashboards Code Work Build & Release Test

Backlogs Queries Plans Product Vision Tags

New

Manage Tags

Run

Edit

Delete

Rename

Add to my favorites

Add to team favorites

Security...

Add to dashboard

Blocked Tasks

Assigned to me

Results Editor Charts

Save query

ID ↑ Work Item Type

358 Product Backlog Item ...

360 Product Backlog Item

361 Product Backlog Item

362 Product Backlog Item

363 Product Backlog Item

364 Bug

PRODUCT BACKLOG ITEM 358

## Favorite a delivery plan

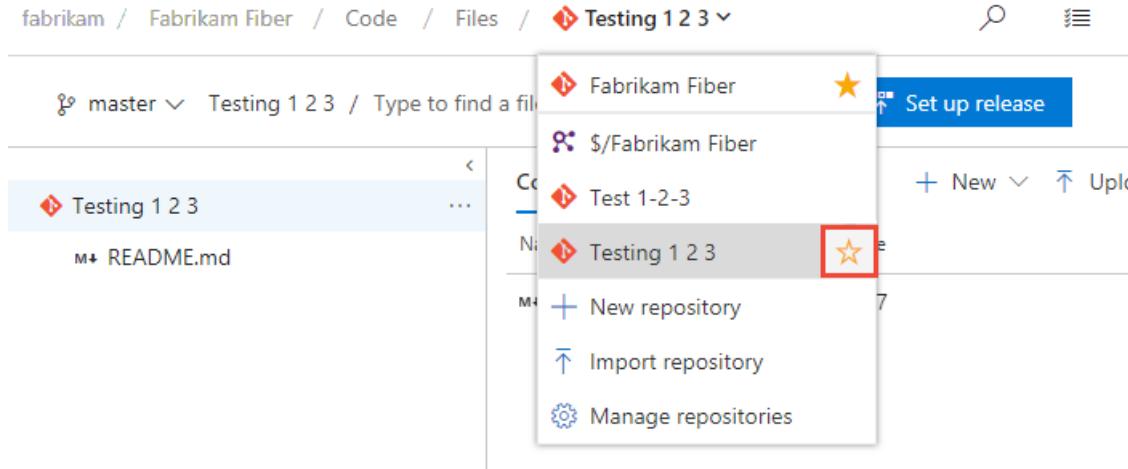
To learn more about delivery plans, see [Review team Delivery Plans](#).

To mark a delivery plan as a favorite, open the **Boards > Plans** page and choose the star icon next to the Delivery Plan.

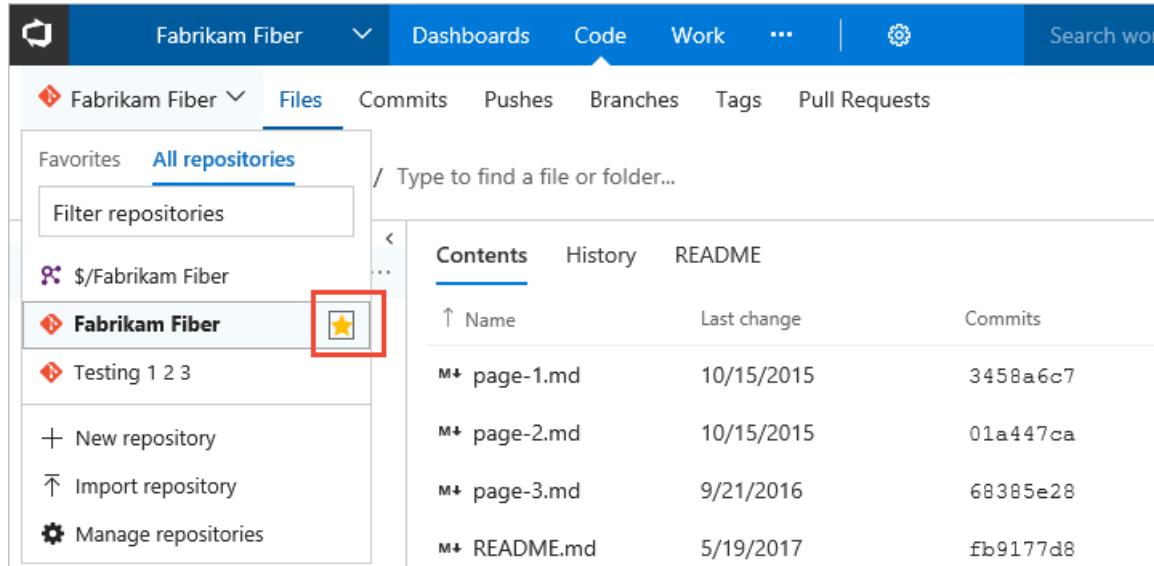
To mark a delivery plan as a favorite, open the **Work > Plans** page and choose the star icon next to the Delivery Plan.

## Favorite a repository

From any **Repos** page, open the repository selector and choose the star icon for the repository you want to favorite.



From any **Code** page, open the repository selector and choose the star icon next to the repository you want to favorite.



## Favorite a build pipeline

Open **Pipelines > Builds** and choose either **Mine** or **Definitions** page. Choose the star icon next to the build definition you want to favorite. Or, open the context menu of the build definition, and then select **Add to my favorites** or **Add to team favorites**.

Build Definitions

Build ID or build number  + New + Import

Mine Definitions Queued XAML

Recently built Status Triggered by History

The screenshot shows the 'Build Definitions' page in the Azure DevOps interface. At the top, there's a search bar for 'Build ID or build number' and buttons for '+ New' and '+ Import'. Below the search bar are tabs for 'Mine', 'Definitions', 'Queued', and 'XAML'. A horizontal row shows 'Recently built', 'Status', 'Triggered by', and 'History'. Under 'Recently built', there are two items: 'fabrikam build' and 'Fabrikam Fiber-CI'. For 'fabrikam build', a context menu is open, with the 'Add to my favorites' option highlighted by a red box. Other options in the menu include 'Queue new build...', 'Edit definition', 'Pause', 'View builds', 'Clone...', 'Export', 'Rename...', 'Save as a template...', 'Delete definition', 'Security...', and 'Add to dashboard'.

Open **Build and Release>Builds** and choose either **Mine** or **Definitions** page. Choose the star icon next to the build definition you want to favorite. Or, open the context menu of the build definition, and then select **Add to my favorites** or **Add to team favorites**.

Build Definitions

Build ID or build number

Mine All Definitions Queued XAML

Recently built Status Triggered by

The screenshot shows the 'Build Definitions' page with a selected build definition 'fabrikam build'. The context menu for this definition is open, with the 'Add to my favorites' option highlighted by a red box. The menu includes options like 'Queue new build...', 'Edit...', 'View definition summary', 'Clone...', 'Export', 'Rename...', 'Save as a template...', 'Delete definition', and 'Security...'.

## Favorite a test plan

To learn more about test plans, see [Create a test plan and test suite](#).

To mark a test plan as a favorite, open **Test Plans > Test Plans** and choose the star icon next to a test plan from the menu that shows All test plans.

To mark a test plan as a favorite, open the **Test > Test Plans** page and choose the star icon next to a test plan from the menu that shows All test plans.

## Unfavorite a view you've favorited

You can unfavorite an artifact from your **Favorites** page. Choose the inbox icon, and then choose **Favorites**. Choose the favorited icon of a currently favorited artifact.

The screenshot shows the 'Favorites' page in the Azure DevOps interface. At the top, there's a navigation bar with icons for search, inbox (highlighted with a red box), backlog, and profile. Below the navigation, there are tabs for 'Work Items', 'Pull requests', and 'Favorites' (also highlighted with a red box). The main content area is organized into sections: 'Projects', 'Teams', 'Dashboards', 'Plans', and 'Queries'. Each section lists artifacts with their names and a yellow star icon to the right, indicating they are favorited. The 'Favorites' tab is selected, and the star icons are consistently yellow across all listed items.

Category	Artifact	Status
Projects	Fabrikam Fiber	★
	Phone	★
	Voice	★
	Web	★
Dashboards	Fabrikam Fiber Team Analytics	★
	Backlog team plans	★
Plans	Fabrikam Fiber Feature plans	★
	All items	★
	All items on all projects	★
Queries	Assigned to me	★

Similarly, you can unfavorite an artifact from the same page where you favorited it.

You can unfavorite an artifact from the **Projects > Favorites** page and choose the favorited icon of a currently favorited artifact.

Similarly, you can unfavorite an artifact from the same page where you favorited it.

## Try this next

[Follow a user story, bug, issue, or other work item or pull request](#)

## Related articles

- [Manage personal notifications](#)
- [Set your preferences](#)

# Filter lists, boards, and directories

12/13/2022 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Several applications and pages support filtering, which is very useful when a large number of artifacts or items have been defined. Most directory views provide one or more filter functions.

You can filter most items using keywords or a user name for an author of an item or where work is assigned to them. You can filter lists and boards in the following areas:

- Git repositories: Branches, Commits, Commit history, Pull Requests, Pushes, and Repositories
  - Work tracking: Work Items, Kanban boards, Backlogs, Sprint Backlogs, and Taskboards
  - Directories: Dashboards, Boards, Backlogs, Sprints, Queries, Builds, Releases
- 
- Git repositories: Branches, Commits, Commit history, Pull Requests, Pushes, and Repositories
  - Work tracking: Work Items, Kanban boards, Backlogs, Sprint Backlogs, and Taskboards

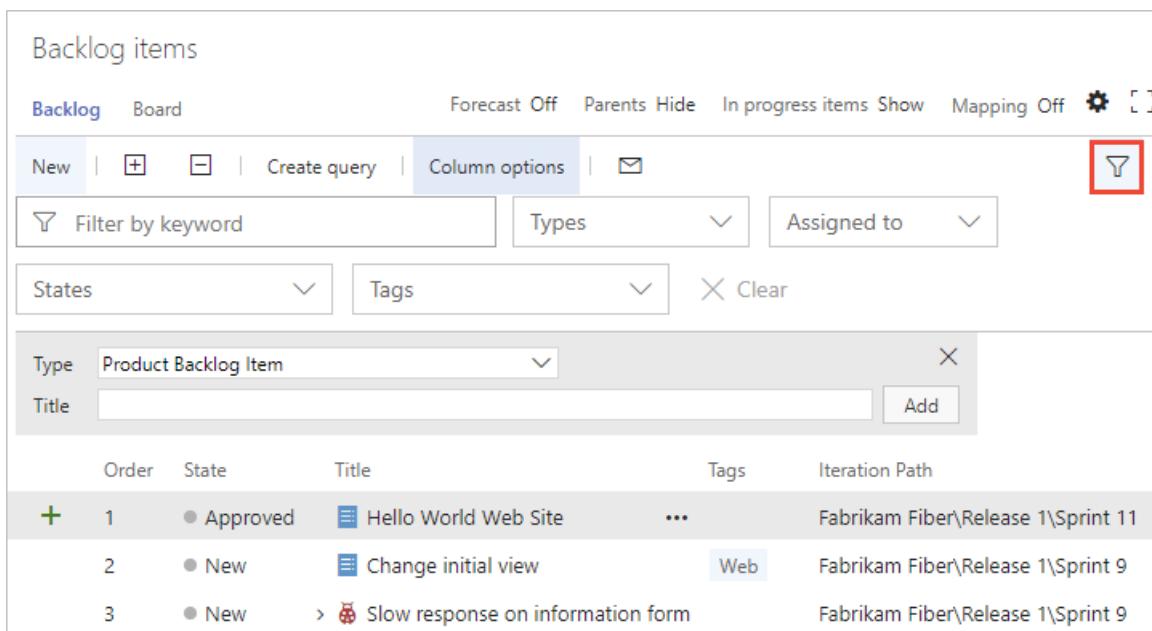
## NOTE

You may have fewer or additional filter options based on the [features you've enabled](#) or the platform and version that you are working from.

## Filter based on keywords, tags, or fields

To turn filtering on, choose the  filter icon.

You can filter work items by typing a keyword or using one or more of the fields provided, such as work item type, assigned to, state, and tags. Based on the keyword that you enter, the filter function will list work items based on any visible/displayed column or field, including tags. Also, you can enter a value for an ID, whether or not the ID field is visible.



The screenshot shows the 'Backlog items' board in Azure DevOps. At the top, there's a toolbar with 'Backlog' selected, followed by 'Board', 'Forecast Off', 'Parents Hide', 'In progress items Show', 'Mapping Off', and a gear icon. Below the toolbar is a row of buttons: 'New', '+', '[-]', 'Create query', 'Column options', and a filter icon (magnifying glass) which is highlighted with a red box. To the right of these buttons are dropdowns for 'Types' and 'Assigned to'. Further down are dropdowns for 'States' and 'Tags', and a 'Clear' button. A modal window is open at the bottom left, titled 'Type', showing 'Product Backlog Item' selected in a dropdown. There's also a 'Title' input field and an 'Add' button. The main board area displays a table with columns: Order, State, Title, Tags, and Iteration Path. Three backlog items are listed:

Order	State	Title	Tags	Iteration Path
1	Approved	Hello World Web Site	...	Fabrikam Fiber\Release 1\Sprint 11
2	New	Change initial view	Web	Fabrikam Fiber\Release 1\Sprint 9
3	New	Slow response on information form		Fabrikam Fiber\Release 1\Sprint 9

The filtered set is always a flat list, even if you've selected to show parents.

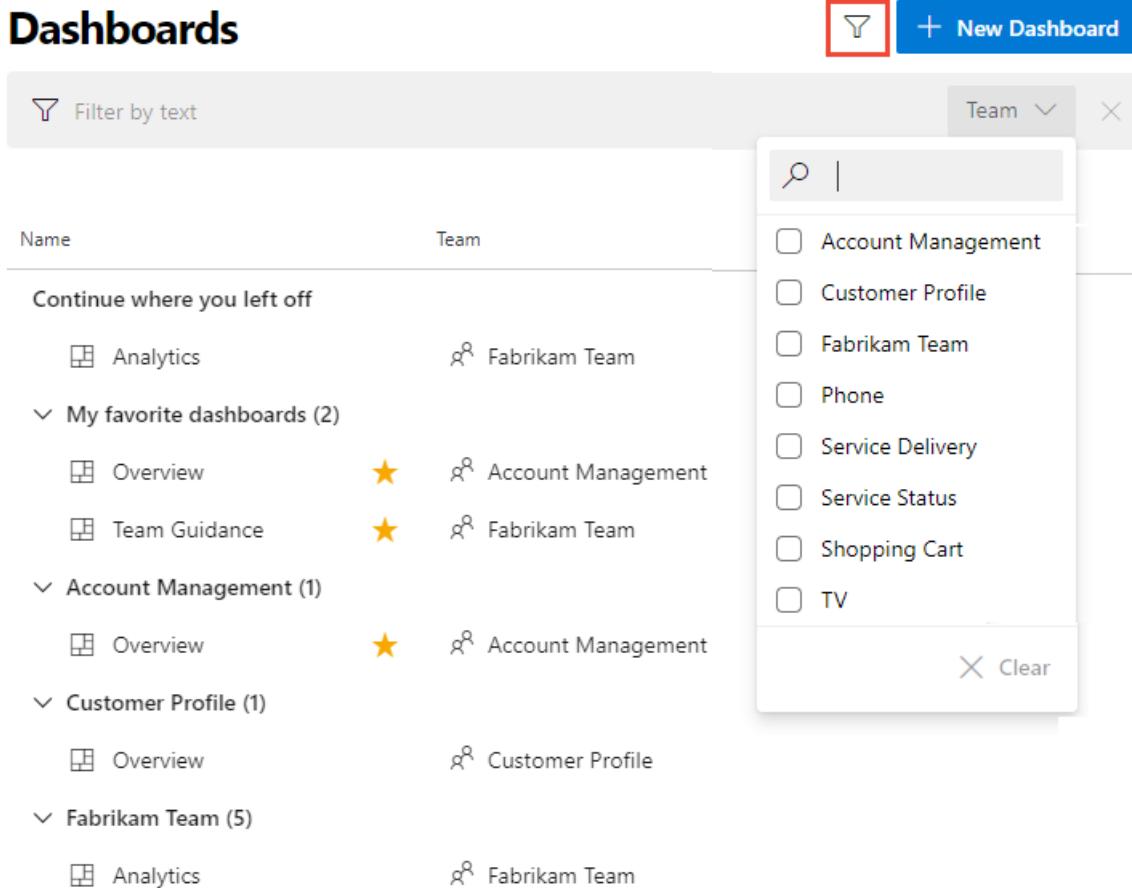
## Characters ignored by keyword filter criteria

The filter criteria ignores the following characters: `,` (comma), `.` (period), `/` (forward slash), and `\` (back slash).

## Filter directories

Choose the  filter icon to filter a directory list by keyword, team, or other supported field. Keywords apply to titles, descriptions, and team names.

For example, here we turn filtering on for the dashboard directory.



The screenshot shows the 'Dashboards' directory list. At the top right, there is a 'Filter by text' button with a magnifying glass icon, which is highlighted with a red box. Below it is a search bar with the placeholder 'Team'. A dropdown menu is open, showing a list of teams with checkboxes: Account Management, Customer Profile, Fabrikam Team, Phone, Service Delivery, Service Status, Shopping Cart, and TV. At the bottom right of the dropdown is a 'Clear' button.

Name	Team
Continue where you left off	
Analytics	Fabrikam Team
My favorite dashboards (2)	
Overview	Account Management
Team Guidance	Fabrikam Team
Account Management (1)	
Overview	Account Management
Customer Profile (1)	
Overview	Customer Profile
Fabrikam Team (5)	
Analytics	Fabrikam Team

## Related articles

- [Commit history](#)
- [Working with Git tags](#)
- [Filter backlogs and queries](#)
- [Filter your Kanban board](#)
- [Add tags to work items](#)

# Get started with search

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can quickly find work items, code files, wiki pages, or packages based on a keyword, wildcards, and other supported search filters with the search function.

Take an at-a-glance look at all of the [search features](#) further in this article.

## Prerequisites

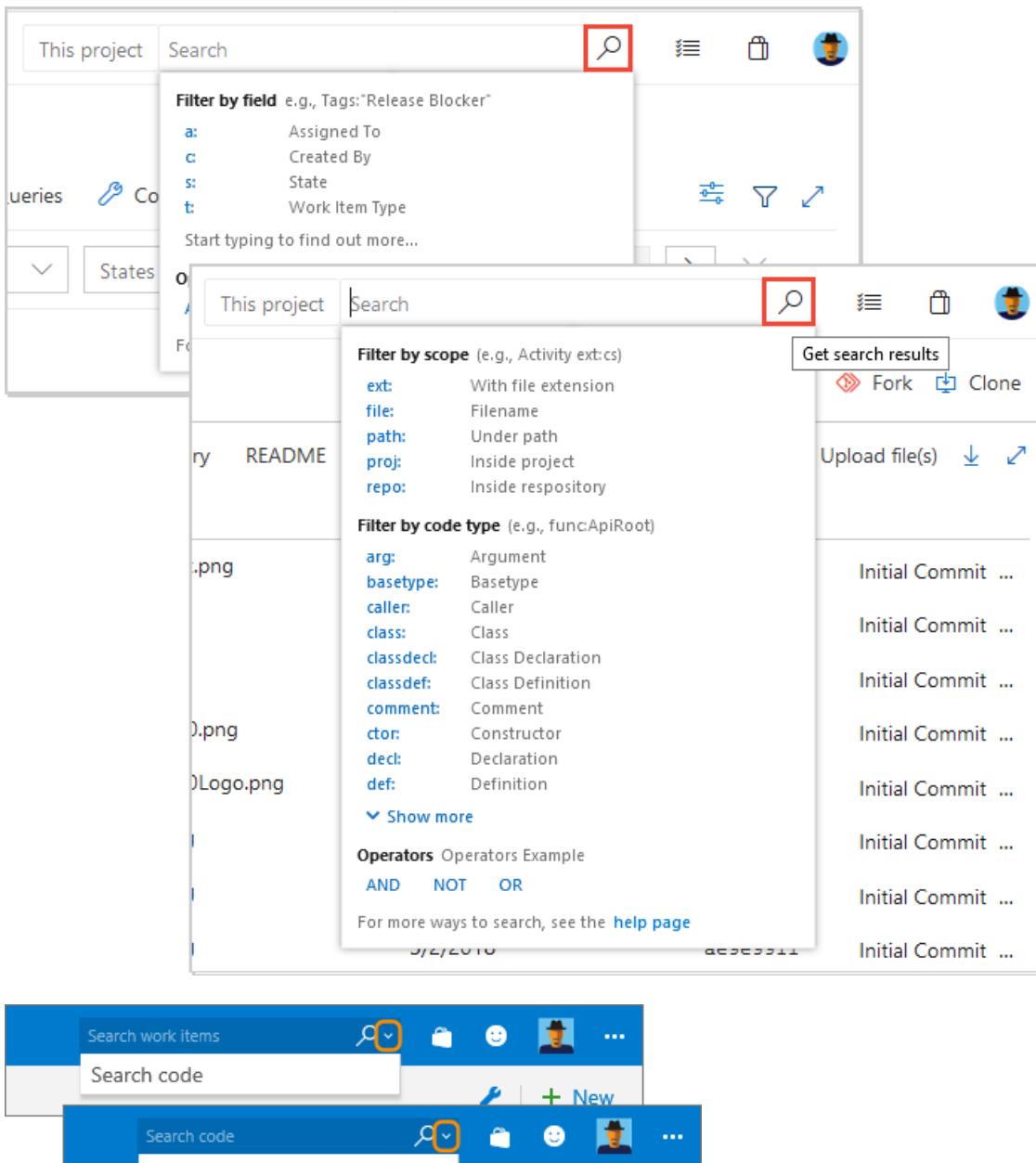
- Every project member can use the search functions, including project members granted Stakeholder, Basic, and higher levels of access.
- When searching across the organization or collection, only results for which a project member has access are listed.
- Stakeholder wiki search results are limited to provisioned wikis. Because published wikis require access to regular repositories, which Stakeholders don't have access to, results for published wikis don't appear in the search results. Similarly, Code search results don't appear for Stakeholders.

### IMPORTANT

For Code search, a Collection Administrator must [Install and configure search](#).

## Start your search with a keyword

Start your search using a keyword. You can then apply other options, as needed, to broaden or narrow your search results.



- If you get no results matching the input, try removing filters and retry the search. Broadening the search and after you view the search results, you can apply appropriate filters again and search again for relevant results.
- Check for the spelling of your search terms. Currently Work item search doesn't support ignoring of users' spelling mistakes.
- If there are lots of hits when you're using a wildcard search, such as when you're using a simple wildcard search string, you may see a message that no matching files are found. In this case, narrow your search to reduce the number of matches. Specify more characters of the word or words that you want to find, or add a condition or filter to limit the number of possible matches.
- Searches aren't case-sensitive.

## Search features, usage, and examples

The following features apply to all searches, including work items, code, wikis, and packages.

The following features apply to all searches, including work items, code, and packages.

---

### Search feature

## Usage

### Example

---

#### Keyword

Search based on one or more keywords.

`validate` finds instances that contain the word *validate*.

---

#### Exact match

Search based on an exact match, enclosed in double-quotes.

`"Client not found"` finds instances that contain the exact phrase match *Client not found*.

---

#### Wildcard

- Add wildcard characters, `*` and `?`, to keywords to extend the search criteria.
- Add `*` at the end of a keyword to find items that start with the keyword.
- Add `?` in the middle to represent any alphanumeric character.
- Use wildcard characters anywhere in your search string except as a prefix. You can use prefix wildcards with the other search filter functions.
- You can use more than one wildcard to match more than one character.
- `alpha?version` finds instances of alpha1version and alphaXversion.
- `Browser*` finds instances of BrowserEdge, BrowserIE, and BrowserFirefox.
- `CodeSenseHttp*` finds files containing words that start with *CodeSenseHttp*, such as `CodeSenseHttpClient` and `CodeSenseHttpClientTest`.

---

#### Boolean operators

- Find two or more keywords using Boolean operators: `AND`, `OR`, and `NOT` (must be uppercase).
- Add parenthesis to clauses to support logical groupings.
- Because `AND` is the default operator, an entry of two keywords with no operator is the same as an `AND` search.
- `Validate AND revisit` finds files that contain both the words *validate* and *revisit*.
- `Validate OR revisit` finds files that contain either of the words *validate* or *revisit*.
- `Validate NOT revisit` finds files that contain the word *validate* but not the word *revisit*.
- `(Validate NOT revisit) OR "release delayed"` finds files that contain the word *validate* but not the word *revisit* or files that contain the phrase *release delayed*.

---

#### Proximity

- Search for files based on vicinity using proximity operators: NEAR, BEFORE, and AFTER (must be uppercase).
- By default, proximity search looks for terms within five tokens distance.
- `term1 BEFORE term2` returns all files where term1 occurs BEFORE term2 within a distance of five tokens between them.
- `term1 AFTER term2` returns the same results as `term2 BEFORE term1`.
- `term1 NEAR term2` returns all files where term1 is within five token distance from term2 in any direction.  
`term1 NEAR term2` returns the same results as `term1 BEFORE term2 OR term2 BEFORE term1`.

## Special characters

- Escape the special characters `(`, `)`, `[`, `]`, `:`, `*`, and `?` by enclosing them in a phrase delimited with double-quotes.
- Include special characters in a search string, or search specifically for special characters, according to the following rules:
  - CodeA23?R finds files containing words that start with CodeA23
  - Have any alphanumeric character next, and end with R. For example, CodeA234R and CodeA23QR.
  - Search for any special character that isn't a part of the query language.
- `"flatten()"` finds the literal string *flatten()*. Search for a literal occurrence of the double-quote character " by preceding it with the escape character `\` and enclosing the search string in double-quotes.
- `"\"react-redux\""` finds the literal string "react-redux".

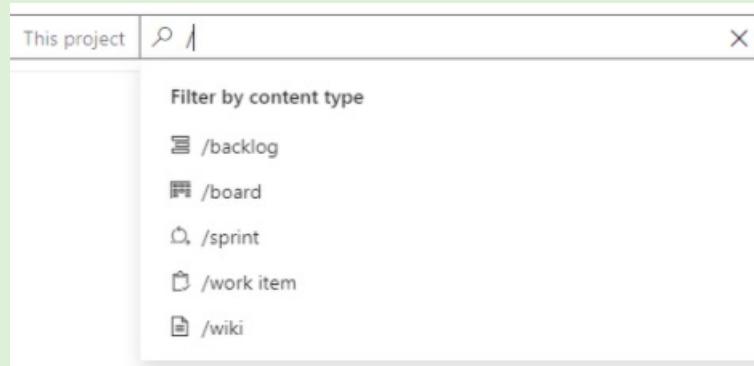
## Search from a different page

You can search from any of the following pages:

- The Projects page for the organization: starts a search across all projects.
- The Project overview page: automatically applies a filter to search within the selected project.
- The Boards page for a project: automatically displays recent work items and backlogs accessed by the user.
- Azure Repos, Pipelines, Test Plans, or an Artifacts page for a project: automatically displays functional filters for code searches.
- The wiki page for a project: automatically go to a wiki page you recently opened.

### TIP

Use the content type filter to access a page that you recently accessed.



For more information about searching and filtering in Azure Boards, see [Filter backlogs, boards, and plans](#).

For more information about searching wikis, see [Provisioned vs. published wiki](#).

### TIP

**No results found for ...**

If there's a large number of hits when using a wildcard search, such as when using a very simple wildcard search string, you may see a message that no matching files were found. In this case, narrow your search to reduce the number of matches. For example, specify more characters of the word(s) you want to find, or add a condition or filter to limit the number of possible matches.

## Additional search functions

To search for various settings, users, projects, and more, see the following table for other types of search tasks and corresponding actions.

### Search task

#### Action

##### [Find an organization setting](#)

Go to your organization and select **Organization settings**.

##### [Find a project setting](#)

Go to your project and select **Project settings**.

##### [Find a user setting](#)

Go to your **User settings page**.

##### Find a user

Go to your organization and select **Organization settings > Users**, and then enter the name in the filter box.

##### Find an organization

Scroll through the left side of your screen, which lists all organizations.

##### Find a project

Go to your organization, and then enter the project name in the Filter projects box.

##### View file history and compare versions

Go to **Repos > Files**, highlight your file, and then select **History**.

#### NOTE

When you search from the **Organization settings** page, your search results include both organization-level and project-level settings.

## Search re-index requirements

Search for Azure DevOps Server has the following limitation:

- If you do a disaster recovery (DR) operation and move your server back to an earlier snapshot of your SQL database, [re-index all your collections](#).

## Marketplace extensions

- [Code search](#) - Extends search with fast, flexible, and precise search results across all your code. Required for searching repositories.
- [Azure Paths Search](#) - Adds a special search hub to Boards for searching within iterations and area paths without having to create and maintain custom queries.

**NOTE**

Some extensions aren't supported features of Azure DevOps and therefore aren't supported by the product team. For questions, suggestions, or issues you have when using these extensions, visit their corresponding extension page on the [Visual Studio Marketplace](#).

## Next steps

[Functional work item search](#) or [Functional code search](#) or [Functional artifact or package search](#)

## Related articles

- [Code search blog posts](#)
- [Work item search blog posts](#)

# Manage or enable features

12/13/2022 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 | Azure DevOps Server 2020

As new features are introduced, you can turn them on or off. That way, you can try them out, provide feedback, and work with those features that meet your requirements.

Some preview features provide access to entire new functionality. Others, such as the New Wiki experience, reflect a change to the user interface, but little or no change in functionality.

## NOTE

It may take up to three weeks after a release to Azure DevOps Services for the preview feature to appear in your organization. The [latest release notes](#) usually provide information on new preview features. You can turn on or off select features for Azure DevOps Services. Preview features become available first on Azure DevOps Services and then become standard features with an update to Azure DevOps Server. At some point, the preview feature moves out of preview status and becomes a regular feature of the web portal.

There are a few features you or an administrator can enable or disable. Some features provide access to entire new functionality, while others provide a change to the user interface.

The follow table indicates which preview features can be enabled per user or team member, and those that can be enabled for the organization. You must be a member of the Project Collection Administrators group to change a preview feature at the organization-level.

PREVIEW FEATURES	PER USER	PER ORGANIZATION
<a href="#">Analytics Views</a>	✓	✓
<a href="#">Copy Dashboard Experience</a>	✓	✓
Dependency Tracker Preview Features (ignore this setting)	✓	✓
<a href="#">Experimental themes</a>	✓	✓
<a href="#">Full Access to Azure Pipelines for Stakeholders</a>		✓
<a href="#">Limit user visibility and collaboration to specific projects</a>		✓
<a href="#">New account manager</a>	✓	✓
<a href="#">New Artifacts (Feeds) Experience (accessibility updates)</a>	✓	✓
<a href="#">New Boards Hubs</a>	✓	✓
<a href="#">New boards reports</a>	✓	✓

PREVIEW FEATURES	PER USER	PER ORGANIZATION
New release progress views	✓	✓
New service connections experience	✓	✓
New Settings Search in the organization settings panel	✓	✓
New Teams page	✓	✓
New Wiki experience	✓	✓
Organization Permissions Settings Page v2	✓	✓
Project Permissions Settings page	✓	✓
Task Insights for Failed Pipeline Runs	✓	✓
YAML templates editor	✓	✓

The following table indicates those features that you can enable as a user, project administrator, or project collection administrator. These preview features are only available to manage for Azure DevOps Server 2020 RTW.

FEATURE	USER	PROJECT	COLLECTION
New service connections experience	✓		✓
Selective artifacts download feature for collection/project		✓	✓

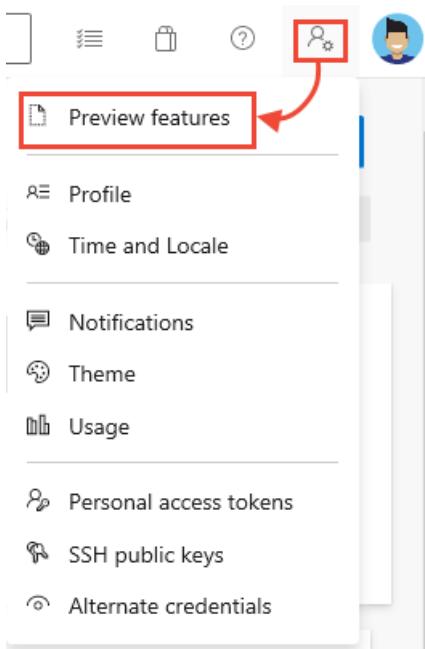
## Enable features for your use

From time to time, a new feature is introduced in Preview mode, which allows you to turn it on or off.

To access the Preview features options, open your profile menu. The profile menu appears as shown below based on whether the **New Account Manager** feature has been enabled or not. The **New Account Manager** preview feature turns on enhanced user interface options for managing account information. The menu options move under the **User settings** icon from where they were previously under the **Account manager for your account** icon.

- [New Account Manager enabled](#)
- [New Account Manager not enabled](#)

Choose  [User settings](#), and then choose [Preview features](#).



To enable or disable a feature, choose the slider.

## Preview features

X

The following preview features are available for your evaluation. Help us make them better!

for me [Jamal Hartnett]

### Analytics Views

On

Enable a hub for creating data sets to use for custom reporting.

### Copy Dashboard Experience

On

Exposes copy dashboard experience on a dashboard.

### Dependency Tracker Preview Features

On

Enable Dependency Tracker preview features

### Experimental Themes

On

Adds an early preview of various themes to the Theme management panel.

### New account manager

On

Enables the new account manager which allows users to switch accounts easily and have quicker access to help menu options from the top navigation bar

### New Artifacts (Feeds) Experience

On

Enable the updated Artifacts (Feed) UI experience

### New Boards Hubs

On

Enables new web platform Boards hubs.

### New boards reports

On

Turn on interactive reports in boards pages, replacing CFD, Velocity and Burndown charts in boards headers.

### New release progress views

On

Turn on the new release views to visualize the progress of your deployment pipelines. [Learn more](#)

### New service connections experience

On

Turns on the new service connections experience.

### New Settings Search in the organization settings panel

On

Enables settings search

### New Teams Page

On

Enables new teams page

### New Wiki Experience

On

Enables new wiki experience in the wiki hub.

### Organization Permissions Settings Page v2

On

Lights up version 2 of the permissions tab

### Project Permissions Settings page

On

Lights up new project permissions hub

### Organization Permissions Settings Page v2

On

Lights up version 2 of the permissions tab

### Project Permissions Settings page

On

Lights up new project permissions hub

### Task Insights for Failed Pipeline Runs

On

Enables insights toast for the failed tasks in Pipeline Run.

### YAML templates editor

On

New YAML editor with support for editing templates.

For information on other user settings and preferences, see [Set user preferences](#).

## Enable features at the organization level

When you enable a feature at the organization level, you essentially turn it on for all users of your account. Each user can then disable the feature if they so choose. If you disable a feature at the organization level, user settings are not changed. Users can enable or disable the feature on their own.

### TIP

If you don't see the **for this account** menu option, then you aren't a member of the Project Collection Administrators group. To get added as one, see [Change project collection-level permissions](#).

## Preview features

X

The following preview features are available for your evaluation. Help us make them better!

for this account [fabrikam]

### Analytics Views

Off

Enable a hub for creating data sets to use for custom reporting.

### Copy Dashboard Experience

On

Exposes copy dashboard experience on a dashboard.

### Experimental Themes

Off

Adds an early preview of various themes to the Theme management panel.

### Full access to Azure Pipelines for Stakeholders

On

Gives users with the Stakeholder license full access to Azure Pipelines for private projects. Limit what they can do by using security groups and permissions. Turning on this feature doesn't affect public projects, where Stakeholders always have full access. [Learn more](#)

### Limit user visibility and collaboration to specific projects

On

Enable organization administrators in Azure DevOps to restrict users from seeing and collaborating with users in different projects

### New account manager

On

Enables the new account manager which allows users to switch accounts easily and have quicker access to help menu options from the top navigation bar

### New Artifacts (Feeds) Experience

On

Enable the updated Artifacts (Feed) UI experience

### New Boards Hubs

Off

Enables new web platform Boards hubs.

### New boards reports

On

Turn on interactive reports in boards pages, replacing CFD, Velocity and Burndown charts in boards headers.

### New release progress views

On

Turn on the new release views to visualize the progress of your deployment pipelines. [Learn more](#)

### New service connections experience

On

Turns on the new service connections experience.

### New Settings Search in the organization settings panel

On

Enables settings search

### New Teams Page

On

Enables new teams page

### New Wiki Experience

On

Enables new wiki experience in the wiki hub.

### Organization Permissions Settings Page v2

On

Lights up version 2 of the permissions tab

### Project Permissions Settings page

On

Lights up new project permissions hub

### Task Insights for Failed Pipeline Runs

On

Enables insights toast for the failed tasks in Pipeline Run.

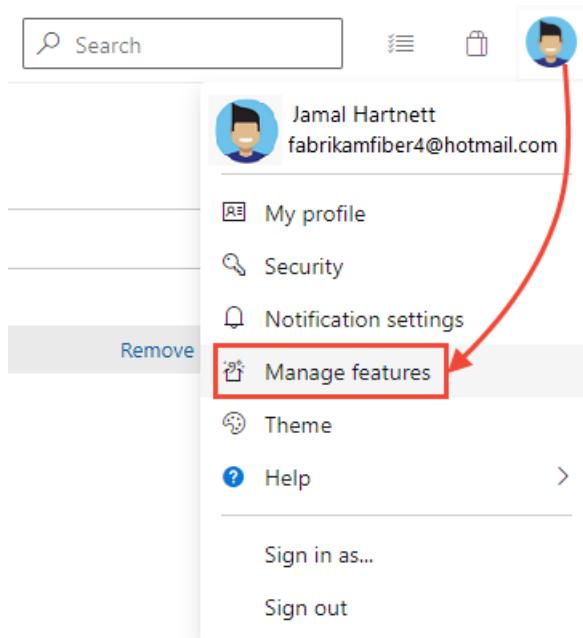
### YAML templates editor

On

New YAML editor with support for editing templates.

## Enable or disable a feature

1. Open your profile menu by choosing your image icon and select **Manage features**.



2. Select the level from the menu provided.

**TIP**

If you don't see the **for this project** or **for this collection** menu options, then you aren't an administrator. To get added as one, see [Change project collection-level permissions](#).

3. To enable or disable a feature, choose the slider.

**User-level**

**Manage features** X

The following features are optional and can be turned on or off.

for me [Jamal.Hartnett] ▼

New service connections experience  On

Turns on the new service connections experience.

**Project-level**

**Manage features** X

The following features are optional and can be turned on or off.

for this project [FabrikamBasic] ▼

Selective artifacts download feature for collection/project  Off

Turns on selective artifacts download feature in release pipelines.  
This feature isn't supported if you have agents behind proxy in your release pipelines.

**Collection-level**

## Manage features



The following features are optional and can be turned on or off.

for this collection [DefaultCollection] ▼

### New service connections experience

On

Turns on the new service connections experience.

### Selective artifacts download feature for collection/project

Off

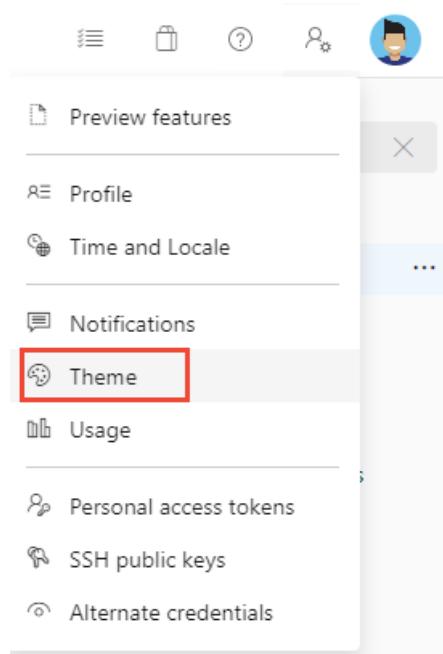
Turns on selective artifacts download feature in release pipelines.

This feature isn't supported if you have agents behind proxy in your release pipelines.

When you enable a feature at the project or collection-level, you essentially turn it on for all users. If you disable a feature at the project or collection-level, user settings are not changed. Users can enable or disable the feature on their own.

## Experimental themes

When you select **Theme** from the Profile menu you can select between **Dark** and **Light** themes for the display of Azure DevOps web portal.

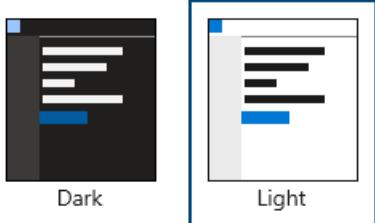


With **Experimental themes** enabled, you can select among a number of additional themes.

## Choose your theme

X

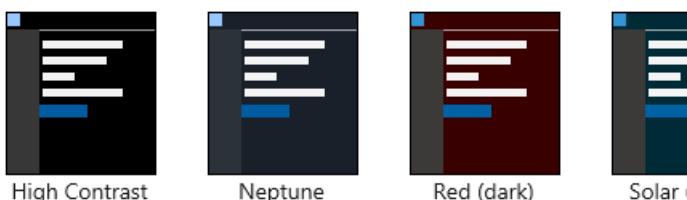
### Default themes



### Custom themes



### Preview themes



## Features now enabled for all Azure DevOps Services

### General

- [New user hub](#)
- [New PAT experience](#)
- [New Navigation](#)
- [Wiki](#)
- [Combine email recipients](#)
- [New experience in Code, Work Item, & Wiki search](#)
- [Out of the box notifications](#)
- [Team expansion for notifications](#)
- [Streamlined User Management](#)

### Azure Artifacts

- [NuGet.org upstream sources](#)
- [Updated package experience](#)

### Azure Boards, Dashboards, and Analytics

- [New Delivery Plans Experience](#)
- [Enable group by tags for work item chart widget on dashboard](#)
- [New Rich Text Editor](#)
- [New Queries Experience](#)
- [New Work Items](#)
- [New Dashboards Experience](#)

### Azure Repos

- [New TFVC pages](#)
- [Git Forks](#)
- [New Repos pull request experience](#)
- [New Repos settings experience](#)
- [New Repos landing pages](#)
- [Pull Request Status Policy](#)

### Azure Pipelines

- [Historical graph for agent pools](#)
- [Pipeline decorators](#)
- [Multi-stage pipelines](#)
- [Test tab in new web platform](#)
- [Test analytics in new web platform](#)
- [New builds hub](#)
- [Build with multiple queues](#)
- [New Releases Hub](#)
- [Approval gates in releases - New Release Definition Editor](#)
- [Symbol server](#)
- [Task tool installers](#)

### Azure Test Plans

- [New Test Plans Page](#)
- [New Test Plan Experience](#)

## Related articles

- [Set user preferences](#)
- [Azure DevOps Feature Timeline](#)

# Get started with search

12/13/2022 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

You can quickly find work items, code files, wiki pages, or packages based on a keyword, wildcards, and other supported search filters with the search function.

Take an at-a-glance look at all of the [search features](#) further in this article.

## Prerequisites

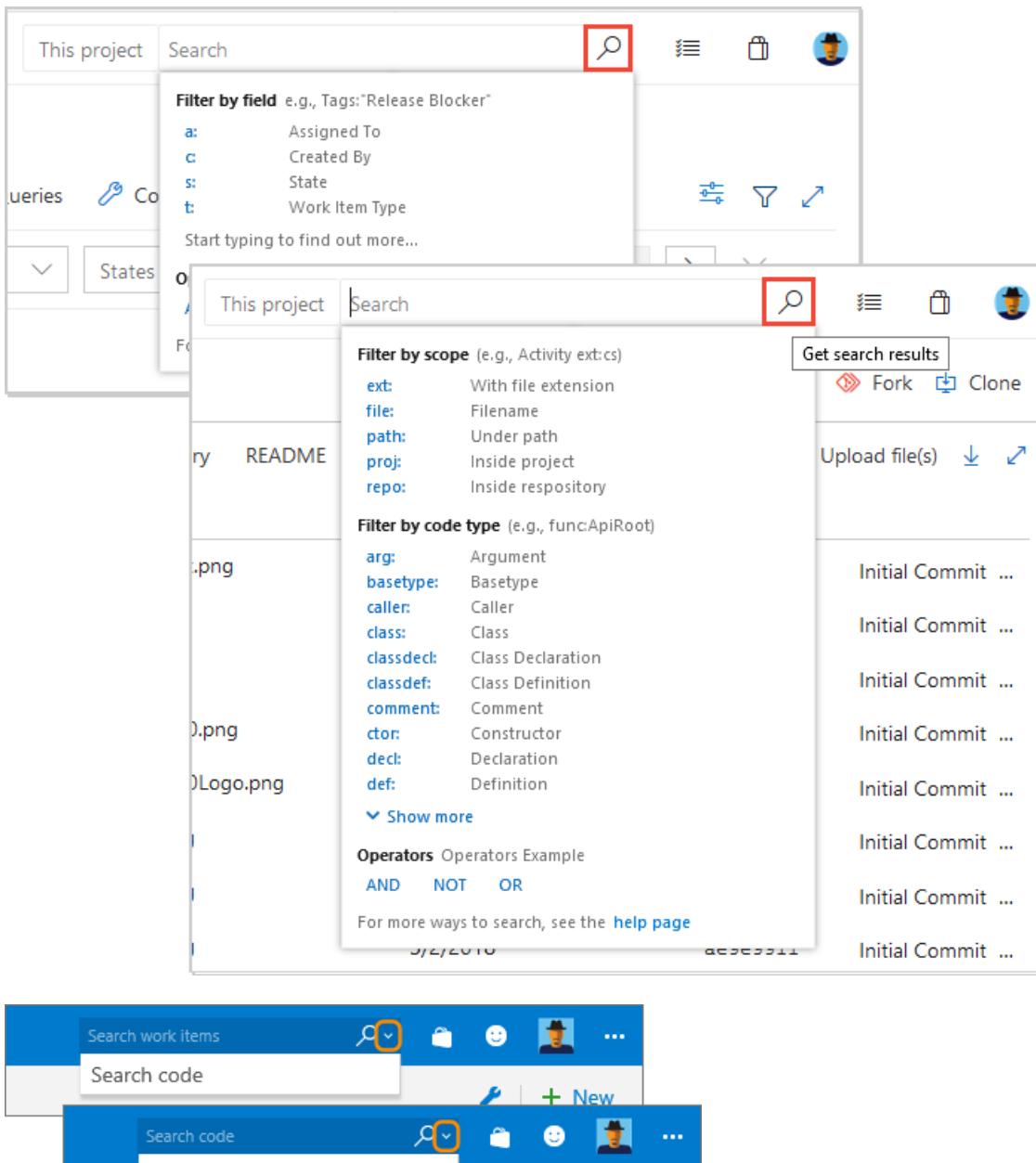
- Every project member can use the search functions, including project members granted Stakeholder, Basic, and higher levels of access.
- When searching across the organization or collection, only results for which a project member has access are listed.
- Stakeholder wiki search results are limited to provisioned wikis. Because published wikis require access to regular repositories, which Stakeholders don't have access to, results for published wikis don't appear in the search results. Similarly, Code search results don't appear for Stakeholders.

### IMPORTANT

For Code search, a Collection Administrator must [Install and configure search](#).

## Start your search with a keyword

Start your search using a keyword. You can then apply other options, as needed, to broaden or narrow your search results.



- If you get no results matching the input, try removing filters and retry the search. Broadening the search and after you view the search results, you can apply appropriate filters again and search again for relevant results.
- Check for the spelling of your search terms. Currently Work item search doesn't support ignoring of users' spelling mistakes.
- If there are lots of hits when you're using a wildcard search, such as when you're using a simple wildcard search string, you may see a message that no matching files are found. In this case, narrow your search to reduce the number of matches. Specify more characters of the word or words that you want to find, or add a condition or filter to limit the number of possible matches.
- Searches aren't case-sensitive.

## Search features, usage, and examples

The following features apply to all searches, including work items, code, wikis, and packages.

The following features apply to all searches, including work items, code, and packages.

---

### Search feature

## Usage

### Example

---

#### Keyword

Search based on one or more keywords.

`validate` finds instances that contain the word *validate*.

---

#### Exact match

Search based on an exact match, enclosed in double-quotes.

`"Client not found"` finds instances that contain the exact phrase match *Client not found*.

---

#### Wildcard

- Add wildcard characters, `*` and `?`, to keywords to extend the search criteria.
  - Add `*` at the end of a keyword to find items that start with the keyword.
  - Add `?` in the middle to represent any alphanumeric character.
  - Use wildcard characters anywhere in your search string except as a prefix. You can use prefix wildcards with the other search filter functions.
  - You can use more than one wildcard to match more than one character.
  - `alpha?version` finds instances of alpha1version and alphaXversion.
  - `Browser*` finds instances of BrowserEdge, BrowserIE, and BrowserFirefox.
  - `CodeSenseHttp*` finds files containing words that start with *CodeSenseHttp*, such as `CodeSenseHttpClient` and `CodeSenseHttpClientTest`.
- 

#### Boolean operators

- Find two or more keywords using Boolean operators: `AND`, `OR`, and `NOT` (must be uppercase).
  - Add parenthesis to clauses to support logical groupings.
  - Because `AND` is the default operator, an entry of two keywords with no operator is the same as an `AND` search.
  - `Validate AND revisit` finds files that contain both the words *validate* and *revisit*.
  - `Validate OR revisit` finds files that contain either of the words *validate* or *revisit*.
  - `Validate NOT revisit` finds files that contain the word *validate* but not the word *revisit*.
  - `(Validate NOT revisit) OR "release delayed"` finds files that contain the word *validate* but not the word *revisit* or files that contain the phrase *release delayed*.
- 

#### Proximity

- Search for files based on vicinity using proximity operators: NEAR, BEFORE, and AFTER (must be uppercase).
  - By default, proximity search looks for terms within five tokens distance.
  - `term1 BEFORE term2` returns all files where term1 occurs BEFORE term2 within a distance of five tokens between them.
  - `term1 AFTER term2` returns the same results as `term2 BEFORE term1`.
  - `term1 NEAR term2` returns all files where term1 is within five token distance from term2 in any direction.  
`term1 NEAR term2` returns the same results as `term1 BEFORE term2 OR term2 BEFORE term1`.
-

## Special characters

- Escape the special characters `(`, `)`, `[`, `]`, `:`, `*`, and `?` by enclosing them in a phrase delimited with double-quotes.
- Include special characters in a search string, or search specifically for special characters, according to the following rules:
  - CodeA23?R finds files containing words that start with CodeA23
  - Have any alphanumeric character next, and end with R. For example, CodeA234R and CodeA23QR.
  - Search for any special character that isn't a part of the query language.
- `"flatten()"` finds the literal string *flatten()*. Search for a literal occurrence of the double-quote character " by preceding it with the escape character `\` and enclosing the search string in double-quotes.
- `"\"react-redux\""` finds the literal string "react-redux".

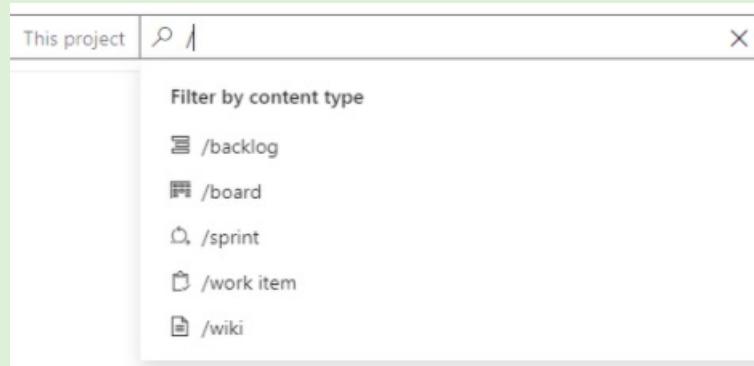
## Search from a different page

You can search from any of the following pages:

- The Projects page for the organization: starts a search across all projects.
- The Project overview page: automatically applies a filter to search within the selected project.
- The Boards page for a project: automatically displays recent work items and backlogs accessed by the user.
- Azure Repos, Pipelines, Test Plans, or an Artifacts page for a project: automatically displays functional filters for code searches.
- The wiki page for a project: automatically go to a wiki page you recently opened.

### TIP

Use the content type filter to access a page that you recently accessed.



For more information about searching and filtering in Azure Boards, see [Filter backlogs, boards, and plans](#).

For more information about searching wikis, see [Provisioned vs. published wiki](#).

### TIP

**No results found for ...**

If there's a large number of hits when using a wildcard search, such as when using a very simple wildcard search string, you may see a message that no matching files were found. In this case, narrow your search to reduce the number of matches. For example, specify more characters of the word(s) you want to find, or add a condition or filter to limit the number of possible matches.

## Additional search functions

To search for various settings, users, projects, and more, see the following table for other types of search tasks and corresponding actions.

### Search task

#### Action

##### [Find an organization setting](#)

Go to your organization and select **Organization settings**.

##### [Find a project setting](#)

Go to your project and select **Project settings**.

##### [Find a user setting](#)

Go to your **User settings page**.

##### Find a user

Go to your organization and select **Organization settings > Users**, and then enter the name in the filter box.

##### Find an organization

Scroll through the left side of your screen, which lists all organizations.

##### Find a project

Go to your organization, and then enter the project name in the Filter projects box.

##### View file history and compare versions

Go to **Repos > Files**, highlight your file, and then select **History**.

#### NOTE

When you search from the **Organization settings** page, your search results include both organization-level and project-level settings.

## Search re-index requirements

Search for Azure DevOps Server has the following limitation:

- If you do a disaster recovery (DR) operation and move your server back to an earlier snapshot of your SQL database, [re-index all your collections](#).

## Marketplace extensions

- [Code search](#) - Extends search with fast, flexible, and precise search results across all your code. Required for searching repositories.
- [Azure Paths Search](#) - Adds a special search hub to Boards for searching within iterations and area paths without having to create and maintain custom queries.

**NOTE**

Some extensions aren't supported features of Azure DevOps and therefore aren't supported by the product team. For questions, suggestions, or issues you have when using these extensions, visit their corresponding extension page on the [Visual Studio Marketplace](#).

## Next steps

[Functional work item search](#) or [Functional code search](#) or [Functional artifact or package search](#)

## Related articles

- [Code search blog posts](#)
- [Work item search blog posts](#)

# Functional code search

12/13/2022 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Functional code search extends your ability to refine your search across repositories beyond what's documented in [Get started with search](#). To do code searches, the [Code Search](#) Marketplace extension must be installed for your organization or collection.

## Prerequisites

- Install [Code Search](#)

For more information, see [Install and configure search](#).

- To use Code Search, you must have at least Basic access.
- Users with Stakeholder access don't have access to code, so they can't search for code.
- Users with Stakeholder access for a public project have [full access to code](#), so they can search for code. To access code in a private project, you must have at least Basic access.
- When you're searching across the organization or collection, only results for which a project member has access are listed.

## Code search best practices

- Get the results you want even faster by starting with a higher-level search. You can narrow your search by using project, repository, path, file name, and other filter operators.
- When you're not sure of the exact term you're looking for, [Use wildcards to widen your search](#) and [Boolean operators to fine-tune it](#).
- Find more information about an item of interest faster and with minimal efforts. When you find an item of interest, place the cursor on it and use the shortcut menu to quickly search for that text across all your projects and files.
- Easily trace how your code works by using the shortcut menu to search for related items such as definitions and references – directly from inside a file or from the search results.
- Go quickly to the implementation of, for example, an API your code might be taking dependency on by narrowing down your results to exact code type matches. Use code type filters to search for specific kinds of code such as:
  - definitions
  - references
  - functions
  - comments
  - strings
  - namespaces, and more.

### NOTE

You can't search code in forked repositories.

## Functions to find specific types of code

As you enter your search, select functions and keywords from the drop-down list to quickly create your query. Use the **Show more** link to display all the available functions and keywords. Mix and match the functions as required.

You can also select one or a combination of filters from the list in the left column. Again, the **Show more** link displays all the available functions and keywords.

Instead, you can enter the functions and parameters directly into the search. The following table shows a list of functions for selecting specific types or members in your C#, C, C++, Java, and Visual Basic.NET code.

TO FIND CODE WHERE <i>FINDTHIS</i> APPEARS AS A ...	... SEARCH FOR ARGUMENT ARG: <i>FINDTHIS</i>
Argument	<b>arg:</b> <i>findThis</i> Deprecated in July 2019
Base type	<b>basetype:</b> <i>findThis</i>
Calling function	<b>caller:</b> <i>findThis</i> Deprecated in July 2019
Class definition or declaration	<b>class:</b> <i>findThis</i>
Class declaration	<b>classdecl:</b> <i>findThis</i> Merged with class:
Class definition	<b>classdef:</b> <i>findThis</i> Merged with class:
Comment	<b>comment:</b> <i>findThis</i>
Constructor	<b>ctor:</b> <i>findThis</i> Merged with method:
Declaration	<b>decl:</b> <i>findThis</i>
Definition	<b>def:</b> <i>findThis</i>
Destructor	<b>dtor:</b> <i>findThis</i> Merged with method:
Enumerator	<b>enum:</b> <i>findThis</i>
Extern	<b>extern:</b> <i>findThis</i> Deprecated in July 2019
Field	<b>field:</b> <i>findThis</i>
Friend function	<b>friend:</b> <i>findThis</i> Deprecated in July 2019
Function	<b>func:</b> <i>findThis</i> Merged with method:
Function declaration	<b>funcdecl:</b> <i>findThis</i> Merged with method:
Function definition	<b>funcdef:</b> <i>findThis</i> Merged with method:
Global	<b>global:</b> <i>findThis</i> Deprecated in July 2019

TO FIND CODE WHERE <i>FINDTHIS</i> APPEARS AS A ...	... SEARCH FOR ARGUMENT ARG: <i>FINDTHIS</i>
Header	<b>header:</b> <i>findThis</i> Deprecated in July 2019
Interface	<b>interface:</b> <i>findThis</i>
Macro	<b>macro:</b> <i>findThis</i>
Macro definition	<b>macrodef:</b> <i>findThis</i> Merged with macro:
Macro reference	<b>macroref:</b> <i>findThis</i> Merged with macro:
Method	<b>method:</b> <i>findThis</i>
Method declaration	<b>methoddecl:</b> <i>findThis</i> Merged with method:
Method definition	<b>methoddef:</b> <i>findThis</i> Merged with method:
Namespace	<b>namespace:</b> <i>findThis</i>
Property	<b>prop:</b> <i>findThis</i>
Reference	<b>ref:</b> <i>findThis</i>
String literal	<b>strlit:</b> <i>findThis</i>
Struct	<b>struct:</b> <i>findThis</i> Merged with type:
Struct declaration	<b>structdecl:</b> <i>findThis</i> Merged with type:
Struct definition	<b>structdef:</b> <i>findThis</i> Merged with type:
Template argument	<b>tmplarg:</b> <i>findThis</i> Deprecated in July 2019
Template specification	<b>tmplspec:</b> <i>findThis</i> Deprecated in July 2019
Type	<b>type:</b> <i>findThis</i>
Typedef	<b>typedef:</b> <i>findThis</i> Merged with type:
Union	<b>union:</b> <i>findThis</i> Deprecated in July 2019

## Functions to select projects, repositories, paths, and files

Functions make it easy to narrow the search to specified locations, specific types of files within these locations, or specified filenames. Narrow the search to a specific location using the `proj`, `repo`, or `path` filters. Mix and match the functions as required.

USAGE	EXAMPLE
Find all occurrences of the word <i>QueueJobsNow</i> in the Fabrikam project.	<code>QueueJobsNow proj:Fabrikam</code>
Find all occurrences of the word <i>QueueJobsNow</i> in the Contoso repository.	<code>QueueJobsNow repo:Contoso</code>
Find all occurrences of the word <i>QueueJobsNow</i> in the path <i>VisualStudio/Services/Framework</i> and its subpaths.	<code>QueueJobsNow path:VisualStudio/Services/Framework</code>
Enclose the argument to the filter in double-quotes if it contains a space.	<code>QueueJobsNow path:"VisualStudio/Windows Phones and Devices/Services"</code>
Find all occurrences of the word <i>QueueJobsNow</i> in all files where the filename starts with <i>queueRegister</i> .	<code>QueueJobsNow file:queueRegister*</code>
Find all files with the name <i>QueueRegister</i> without an extension. Use quotes to find files without extensions.	<code>file:"queueRegister"</code>
Find all occurrences of the word <i>QueueJobsNow</i> in only C# source files. A plain text search string that doesn't include file type functions also finds files where the string matches part of the filename.	<code>QueueJobsNow ext:cs</code>

## Find related items or other terms

One of the powerful features of Code Search is the capability to expand your search interactively, based on the results of previous searches. For example, you can easily broaden your search to related files when tracing or debugging code.

Place the insertion point on a term in the file and open the shortcut menu (mouse: right-click) to start a new search for other files containing the selected term. You can search for it as text, for the definition if you select an object name, or for references to a selected object.

For more information about the following search functions, see [Get started with search](#).

- Keyword
- Exact match
- Wildcard
- Boolean operators
- Proximity

## More code search operations

See the following examples of even more code search functions. You can use the code type search functions with files written in C#, C, C++, Java, and Visual Basic.NET. Open the search results in a new browser tab from the main search box, and select **Ctrl + Enter**. In Google Chrome, select **Ctrl + Shift + Enter** to switch the focus to the new browser tab.

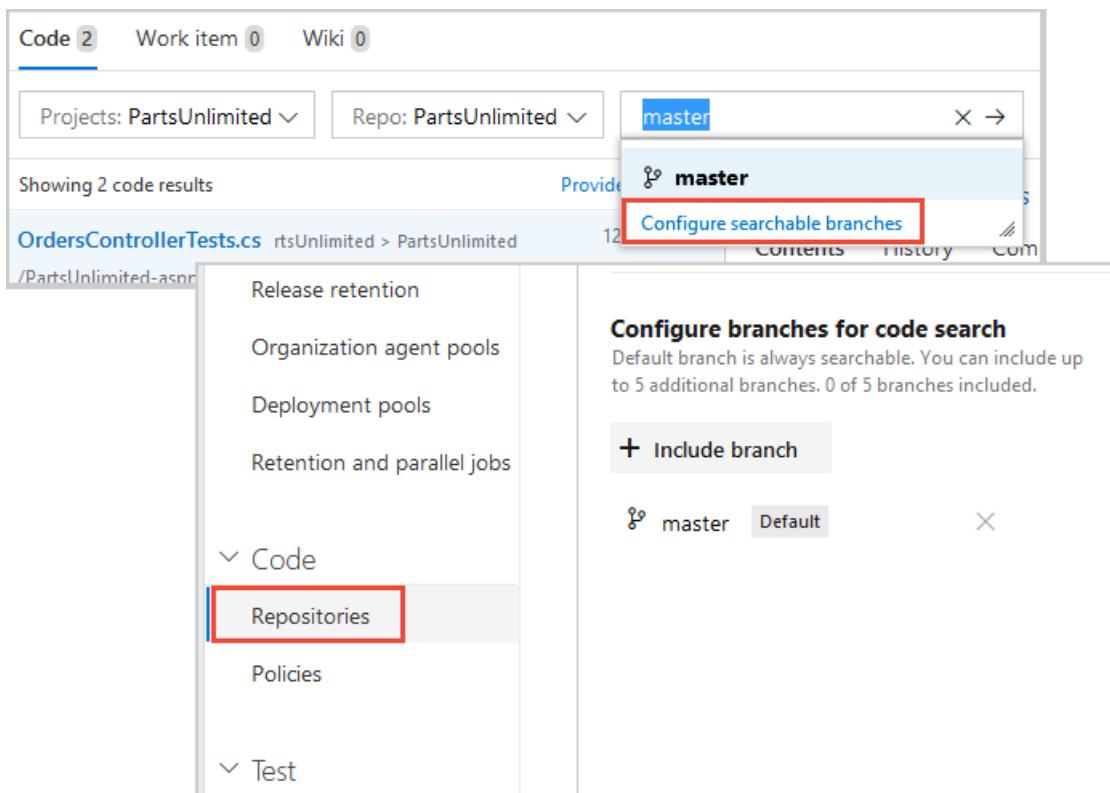
USAGE	EXAMPLE
Find all instances of "ToDo" comments in your code	Select <code>comment:</code> and enter <code>todo</code>

USAGE	EXAMPLE
Search in specific locations, such as within a particular path	Use a search string such as Driver path:MyShuttle/Server
Search for files by name or just by file extension	Driver file:GreenCabs.cs . The search string error ext:resx could be useful if you want to review all error strings in your code. Even if your plain text search string matches part of a filename, the file appears in the list of found files. This search works without matching specific file type functions.

## Search Git projects and repositories

In a Git project, you see a list of the repositories that it contains. Use the project and repository checkboxes to widen your search. You can search more or all projects, or narrow your search to fewer projects and repositories. If there are more than a few projects or repositories, use the **Show more** link to see them all.

Code Search can index multiple branches in a Git repository. By default it indexes files in only the default branch of your Git repositories. Your default branch is usually the **main** branch. Specify the branches for each repository, indexing in the Options tab of the **Repositories** section, [project settings page](#).



## Search TFVC projects

In a TFVC project, you see a list of folder paths in that project for which you have read access - you won't see any projects and folders for which you don't have read permission. Select paths in the folder tree to narrow your search if necessary.

**TIP**

Code Search remembers your last settings, such as the project and repository or path that you searched in. Clear the checkboxes to search across all projects easily with the **Clear all** links when you want to search in a different scope. In the results pane, Code Search highlights up to the first 100 hits or matches found in the target files.

## Search code with REST API

You can use APIs to extend or supplement the capabilities listed in this article. For information about Code Search with REST API, see [Fetch Code Search Results](#).

## Next steps

[Search work items](#)

## Related articles

- [Get started with Search](#)
- [Search artifacts and packages](#)
- [Search work items](#)
- [Search FAQs](#)

# Functional work item search

12/13/2022 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Functional work item search command filters extend your ability to refine your search of work items based on assignment, work item type, specific fields, and more. This is in addition to the filter functions documented in [Get started with search](#). Work item search is a built-in feature available to all Azure DevOps users.

You can use Work Item Search by default without any installation when the Boards service is installed and enabled in Azure DevOps Services.

By using Work Item Search, you can do the following tasks and more.

SEARCH TASK	DESCRIPTION
<a href="#">Search over all your projects</a>	Search in your own and your partner teams' backlog. Use cross-project searches over all the work items to search across your enterprise's entire work items. Narrow your search by using project and area path filters.
<a href="#">Search across all work item fields</a>	Quickly and easily find relevant work items by searching across all work item fields, including custom fields. Use a full text search across all fields to efficiently locate relevant work items. The snippet view indicates where matches were found.
<a href="#">Search in specific fields</a>	Use the quick in-line search filters to narrow down to a list of work items in seconds. Use the filters on any work item field. The list of suggestions helps complete your search faster. For example, a search such as <code>AssignedTo:Chris</code> <code>WorkItemType:Bug State:Active</code> finds all active bugs assigned to a user named Chris.
<a href="#">Search across test</a>	Search across Test Plans, Test Suites, and other test work item types.
<a href="#">Take advantage of integration with work item tracking</a>	The Work Item Search interface integrates with familiar controls for managing your work items; letting you view, edit, comment, share, and more.

## Prerequisites

- All users can use work item search.

### Search by work item ID

Enter the work item ID in the Azure DevOps title bar to quickly go to it. Searching for a work item ID opens the work item in a modal dialog, providing quick access to read and edit work items.

This project **119**

USER STORY 119

119 Login behaviour for booking

General

State: Closed Area: FabrikamFiber Web  
Reason: Acceptance te ... Iteration: FabrikamFiber Web\Iteration 2

Description

Login behaviour for booking

TEST SUITE 2 **272541**

FabrikamTest

Mateo Escobedo 0 Add tag

In Progress State: In Progress Area: Fabrikam  
New test suite Reason: New test suite Iteration: Fabrikam\Sprint 1

Description

## Full text search across all fields

You can easily search across all work item fields, including custom fields, which enables more natural searches. The snippet view indicates where matches were found.

## Search across all fields

Easily search across all work item fields.  
The snippet indicates where the matches were found.

- Use simple search strings for words or phrases. Work item search matches derived forms of your search terms; for example, a search for "updating" also finds instances of the word "updated" and "update". Searches aren't case-sensitive.
- When you search from inside a project, the default is to search only within that project.
- While searching from inside a team, the default is to search only within the default area path of that team.
  - When you have one project selected, you see a list of area paths in that project for which you have read access - you won't see any projects and area paths for which you don't have read permission
  - Select area paths in the tree to narrow your search if necessary.
- The selected projects are always at the top of the list. Notice that hit counts are also shown for projects that aren't selected.
- Open the search results in a new browser tab from either the main search function or by selecting **Ctrl + Shift + Enter**.

## Work item search best practices

- Use a text search across all fields to efficiently locate relevant work items. Text search is useful when you're trying to, for example, search for all work items that had similar exception trace.
- Use the quick in-line search filters on any work item field to narrow down to a list of work items in seconds. The list of suggestions helps complete your search faster.

## Search vs. managed work item queries

You have two ways to find and list work items: managed queries and the main search function. If you're looking for a single work item, use the main search. If you want to generate a list of work items to triage, update, chart, or share with others, use a managed query.

With the main search function, you can search against a more fully indexed set of fields than that of managed queries.

---

### Use a managed query

#### Search

- 
- List items to perform bulk updates to fields.
  - Review work that's in progress or recently closed.
  - Triage work: set priority, review, update.

- Create a chart and add it to a dashboard.
- Create a chart to get a count of items or sum a field.
- Create a chart that shows a burndown or burnup over time.
- View a tree of parent-child related work items.
- List work items with link relationships.
- List work items for a single project, multiple projects, or across all projects.
  
- Find a specific work item using its ID or a keyword.
- Find one or more work items across all projects in a fast, flexible manner.
- Perform full text search across all work item fields.
- Review work items assigned to a specific team member.
- Search against specific work item fields to quickly narrow down a list of work items.
- Determine what key words will support a managed search.
- List work items for a single project, multiple projects, or across all projects.

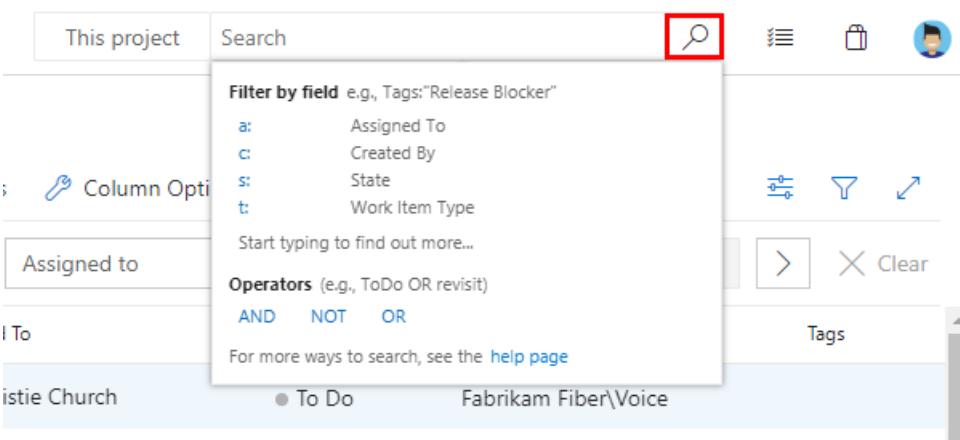
To get started, see the following articles:

- [View and run a query](#)
- [Use search](#)
- [Define a query](#)

For specific managed query examples, see [Query quick reference](#), [Example queries](#).

## Apply supported functions to work item search

1. Fine-tune your search by specifying the fields to search. Enter  a: and a user name to search for all items assigned to that user.



See the following quick filters that you can use:

- a: for **Assigned to**:
- c: for **Created by**:
- s: for **State**
- t: for **Work item type**

2. Start entering the name of a field in your work items; for example, enter  ta .

The screenshot shows a search interface with a search bar containing 'ta'. Below the search bar is a dropdown menu titled 'Filter by field' with suggestions: 'Tags', 'Target', 'Target Date', 'Target Resolve Date', 'Task Due Date', and 'Task Type'. Below this is a section for 'Operators' with examples: 'tags:Critical' and 'AND NOT OR'. At the bottom, it says 'For more ways to search, see the [help page](#)'.

The dropdown list shows work item field name suggestions that match user input. These suggestions help you complete the search faster. For example, a search such as `tags:Critical` finds all work items tagged 'Critical'.

3. Add more filters to further narrow your search, and use Boolean operators to combine terms if necessary. For example, `a: Chris t: Bug s: Active` finds all active bugs assigned to a user named Chris.
4. Narrow your search to specific types and states, by using the selector lists at the top of the results page.
5. Widen your search across all projects, or narrow it to specific types and states. Use the filter to show the selector lists.

The screenshot shows the search interface with the 'Search this organization' button highlighted with a red box. Other visible elements include the search bar ('login'), filter panel button ('Show filter panel'), and view options ('Relevance', 'View').

6. Select the criteria you want in the drop-down selector lists, or search across the entire organization.

The screenshot shows the search interface with the 'Search this organization' button highlighted with a red box. Below it, two dropdown menus are open: 'Types: All' (with 'All Types' checked) and 'States: All' (with 'All States' checked). The 'Work item' tab is selected in the navigation bar.

7. Sort the results as you need using the drop-down list of field names, work item types, or by relevance.

The screenshot shows the Azure DevOps search interface. At the top, there are navigation links for 'This project', 'login', and a search bar. Below the search bar, there are filters for 'Area: FabrikamFiber Web' and 'Types: All'. The main area displays three work items related to 'Login':

- 62 Login page**: Resolved, Description: Login page
- 119 Login behaviour for booking**: Closed, Description: Login behaviour for booking
- 97 Login and logout behaviours**: Closed, Description: Login and logout behaviours

To the right of the results, a 'Sort by' dropdown menu is open, with 'Relevance' selected. Other options include 'Assigned To', 'Changed Date', 'Created Date', 'ID', 'State', 'Tags', 'Title', and 'Work Item Type'.

1. Fine-tune your search by specifying the fields to search. Enter `a:` and a user name to search for all items assigned to that user.

This screenshot shows the search filter interface. It includes a search bar at the top and a sidebar with the following sections:

- Filter by field**: e.g., Tags:"Release Blocker"
  - `a:` Assigned To
  - `c:` Created By
  - `s:` State
  - `t:` Work Item Type
- Start typing to find out more...**
- Operators**: (e.g., ToDo OR revisit)
  - AND
  - NOT
  - OR
- For more ways to search, see the [help page](#)**

See the following quick filters that you can use:

- `a:` for **Assigned to**:
- `c:` for **Created by**:
- `s:` for **State**
- `t:` for **Work item type**

2. Start entering the name of a field in your work items; for example, enter `ta`.

The screenshot shows a search interface with a search bar containing 'ta'. Below the search bar is a dropdown menu titled 'Filter by field' with examples like 'e.g., Tags:"Release Blocker"'. The menu lists several work item fields: Tags, Target, Target Date, Target Resolve Date, Task Due Date, and Task Type. Below this, there's a section for 'Operators' with examples like 'a:@me (t:Task OR t:Bug)' and Boolean operators AND, NOT, OR. At the bottom, it says 'For more ways to search, see the [help page](#)'.

The dropdown list shows work item field name suggestions that match user input. These suggestions help you complete the search faster. For example, a search such as **tags:Critical** finds all work items tagged 'Critical'.

3. Add more filters to further narrow your search, and use Boolean operators to combine terms if necessary. For example, **a: Chris t: Bug s: Active** finds all active bugs assigned to a user named Chris.
4. Narrow your search to specific types and states, by using the drop-down selector lists at the top of the results page.
5. Widen your search across all projects, or narrow it to specific types and states. Use the filter to show the selector lists.

This screenshot shows the search results page. The search bar contains 'login'. Below the search bar are links for 'item 3', 'Wiki 0', and a 'Search this organization' button. To the right are dropdowns for 'Relevance' and 'View', and a 'Show filter panel' button. A red box highlights the 'Search this organization' button.

6. Select the criteria you want in the drop-down selector lists, or search across the entire organization.

This screenshot shows the search results page with the 'Work item' dropdown selected. The dropdown shows 'FabrikamFiber Web' with a red box around the 'x' icon. Below it is a list of area paths: 'Team1' and 'Team2'. To the right of the dropdown are two filter panels: 'Types: All' and 'States: All', both with red boxes around them. The 'Types' panel shows 'All Types' (checked) and 'User Story'. The 'States' panel shows 'All States' (checked) and 'Closed', 'Resolved'.

7. Sort the results as you need using the drop-down list of field names, work item types, or by relevance.

This screenshot shows the Microsoft Azure DevOps interface for searching work items. At the top, there are navigation links for 'This project', 'login', and a search bar. Below that, a header bar includes 'Code 0', 'Work item 3', 'Wiki 0', and a link to 'Search this organization'. On the right side of the header, there is a 'Sort by' dropdown menu with 'Relevance' selected, indicated by a red box.

The main area displays search results for 'Login page' under the 'FabrikamFiber Web' area. The results show three work items:

- ID 62 Login page**: Resolved, Description: Login page
- ID 119 Login behaviour for booking**: Closed, Description: Login behaviour for booking
- ID 97 Login and logout behaviours**: Closed, Description: Login and logout behaviours

The 'Sort by' dropdown menu lists various fields for sorting work items, including 'Assigned To', 'Changed Date', 'Created Date', 'ID', 'Relevance' (which is checked), 'State', 'Reason', 'Tags', 'Title', and 'Work Item Type'.

## Quick filters for matching in specific fields

Quick inline search filters let you refine work items in seconds. The dropdown list of suggestions helps complete your search faster. Mix and match the functions to create quick powerful searches.

USAGE	EXAMPLE
Scope your search terms to match in any work item field including custom fields. Enter the field name followed by the search terms.	<code>tags:Critical</code> finds work items having a field 'tags' containing the term 'Critical'.
Use multiple inline search filters to scope your search by any work item field, including custom fields.	<code>t: Bug path:"project\search"</code> finds all bugs in the area path "project\search".
Use the operators <code>&gt;</code> , <code>&gt;=</code> , <code>&lt;</code> , <code>&lt;=</code> , <code>=</code> , and <code>!=</code> for date, integer, and float fields.	<code>t: Bug CreatedDate &gt; @Today -7</code> finds all bugs created in the last week.
For the search query that contains multiple terms and users looking for exact match, embed the search term inside <code>" "</code>	<code>BuildPath: "tools.demoproject.com"</code> finds all work items that necessarily contain the path "tools.demoproject.com".

## Quick Filters

Quick in-line search filters lets you refine work items by specific criteria on any work item field, in seconds!

🔍

## Scope projects and area and iteration paths using filters

Filters make it easy to narrow the search to specified projects and area paths.

Narrow the search to a specific location using the `proj`, `area`, `iteration`, `path`, and `comment` filters:

USAGE	EXAMPLE
Finds all occurrences of the word <b>Wiki</b> in the <b>Fabrikam</b> project.	<code>Wiki proj:Fabrikam</code>
Finds all occurrences of the word <b>Wiki</b> in the area path <b>Contoso/Mobile</b> and its subpaths.	<code>Wiki area:Contoso/Mobile</code>
Finds all occurrences of the word <b>Wiki</b> in the iteration path <b>Contoso/Sprint101</b> and its subpaths.	<code>Wiki iteration:Contoso/Sprint101</code>
Enclose the argument to the filter in double-quotes if it contains a space.	<code>Wiki path:"Contoso/Windows Phones and Devices/Services"</code>
Finds backlog comments	<code>comment:todo</code>

## See more of the work item

You can quickly get a full screen view of the selected work item using  **expand** and  **shrink** in the toolbar. However, another way to see more of the work item, while you can still select work items from the list of matching results, is to hide the left column filter pane by choosing < at the top left of the column. Use > to restore the filter pane.

If you're using a portrait orientation screen, use the **Preview pane: Right** link at the top right of the window to display the code below the search results list.

### TIP

Search remembers the state of the filter pane, configuration of the work item view pane, and its position between sessions as part of your user preferences.

## Search Work Items with REST API

You can use APIs to extend or supplement the capabilities listed in this article. For information about Work Item Search with REST API, see [Fetch Work Item Search Results](#).

## Next steps

[Supported filter functions and more for work items](#)

## Related articles

- [Get started with Search](#)
- [Search code](#)
- [Search artifacts and packages](#)
- [Search FAQs](#)

# Migrate data from Azure DevOps Server to Azure DevOps Services

12/13/2022 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The data migration tool for Azure DevOps provides a high fidelity way to migrate collection databases from Azure DevOps Server to Azure DevOps Services. It's recommended that you download the [migration guide and tool](#) if you're looking to use this service to import your collection(s). The guide serves as a walk through of the different steps involved in an import. Providing best practices, checklists, and helpful tips to make your import as easy as possible. The guide should be used in conjunction with the more technical documentation referenced below to successfully import to Azure DevOps Services.

## Supported Azure DevOps Server versions for import

### IMPORTANT

It can take up to 2-3 weeks after a new RTW version of Azure DevOps Server is released for import support to come online for that version. It's important to take this into consideration when choosing to upgrade shortly after a new RTW Azure DevOps Server release.

The data migration tool for Azure DevOps supports the two latest releases of Azure DevOps Server at a given time. Releases include updates and major releases. Currently the following versions of Azure DevOps Server are supported for import:

- Azure DevOps Server 2020.1.2
- Azure DevOps Server 2020.1.1

### NOTE

The data migration tool doesn't support imports from Azure DevOps Server release candidates (RC). If you're planning on importing your collection database to Azure DevOps Services using this service, it's important that you don't upgrade your production database to an RC release. If you do upgrade, then you will need to wait and upgrade to the release to web (RTW) version when it's available or restore a backup copy of your database from a previous Azure DevOps Server version to import.

Normal release cadence for new Azure DevOps Server versions is once every three-to-four months. Meaning that support for a given version of Azure DevOps Server for migration to Azure DevOps Services should last for anywhere between six-to-eight months. It's important to ensure that your planning accounts for this support window to avoid having to suddenly upgrade to migrate.

## Preview features

#### **NOTE**

If you're not including preview features when running the migration tool, then you will need to re-run the migration tool prepare to generate a new import.json to queue an import. You DO NOT need to include preview features when you re-generate your import.json.

If you had previously been including preview features then you DO NOT need to take any additional actions after Monday, April 23, 2020.

The following features can be included with your import, but are currently in a preview state.

- [Analytics](#) - Note this is only supported for Azure DevOps Server 2019 and later.

When queueing an import you can elect to include preview features with your import. If you do, data related to these features will be copied into your new organization along with all your other data. Should you choose to not include these features then their data will not be copied.

For a list of items not included with an import, see the [migration guide and tool](#).

## Data migration tool for Azure DevOps resources

In general you should use the [Migration guide and tool](#) when going through an import. When it's required, the guide links back to the following articles. These articles offer deeper technical knowledge on various import topics.

### **Import process**

- [Validate a collection for import](#)
- [Prepare a collection for import](#)
- [Prepare for import](#)
  - [Prepare large collections for import](#)
- [Run an import](#)
- [Post import steps](#)

### **Troubleshooting**

- [Troubleshooting validation errors](#)
- [Troubleshooting process errors](#)
- [Troubleshooting import errors](#)

## Q & A

### **Q: Will my Personal Access Tokens also migrate when I migrate from on-premises to Azure DevOps Services?**

A: No. Your tokens will not migrate and you will need to [regenerate your Personal Access Tokens](#) on Azure DevOps Services.

### **Q: If I have feedback or additional questions is there somewhere I can reach out?**

A: You can search the [developer community portal](#) to see if your question is asked and answered and if not, please open up a new issue. If you need assistance with a failed import, please contact Azure DevOps [customer support](#).

## Related articles

- [Migration and process model FAQs](#)

# Migration options

12/13/2022 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

When you decide to make the move from Azure DevOps Server to Azure DevOps Services, you might start fresh with an empty organization. Often, however, you will have existing code, work items, and other assets that you want to move. There are many approaches to doing this which vary in both the fidelity of the data transfer and the complexity of the process.

Prior to migrating data, review the differences that exist between [Azure DevOps Server and Azure DevOps Services](#).

## Option 1: Copy the most important assets manually

By far the easiest option for moving data into Azure DevOps Services is to manually copy your most important assets and start relatively fresh. This can be difficult when you are in the middle of a large project, but you can make it easier if you do some advance planning and schedule your move when it makes sense for your team.

For example, when the Azure DevOps team chose to move from Azure DevOps Server to Azure DevOps Services, we also decided to move from Team Foundation Version Control (TFVC) to Git. This required a fair bit of planning, but when we actually performed our migration, we created a new Git repo using the "tip" version of our TF VC sources, and left our history behind in Azure DevOps Server. We also moved our active work items, and left behind all our old bugs, completed user stories and tasks, and so on.

Here's the general process:

1. Identify the most important assets that you need to migrate - typically source code, work items, or both. Other assets in Azure DevOps Server - build pipelines, test plans, and so forth - are harder to manually migrate.
2. Identify a good time to make the transition.
3. Prepare your target organizations. Create the organizations and team projects that you need, provision users, and so on.
4. Migrate your data.
5. Consider making the source Azure DevOps Server deployments read-only.

## Option 2: High fidelity database migration.

The Azure DevOps Server & Azure DevOps Services product team provides a high fidelity data migration tool. A downloadable Migration Guide is available at <https://aka.ms/AzureDevOpsImport>.



Because the data migration tool operates at a database level, it can provide a very high fidelity migration. If you want to move your existing Azure DevOps Server data into Azure DevOps Services, we strongly recommend using this option.

## Option 3: Using public API-based tools for higher fidelity migration

If for some reason you cannot use the data migration tool but still want a higher fidelity migration than Option 1, you can choose from a variety of tools that use public APIs to move data. Generally these tools can provide a higher fidelity migration than a manual copy of "tip" data, but they are still relatively low fidelity. For example:

- None of them will preserve the dates of TF VC changesets.
- Many of them will not preserve the changed dates of work item revisions.
- None of them will migrate all Azure DevOps Server artifacts.

In general, we only recommend this approach if the extra fidelity beyond a manual copy is critical. If you decide to take this approach, you might consider hiring a consultant who has experience with one or more of the tools. You should definitely consider doing a test migration before doing your final migration.

Many organizations need a very high fidelity migration for only a subset of their work. New work could potentially start directly in Azure DevOps Services. Other work, with less stringent fidelity requirements, could be migrated using one of the other approaches. You will have to weigh the pros and cons of the various approaches against your motivations for moving into Azure DevOps Services and decide for yourself what is the right strategy.

## Related articles

- [About Azure DevOps Services and Azure DevOps Server](#)
- [Pricing, Azure DevOps Services](#)
- [Pricing, Azure DevOps Server](#)

# Validation and import processes

12/13/2022 • 34 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

This article walks you through the preparation that's required to get an import to Azure DevOps Services ready to run. If you encounter errors during the process, see [Troubleshoot import and migration errors](#).

## NOTE

- Visual Studio Team Services (VSTS) is now [Azure DevOps Services](#).
- With the release of Azure DevOps Server 2019, the TFS Database Import Service has been rebranded as the data migration tool for Azure DevOps. This change includes Tfsmigrator (Migrator) becoming the data migration tool. This service works exactly the same as the former import service. If you're running an older version of on-premises Azure DevOps Server with the TFS branding, you can still use this feature to migrate to Azure DevOps as long as you've upgraded to one of the supported server versions.
- Before you begin the import tasks, check to ensure that you're running a [supported version of Azure DevOps Server](#).

We recommend that you use the [Step-by-step migration guide](#) to progress through your import. The guide links to technical documentation, tools, and best practices.

## Prerequisites

- You must set up an Azure Active Directory tenant as described [Azure AD Connect sync: Make a change to the default configuration](#). The data migration tool sets up a link to your Azure Active Directory tenant when your Azure DevOps Services organization is created as part of the beginning of the import process.

By synchronizing your on-premises Active Directory with Azure Active Directory, your team members will be able to use the same credentials to authenticate and your Azure DevOps Services administrators will be able to leverage your Active Directory groups for setting permissions within your Azure DevOps Services organization. To setup the synchronization, you will want to use the Azure AD Connect technology.

## Validate a collection

After you've confirmed that you're running the latest version of Azure DevOps Server, your next step is to validate each collection that you want to migrate to Azure DevOps Services.

The validation step examines various aspects of your collection, including, but not limited to, size, collation, identity, and processes.

You run the validation by using the data migration tool. To start, [download the tool](#), copy the zip file to one of your Azure DevOps Server application tiers, and then unzip it. You can also run the tool from a different machine without Azure DevOps Server installed as long as the machine can connect to the configuration database of the Azure DevOps Server instance. An example is shown here.

1. Open a Command Prompt window on the server, and enter a cd command to change to the directory where the data migration tool is stored. Take a few moments to review the help content that's provided with the tool.
  - a. To view the top-level help and guidance, run the following command:

Migrator /help

b. View the help text for the command:

Migrator validate /help

- Because this is your first time validating a collection, let's keep it simple. Your command should have the following structure:

```
Migrator validate /collection:{collection URL} /tenantDomainName:{name}
```

Where `{name}` provides the name of your Azure Active Directory tenant. For example, to run against the *DefaultCollection* and the *fabrikam* tenant, the command would look like:

```
Migrator validate /collection:http://localhost:8080/DefaultCollection  
/tenantDomainName:fabrikam.OnMicrosoft.com
```

- To run the tool from a machine other than the Azure DevOps Server, you need the `/connectionString` parameter. The connection string parameter points to your Azure DevOps Server configuration database. As an example, if the validate command is being run by the Fabrikam corporation, the command would look like:

```
Migrator validate /collection:http://fabrikam:8080/DefaultCollection  
/tenantDomainName:fabrikam.OnMicrosoft.com /connectionString:"Data Source=fabrikam;Initial  
Catalog=Configuration;Integrated Security=True"
```

IMPORTANT

The data migration tool *does not* edit any data or structures in the collection. It reads the collection only to identify issues.

4. After the validation is complete, you can view the log files and results.

```
C:\TfsMigrator\TfsMigrator>TfsMigrator.exe validate /collection:http://localhost:8080/tfs/Foo
Microsoft Team Foundation Server (R) Tfs Migrator Tool version 14.95.25504.0
Copyright (C) Microsoft Corporation. All rights reserved.

-----
Validating Foo
-----

Validating that the collection is OK to be imported...

Validating Collection Metadata      Passed  <Step 8 of 8>
Validating Project Processes        Passed  <Validating Project 1 of 1>

Collection validation completed.

Results:
+ All collection validations passed

-----
Validating Import Files
-----
```

During validation, you'll receive a warning if some of your pipelines contain per-pipeline retention rules. Azure DevOps Services uses a [project-based retention model](#) and *doesn't* support per-pipeline retention policies. If you proceed with the migration, the policies won't be carried over to the hosted version.

Instead, the default project-level retention policies will apply. Retain builds important to you, to avoid their loss.

After all the validations pass, you can move to the next step of the import process. If the data migration tool flags any errors, you need to correct them before you proceed. For guidance on correcting validation errors, see [Troubleshoot import and migration errors](#).

## Import log files

When you open the log directory, you'll notice several logging files.

The main log file is named *DataMigrationTool.log*. It contains details about everything that was run. To make it easier for you to focus on specific areas, a log is generated for each major validation operation.

For example, if TfsMigrator reports an error in the "Validating Project Processes" step, you can open the *ProjectProcessMap.log* file to view everything that was run for that step instead of having to scroll through the entire log.

You should review the *TryMatchOobProcesses.log* file only if you're trying to import your project processes to use the [inherited model](#). If you don't want to use the inherited model, you can ignore these errors, because they won't prevent you from importing to Azure DevOps Services.

## Generate import files

By now, you've run the data migration tool validation against the collection, and it's returning a result of "All collection validations passed." Before you take a collection offline to migrate it, you need to generate the import files. When you run the `prepare` command, you generate two import files:

- *IdentityMapLog.csv*: Outlines your identity map between Active Directory and Azure Active Directory (Azure AD).
- *import.json*: Requires you to fill out the import specification you want to use to kick off your migration.

### The `prepare` command

The `prepare` command assists with generating the required import files. Essentially, this command scans the collection to find a list of all users to populate the identity map log, *IdentityMapLog.csv*, and then tries to connect to Azure AD to find each identity's match. To do this, your company needs to use the [Azure Active Directory Connect tool](#) (formerly known as the Directory Synchronization tool, Directory Sync tool, or DirSync.exe tool).

If directory synchronization is set up, the data migration tool should be able to find the matching identities and mark them as *Active*. If it doesn't find a match, the identity is marked as *Historical* in the identity map log, and you'll need to investigate why the user isn't included in your directory sync. The import specification file, *import.json*, should be filled out prior to the import.

Unlike the `validate` command, `prepare` *does* require an internet connection, because it needs to connect to Azure AD to populate the identity map log file. If your Azure DevOps Server instance doesn't have internet access, you need to run the tool from a machine that does. As long as you can find a machine with an intranet connection to your Azure DevOps Server instance and an internet connection, you can run this command. For help with the `prepare` command, run the following command:

```
Migrator prepare /help
```

Included in the help documentation are instructions and examples for running the `Migrator` command from the Azure DevOps Server instance itself and a remote machine. If you're running the command from one of the Azure DevOps Server instance's application tiers, your command should have the following structure:

```
Migrator prepare /collection:{collection URL} /tenantDomainName:{name} /region:{region}
```

```
Migrator prepare /collection:{collection URL} /tenantDomainName:{name} /region:{region}  
/connectionString:"Data Source={sqlserver};Initial Catalog=Configuration;Integrated Security=True"
```

The **ConnectionString** parameter is a pointer to the configuration database of your Azure DevOps Server instance. As an example, if the `prepare` command is being run by the Fabrikam corporation, the command would look like:

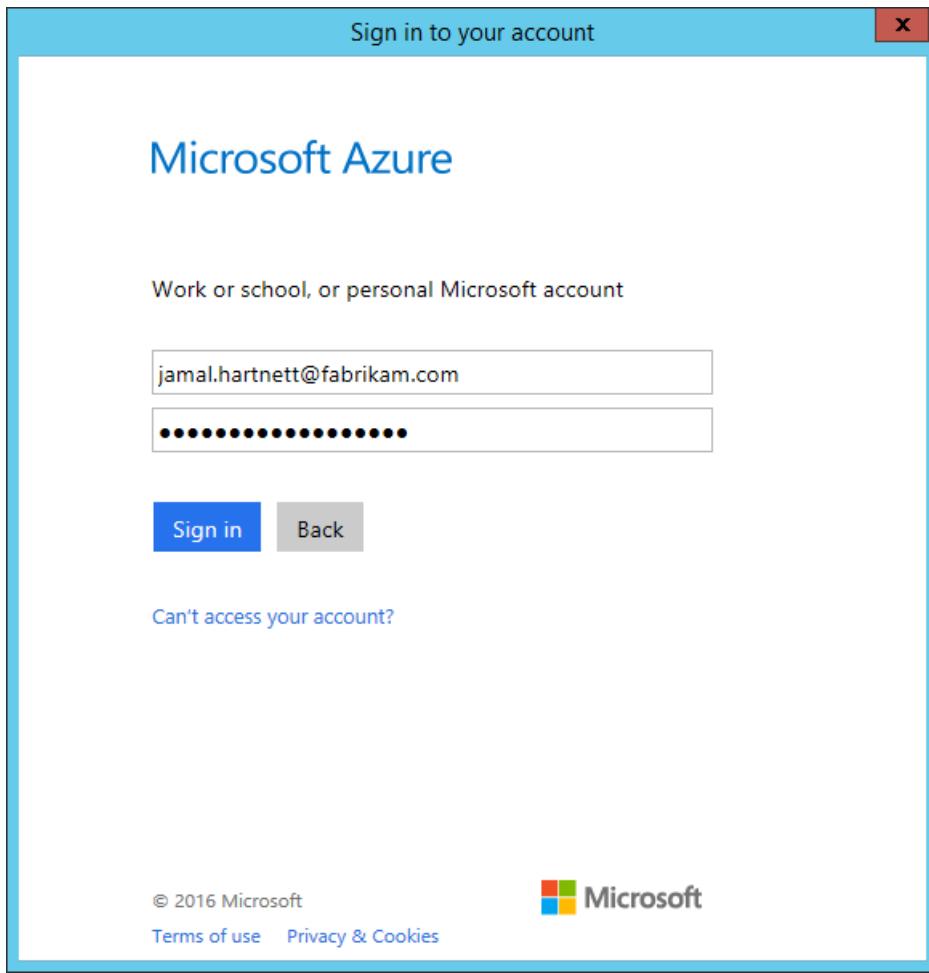
```
Migrator prepare /collection:http://fabrikam:8080/DefaultCollection  
/tenantDomainName:fabrikam.OnMicrosoft.com /region:{region} /connectionString:"Data Source=fabrikam;Initial  
Catalog=Configuration;Integrated Security=True"
```

When the data migration tool runs the `prepare` command, it runs a complete validation to ensure that nothing has changed with your collection since the last full validation. If any new issues are detected, no import files are generated.

Shortly after the command has started running, an Azure AD sign-in window is displayed. You need to sign in with an identity that belongs to the tenant domain that's specified in the command. Make sure that the specified Azure AD tenant is the one you want your future organization to be backed with. In our Fabrikam example, a user would enter credentials that are similar to what's shown in the following screenshot.

**IMPORTANT**

Do *not* use a test Azure AD tenant for a test import and your production Azure AD tenant for the production run. Using a test Azure AD tenant can result in identity import issues when you begin your production run with your organization's production Azure AD tenant.



When you run the `prepare` command successfully in the data migration tool, the results window displays a set of logs and two import files. In the log directory, you'll find a logs folder and two files:

- *import.json* is the import specification file. We recommend that you take time to fill it out.
- *IdentityMapLog.csv* contains the generated mapping of Active Directory to Azure AD identities. Review it for completeness before you kick off an import.

The two files are described in greater detail in the next sections.

### The import specification file

The import specification, *import.json*, is a JSON file that provides import settings. It includes the desired organization name, storage account information, and other information. Most of the fields are autopopulated, and some fields require your input before you attempt an import.

```

1  [
2    "Source": {
3      "Location": "<Provide the SASKey to the Azure storage container with the collection and import files.>",
4      "Files": {
5        "Dacpac": "Tfs_DefaultCollection.dacpac"
6      }
7    },
8    "Target": {
9      "Name": "<Provide a name for the account that will be created during the import.>"
10 },
11   "Properties": {
12     "ImportType": "<Provide the Type of Import: DryRun, ProductionRun>"
13   },
14   "ValidationData": {
15     "TfsMigratorVersion": "16.255.65000.0",
16     "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17     "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18     "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19     "CommandExecutionCount": 0,
20     "CommandExecutionTime": 0.0,
21     "TfsVersion": "Dev15.M117",
22     "DatabaseTotalSize": 181,
23     "DatabaseBlobSize": 0,
24     "DatabaseTableSize": 181,
25     "DatabaseLargestTableSize": 8,
26     "ActiveUserCount": 8,
27     "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28     "Region": "CUS",
29     "ValidationChecksumVersion": 1,
30     "ValidationChecksum": "66516G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
31   },
32   "Identities": [
33     "S-1-5-21-1374400868-3601225936-2087002269-500",
34     "S-1-5-21-2127521184-1604012920-1887927527-11008431",
35     "S-1-5-21-2127521184-1604012920-1887927527-15795496"
36   ]
37 ]

```

0 0 0 json | import.json Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JSON 😊

The *import.json* file's displayed fields and required actions are described in the following table:

FIELD	DESCRIPTION	REQUIRED ACTION
Source	Information about the location and names of the source data files that are used for import.	No action required. Review information for the subfield actions to follow.
Location	The shared access signature key to the Azure storage account that hosts the data-tier application package (DACPAC).	No action required. This field will be covered in a later step.
Files	The names of the files containing import data.	No action required. Review information for the subfield actions to follow.

FIELD	DESCRIPTION	REQUIRED ACTION
DACPAC	A DACPAC file that packages the collection database to be used to bring in the data during the import.	No action required. In a later step, you'll generate this file by using your collection and then upload it to an Azure storage account. You'll need to update the file based on the name you use when you generate it later in this process.
Target	Properties of the new organization to import into.	No action required. Review information for the subfield actions to follow.
Name	The name of the organization to be created during the import.	Provide a name. The name can be quickly changed later after the import has completed. <b>Note:</b> Do <i>not</i> create an organization with this name before you run the import. The organization will be created as part of the import process.
ImportType	The type of import that you want to run.	No action required. In a later step, you'll select the type of import to run.
Validation Data	Information that's needed to help drive your import experience.	The "ValidationData" section is generated by the data migration tool. It contains information that's needed to help drive your import experience. Do <i>not</i> edit the values in this section, or your import could fail to start.

After you complete the preceding process, you should have a file that looks like the following:

```
import.json - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
{} importjson x
1 [
2   "Source": {
3     "Location": "<Provide the SASKey to the Azure storage container with the collection and import files.>",
4     "Files": {
5       "Dacpac": "Tfs_DefaultCollection.dacpac"
6     }
7   },
8   "Target": {
9     "Name": "fabrikam-import"
10 },
11   "Properties": {
12     "ImportType": "<Provide the Type of Import: DryRun, ProductionRun>"
13   },
14   "ValidationData": {
15     "TfsMigratorVersion": "16.255.65000.0",
16     "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17     "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18     "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19     "CommandExecutionCount": 0,
20     "CommandExecutionTime": 0.0,
21     "TfsVersion": "Dev15.M117",
22     "DatabaseTotalSize": 181,
23     "DatabaseBlobSize": 0,
24     "DatabaseTableSize": 181,
25     "DatabaseLargestTableSize": 8,
26     "ActiveUserCount": 8,
27     "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28     "Region": "CUS",
29     "ValidationChecksumVersion": 1,
30     "ValidationChecksum": "66516G8u850KY6XKJmMM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
31   },
32   "Identities": [
33     "S-1-5-21-1374400868-3601225936-2087002269-500",
34     "S-1-5-21-2127521184-1604012920-1887927527-11008431",
35     "S-1-5-21-2127521184-1604012920-1887927527-15795496"
36   ]
37 ]
```

0 1 0 1 json | import.json Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JSON 😊

In the preceding image, note that the planner of the Fabrikam import added the organization name *fabrikam-import* and selected CUS (Central United States) as the region for import. Other values were left as is to be modified just before the planner took the collection offline for the migration.

#### NOTE

Dry-run imports have a '-dryrun' automatically appended to the end of the organization name. This can be changed after the import.

## Supported Azure regions for import

Azure DevOps Services is available in several [Azure regions](#). However, not all regions where Azure DevOps Services is available are supported for import. The following table lists the Azure regions that you can select for import. Also included is the value that you need to place in the import specification file to target that region for import.

GEOGRAPHIC REGION	AZURE REGION	IMPORT SPECIFICATION VALUE
United States	Central United States	CUS

GEOGRAPHIC REGION	AZURE REGION	IMPORT SPECIFICATION VALUE
Europe	Western Europe	WEU
United Kingdom	United Kingdom South	UKS
Australia	Australia East	EAU
South America	Brazil South	SBR
Asia Pacific	South India	MA
Asia Pacific	Southeast Asia (Singapore)	SEA
Canada	Central Canada	CC

## The identity map log

The identity map log is of equal importance to the actual data that you'll be migrating to Azure DevOps Services. As you're reviewing the file, it's important to understand how identity import operates and what the potential results could entail. When you import an identity, it can become either *active* or *historical*. Active identities can sign in to Azure DevOps Services, but historical identities cannot.

### Active identities

Active identities refer to identities that will be users in Azure DevOps Services post-import. In Azure DevOps Services, these identities are licensed and are displayed as users in the organization. The identities are marked as *active* in the **Expected Import Status** column in the identity map log file.

### Historical identities

Historical identities are mapped as such in the **Expected Import Status** column in the identity map log file. Identities without a line entry in the file also become historical. An example of an identity without a line entry might be an employee who no longer works at a company.

Unlike active identities, historical identities:

- *Don't* have access to an organization after migration.
- *Don't* have licenses.
- *Don't* show up as users in the organization. All that persists is the notion of that identity's name in the organization, so that its history can be searched later. We recommend that you use historical identities for users who no longer work at the company or who won't need further access to the organization.

#### NOTE

After an identity is imported as historical, it *can't* become active.

## Understand the identity map log file

The identity map log file is similar to the example shown here:

AD: User(TFS)	AD: Security Identifier	AAD: Expected Import User(VSTS)	Expected Import Status	Validation Date
FABRIKAM\Jamal Hartnett	S-1-5-21-983578539-230207283-3682864982-500	No Match Found (Check AAD Sync)	Historical	2017-10-31T21:15:44Z
FABRIKAM\Mateo Escobedo	S-1-5-21-4100298327-4227319834-4140607669-500	No Match Found (Check AAD Sync)	Historical	2017-10-31T21:15:44Z
FABRIKAM\Helena Petersen	S-1-5-21-124525095-708259637-1543119021-1419599	helena.petersen@fabrikam.com	Active	2017-10-31T21:15:44Z
FABRIKAM\Raisa Pokrovskaya	S-1-5-21-2127521184-1604012920-1887927527-406986	raisa.pokrovskaya@fabrikam.com	Active	2017-10-31T21:15:44Z

The columns in the identity map log file are described in the following table:

**NOTE**

You and your Azure AD admin will need to investigate users that are marked as *No Match Found (Check Azure AD Sync)* to understand why they aren't part of your Azure AD Connect sync.

COLUMN	DESCRIPTION
Active Directory: User (Azure DevOps Server)	The friendly display name used by the identity in Azure DevOps Server. This name makes it easier to identify which user the line in the map is referencing.
Active Directory: Security Identifier	The unique identifier for the on-premises Active Directory identity in Azure DevOps Server. This column is used to identify users in the collection.
Azure Active Directory: Expected Import User (Azure DevOps Services)	Either the expected sign-in address of the matched soon-to-be-active user or <i>No Match Found (Check Azure AD Sync)</i> , indicating that the identity wasn't found during the Azure Active Directory sync and it will be imported as historical.
Expected Import Status	The expected user import status: either <i>Active</i> if there's a match between your Active Directory and Azure Active Directory, or <i>Historical</i> if there isn't a match.
Validation Date	The last time the identity map log was validated.

As you read through the file, notice whether the value in the **Expected Import Status** column is *Active* or *Historical*. *Active* indicates that it's expected that the identity on this row will map correctly on import and will become active. *Historical* means that the identities will become historical on import. It's important to review the generated mapping file for completeness and correctness.

**IMPORTANT**

Your import will fail if major changes occur to your Azure AD Connect security ID sync between import attempts. You can add new users between dry runs, and you can make corrections to ensure that previously imported historical identities become active. However, changing an existing user that was previously imported as active isn't supported at this time. Doing so will cause your import to fail. An example of a change might be completing a dry-run import, deleting an identity from your Azure AD that was imported actively, re-creating a new user in Azure AD for that same identity, and then attempting another import. In this case, an active identity import will be attempted between the Active Directory and newly created Azure AD identity, but it will cause an import failure.

1. Start by reviewing the correctly matched identities. Are all the expected identities present? Are the users mapped to the correct Azure AD identity?

If any values are incorrectly mapped or need to be changed, contact your Azure AD administrator to verify that the on-premises Active Directory identity is part of the sync to Azure AD and has been set up correctly. For more information, see [Integrate your on-premises identities with Azure Active Directory](#).

2. Next, review the identities that are labeled as *historical*. This labeling implies that a matching Azure AD identity couldn't be found, for any of the following reasons:
  - The identity hasn't been set up for sync between on-premises Active Directory and Azure AD.
  - The identity hasn't been populated in your Azure AD yet (for example, there's a new employee).

- The identity doesn't exist in your Azure AD instance.
- The user who owns that identity no longer works at the company.

To address the first three reasons, you need to set up the intended on-premises Active Directory identity to sync with Azure AD. For more information, see [Integrate your on-premises identities with Azure Active Directory](#). You must set up and run Azure AD Connect for identities to be imported as *active* in Azure DevOps Services.

You can ignore the fourth reason, because employees who are no longer at the company should be imported as *historical*.

#### **Historical identities (small teams)**

##### **NOTE**

The identity import strategy proposed in this section should be considered by small teams only.

If Azure AD Connect hasn't been configured, you'll notice that all users in the identity map log file are marked as *historical*. Running an import this way results in all users being imported as *historical*. We strongly recommended that you configure [Azure AD Connect](#) to ensure that your users are imported as *active*.

Running an import with all historical identities has consequences that need to be considered carefully. It should be considered only by teams with a small number of users and for which the cost of setting up Azure AD Connect is deemed too high.

To import all identities as historical, follow the steps outlined in later sections. When you queue an import, the identity that's used to queue the import is bootstrapped into the organization as the organization owner. All other users are imported as historical. Organization owners can then [add the users back in](#) by using their Azure AD identity. The added users are treated as new users. They do *not* own any of their history, and there's no way to re-parent this history to the Azure AD identity. However, users can still look up their pre-import history by searching for their <domain><Active Directory username>.

The data migration tool displays a warning if it detects the complete historical identities scenario. If you decide to go down this migration path, you'll need to consent in the tool to the limitations.

#### **Visual Studio subscriptions**

The data migration tool can't detect Visual Studio subscriptions (formerly known as MSDN benefits) when it generates the identity map log file. Instead, we recommend that you apply the auto license upgrade feature after the import. As long as users' work accounts are [linked](#) correctly, Azure DevOps Services automatically applies their Visual Studio subscription benefits at their first sign-in after the import. You're never charged for licenses that are assigned during the import, so this can be safely handled afterward.

You don't need to repeat a dry-run import if users' Visual Studio subscriptions aren't automatically upgraded in Azure DevOps Services. Visual Studio subscription linking happens outside the scope of an import. As long as their work account is linked correctly before or after the import, users' licenses are automatically upgraded on their next sign-in. After their licenses have been upgraded successfully, the next time you run an import, the users are upgraded automatically on their first sign-in to the organization.

## **Prepare for import**

By now, you have everything ready to execute on your import. You need to schedule downtime with your team to take the collection offline for the migration. When you've agreed upon a time to run the import, you need to upload to Azure both the required assets you've generated and a copy of the database. This process has five steps:

Step 1: [Take the collection offline and detach it](#).

#### **NOTE**

If the data migration tool displays a warning that you can't use the DACPAC method, you have to perform the import by using the SQL Azure virtual machine (VM) method. Skip steps 2 to 5 in that case and follow instructions provided in [Import large collections](#) and then continue to section [determine the import type](#).

Step 2: [Generate a DACPAC file from the collection you're going to import](#).

Step 3: [Upload the DACPAC file and import files to an Azure storage account](#).

Step 4: [Generate an SAS key to the storage account](#).

Step 5: [Complete the import specification](#).

#### **NOTE**

Before you perform a production import, we *strongly* recommend that you complete a dry-run import. With a dry run, you can validate that the import process works for your collection and that there are no unique data shapes present that might cause a production import failure.

### **Step 1: Detach your collection**

[Detaching the collection](#) is a crucial step in the import process. Identity data for the collection resides in the Azure DevOps Server instance's configuration database while the collection is attached and online. When a collection is detached from the Azure DevOps Server instance, it takes a copy of that identity data and packages it with the collection for transport. Without this data, the identity portion of the import *can't* be executed. We recommend that you keep the collection detached until the import has been completed, because there isn't a way to import the changes that occurred during the import.

If you're doing a dry run (test) import, we recommend that you reattach your collection after you back it up for import, because you won't be concerned about having the latest data for this type of import. To avoid offline time altogether, you can also choose to employ an [offline detach](#) for dry runs.

It's important to weigh the cost of choosing to incur zero downtime for a dry run. It requires taking backups of the collection and configuration database, restoring them on a SQL instance, and then creating a detached backup. A cost analysis could prove that taking just a few hours of downtime to directly take the detached backup is better in the long run.

### **Step 2: Generate a DACPAC file**

DACPACs offer a fast and relatively easy method for moving collections into Azure DevOps Services. However, after a collection database size exceeds a certain threshold, the benefits of using a DACPAC start to diminish.

#### **NOTE**

If the data migration tool displays a warning that you can't use the DACPAC method, you have to perform the import by using the SQL Azure virtual machine (VM) method provided in [Import large collections](#).

If the data migration tool doesn't display a warning, use the DACPAC method described in this step.

[DACPAC](#) is a feature of SQL server that allows database changes to be packaged into a single file and deployed to other instances of SQL. A DACPAC file can also be restored directly to Azure DevOps Services, so you can use it as the packaging method for getting your collection's data in the cloud. You use the SqlPackage.exe tool to generate the DACPAC file. The tool is included as part of [SQL Server Data Tools \(SSDT\)](#).

Multiple versions of the SqlPackage.exe tool are installed with SSDT. The versions are stored in folders with names such as 120, 130, and 140. When you use SqlPackage.exe, it's important to use the right version to prepare the DACPAC.

- TFS 2018 imports need to use the SqlPackage.exe version from the 140 folder or higher.

If you installed SSDT for Visual Studio, you'll find your SqlPackage.exe version in one of the following folder paths:

- If you installed SSDT and integrated it with an existing installation of Visual Studio, your SqlPackage.exe folder path is similar to  
C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\Extensions\Microsoft\SQLDB\DAC\130\ .
- If you installed SSDT as a standalone installation, your SqlPackage.exe folder path is similar to  
C:\Program Files (x86)\Microsoft Visual Studio\2017\SQL\Common7\IDE\Extensions\Microsoft\SQLDB\DAC\130\ .
- If you already have an installation of SQL Server, SqlPackage.exe might already be present, and your folder path is similar to %PROGRAMFILES%\Microsoft SQL Server\130\DAC\bin\ .

Both versions of SSDT that you can download from [SQL Server Data Tools](#) include both the 130 and 140 folders and their SqlPackage.exe versions.

When you generate a DACPAC, keep two considerations in mind: the disk that the DACPAC will be saved on and the disk space on the machine that's generating the DACPAC. You want to ensure that you have enough disk space to complete the operation.

As it creates the package, SqlPackage.exe temporarily stores data from your collection in the temp directory on drive C of the machine you're initiating the packaging request from.

You might find that your drive C is too small to support creating a DACPAC. You can estimate the amount of space you'll need by looking for the largest table in your collection database. DACPACs are created one table at a time. The maximum space requirement to run the generation is roughly equivalent to the size of the largest table in the collection's database. If you're saving the generated DACPAC to drive C, you also need to take into account the size of the collection database as reported in the *DataMigrationTool.log* file from a validation run.

The *DataMigrationTool.log* file provides a list of the largest tables in the collection each time the validate command is run. For an example of table sizes for a collection, see the following output. Compare the size of the largest table with the free space on the drive that hosts your temporary directory.

#### **IMPORTANT**

Before you proceed with generating a DACPAC file, ensure that your collection is [detached](#).

[Info @08:23:59.539] Table name	Size in MB
[Info @08:23:59.539] dbo.tbl_Content	38984
[Info @08:23:59.539] dbo.tbl_LocalVersion	1935
[Info @08:23:59.539] dbo.tbl_Version	238
[Info @08:23:59.539] dbo.tbl_FileReference	85
[Info @08:23:59.539] dbo.Rules	68
[Info @08:23:59.539] dbo.tbl_FileMetadata	61

Ensure that the drive that hosts your temporary directory has at least as much free space. If it doesn't, you need to redirect the temp directory by setting an environment variable.

```
SET TEMP={location on disk}
```

Another consideration is where the DACPAC data is saved. Pointing the save location to a far-off remote drive could result in much longer generation times. If a fast drive such as a solid-state drive (SSD) is available locally, we recommend that you target the drive as the DACPAC save location. Otherwise, it's always faster to use a disk that's on the machine where the collection database resides rather than a remote drive.

Now that you've identified the target location for the DACPAC and ensured that you have enough space, it's time to generate the DACPAC file.

Open a Command Prompt window and go to the SqlPackage.exe location. To generate the DACPAC, replace the placeholder values with the required values, and then run the following command:

```
SqlPackage.exe /sourceconnectionstring:"Data Source={database server name};Initial Catalog={Database Name};Integrated Security=True" /targetFile:{Location & File name} /action:extract /p:ExtractAllTableData=true /p:IgnoreUserLoginMappings=true /p:IgnorePermissions=true /p:Storage=Memory
```

- **Data Source:** The SQL Server instance that hosts your Azure DevOps Server collection database.
- **Initial Catalog:** The name of the collection database.
- **targetFile:** The location on the disk and the DACPAC file name.

A DACPAC generation command that's running on the Azure DevOps Server data tier itself is shown in the following example:

```
SqlPackage.exe /sourceconnectionstring:"Data Source=localhost;Initial Catalog=Foo;Integrated Security=True" /targetFile:C:\DACPAC\Foo.dacpac /action:extract /p:ExtractAllTableData=true /p:IgnoreUserLoginMappings=true /p:IgnorePermissions=true /p:Storage=Memory
```

The output of the command is a DACPAC file that's generated from the collection database *Foo* called *Foo.dacpac*.

#### Configure your collection for import

After your collection database has been restored on your Azure VM, configure a SQL login to allow Azure DevOps Services to connect to the database to import the data. This login allows only *read* access to a single database.

To start, open SQL Server Management Studio on the VM, and then open a new query window against the database to be imported.

Set the database's recovery to simple:

```
ALTER DATABASE [<Database name>] SET RECOVERY SIMPLE;
```

Create a SQL login for the database, and assign that login the 'TFSEXECROLE':

```
USE [<database name>]
CREATE LOGIN <pick a username> WITH PASSWORD = '<pick a password>'
CREATE USER <username> FOR LOGIN <username> WITH DEFAULT_SCHEMA=[dbo]
EXEC sp_addrolemember @rolename='TFSEXECROLE', @membername='<username>'
```

Following our Fabrikam example, the two SQL commands would look like the following:

```
ALTER DATABASE [Foo] SET RECOVERY SIMPLE;

USE [Foo]
CREATE LOGIN fabrikam WITH PASSWORD = 'fabrikamimport1!'
CREATE USER fabrikam FOR LOGIN fabrikam WITH DEFAULT_SCHEMA=[dbo]
EXEC sp_addrolemember @rolename='TFSEXECROLE', @membername='fabrikam'
```

**NOTE**

Be sure to enable [SQL Server and Windows authentication mode](#) in SQL Server Management Studio on the VM. If you don't enable authentication mode, the import will fail.

**Configure the import specification file to target the VM**

Update the import specification file to include information about how to connect to the SQL Server instance.

Open your import specification file and make the following updates:

1. Remove the DACPAC parameter from the source files object.

The import specification before the change is shown in the following code:

```
"Source": {  
    "Location": "<Provide the SASKey to the Azure storage container with the collection and  
import files.>",  
    "Files": {  
        "Dacpac": "Tfs_DefaultCollection.dacpac"  
    }  
},
```

The import specification after the change is shown in the following code:

```
"Source": {  
    "Properties": {  
        "ConnectionString": "Data Source=8.8.8.8;Initial Catalog=Tfs_Foo;Integrated Security=False;  
User ID=fabrikam;Password=fabrikam1!;Encrypt=True;TrustServerCertificate=True"  
    }  
},
```

2. Fill out the required parameters and add the following properties object within your source object in the specification file.

```
"Properties":  
{  
    "ConnectionString": "Data Source={SQL Azure VM Public IP};Initial Catalog={Database  
Name};Integrated Security=False;User ID={SQL Login Username};Password={SQL Login  
Password};Encrypt=True;TrustServerCertificate=True"  
}
```

Following the Fabrikam example, after you apply the changes, the import specification would look like the following:

import.json - Untitled (Workspace) - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

{} import.json x

```
1  [
2      "Source": {
3          "Properties": {
4              "ConnectionString": "Data Source=8.8.8.8;Initial Catalog=Tfs_Foo;Integrated Security=False;
5                  User ID=fabrikam;Password=fabrikam1!;Encrypt=True;TrustServerCertificate=True"
6          }
7      },
8      "Target": {
9          "Name": "fabrikam-import"
10     },
11     "Properties": {
12         "ImportType": "DryRun"
13     },
14     "ValidationData": {
15         "TfsMigratorVersion": "16.255.65000.0",
16         "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17         "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18         "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19         "CommandExecutionCount": 0,
20         "CommandExecutionTime": 0.0,
21         "TfsVersion": "Dev15.M117",
22         "DatabaseTotalSize": 181,
23         "DatabaseBlobSize": 0,
24         "DatabaseTableSize": 181,
25         "DatabaseLargestTableSize": 8,
26         "ActiveUserCount": 8,
27         "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28         "Region": "CUS",
29         "ValidationChecksumVersion": 1,
30         "ValidationChecksum": "66516G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
31     },
32     "Identities": [
33         "S-1-5-21-1374400868-3601225936-2087002269-500",
34         "S-1-5-21-2127521184-1604012920-1887927527-11008431",
35         "S-1-5-21-2127521184-1604012920-1887927527-15795496"
36     ]
37 ]
```

Your import specification is now configured to use a SQL Azure VM for import. Proceed with the rest of preparation steps to import to Azure DevOps Services. After the import has finished, be sure to delete the SQL login or rotate the password. Microsoft does not retain the login information after the import has finished.

### **Step 3: Upload the DACPAC file**

## NOTE

If you're using the SQL Azure VM method, you need to provide only the connection string. You don't have to upload any files, and you can skip this step.

Your DACPAC must be placed in an Azure storage container. This can be an existing container or one created specifically for your migration effort. It's important to ensure that your container is created in the right region.

Azure DevOps Services is available in multiple [regions](#). When you're importing to these regions, it's critical to place your data in the correct region to ensure that the import can start successfully. Your data must be placed in the same region that you'll be importing to. Placing the data anywhere else will result in the import being unable to start. The following table lists the acceptable regions for creating your storage account and uploading your data.

DESIRED IMPORT REGION	STORAGE ACCOUNT REGION
Central United States	Central United States
Western Europe	Western Europe
United Kingdom	United Kingdom South
Australia East	Australia East
Brazil South	Brazil South
India South	India South
Canada Central	Canada Central
Asia Pacific (Singapore)	Asia Pacific (Singapore)

Although Azure DevOps Services is available in multiple regions in the US, only the Central United States region accepts new Azure DevOps Services. You can't import your data into other US Azure regions at this time.

You can [create a blob container](#) from the Azure portal. After you've created the container, you need to upload the Collection DACPAC file.

After the import has finished, you can delete the blob container and accompanying storage account. To do so, you can use tools such as [AzCopy](#) or any other Azure storage explorer tool, such as [Azure Storage Explorer](#).

#### NOTE

If your DACPAC file is larger than 10 GB, we recommend that you use AzCopy. AzCopy has multithreaded upload support for faster uploads.

#### Step 4: Generate an SAS key

A [shared access signature \(SAS\) key](#) provides delegated access to resources in a storage account. The key allows you to give Microsoft the lowest level of privilege that's required to access your data for executing the import.

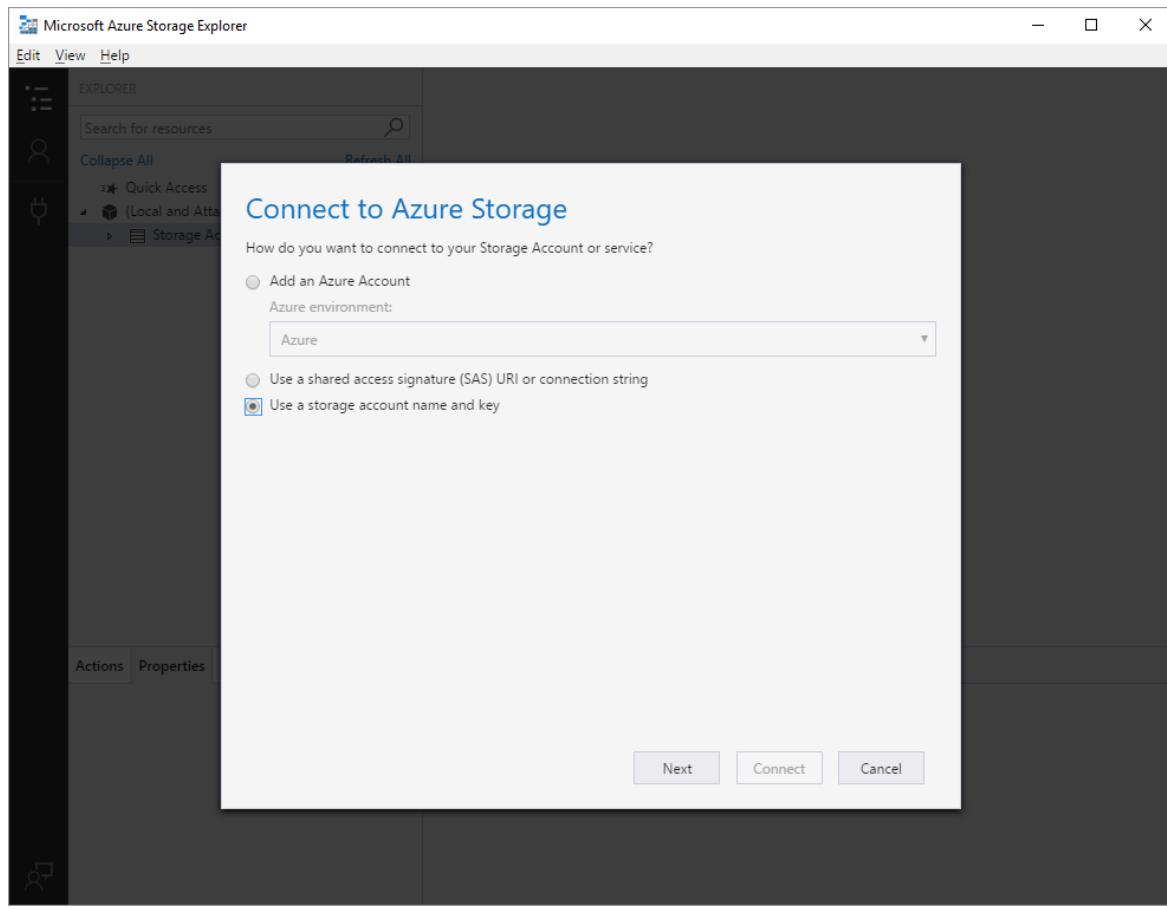
The recommended way to generate an SAS key is to use [Azure Storage Explorer](#). With Storage Explorer, you can easily create container-level SAS keys. This is essential, because the data migration tool does *not* support account-level SAS keys.

#### NOTE

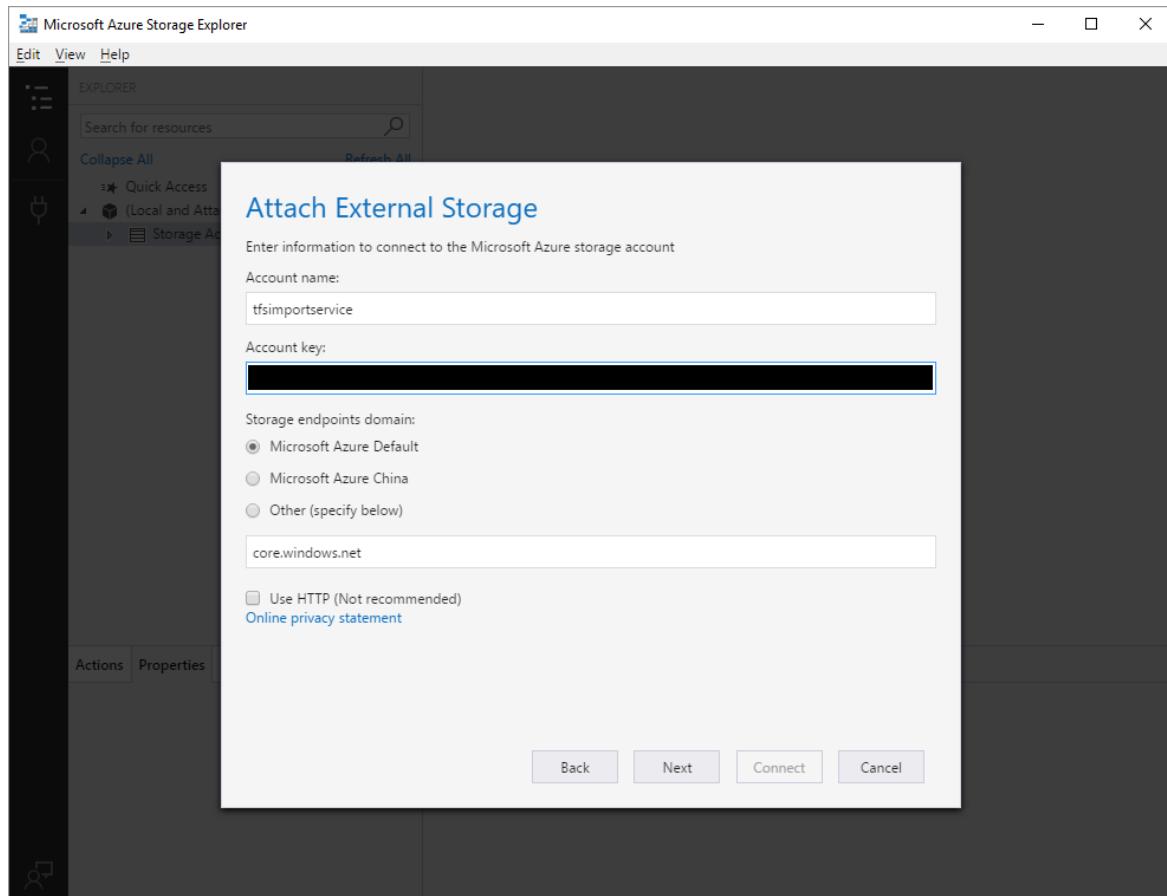
Do *not* generate an SAS key from the Azure portal. Azure portal-generated SAS keys are account scoped and don't work with the data migration tool.

After you install Storage Explorer, you can generate an SAS key by doing the following:

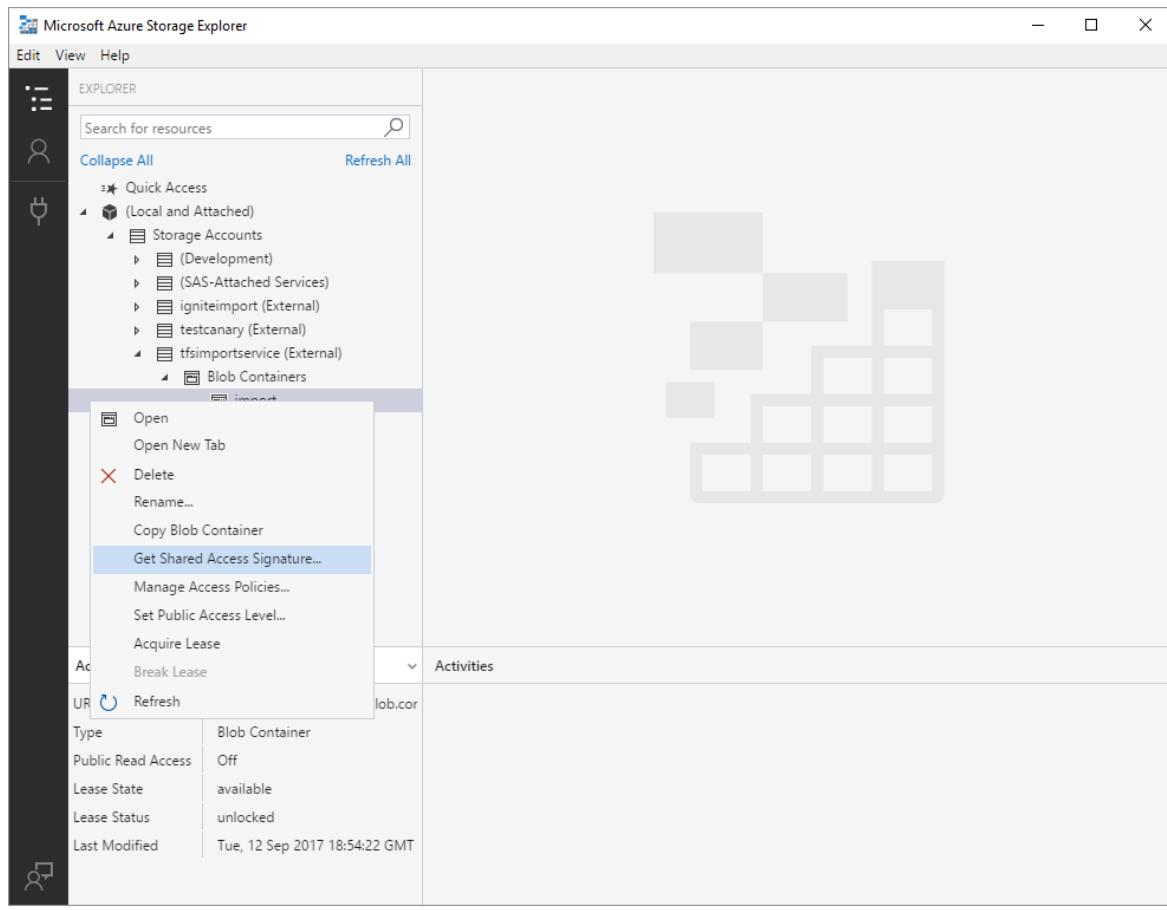
1. Open Storage Explorer.
2. Add an account.
3. Select **Use a storage account name and key**, and then select **Connect**.



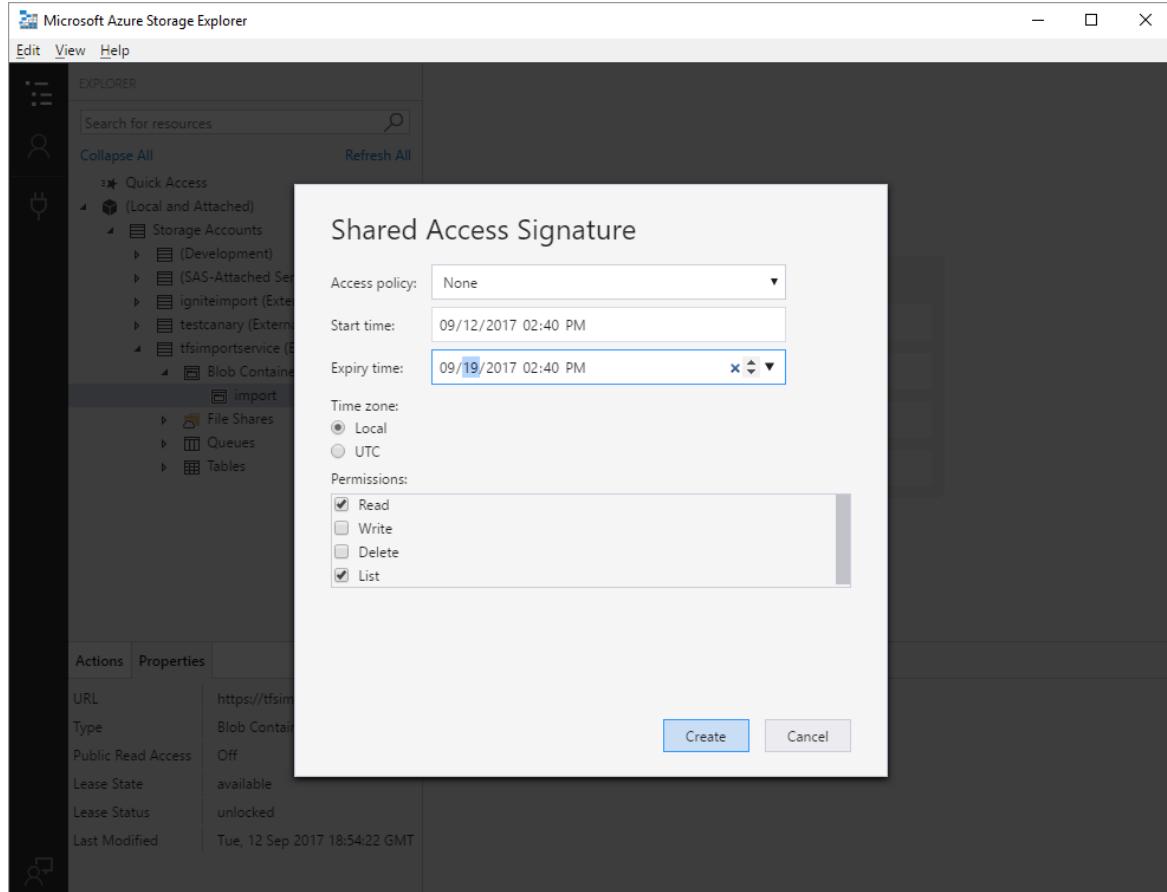
4. On the **Attach External Storage** pane, enter your storage account name, provide one of your two **primary access keys**, and then select **Connect**.



5. On the left pane, expand **Blob Containers**, right-click the container that stores your import files, and then select **Get Shared Access Signature**.



6. For **Expiry time**, set the expiration date for seven days in the future.



7. Under **Permissions** for your SAS key, select the **Read** and **List** check boxes. Write and delete permissions aren't required.

## NOTE

- Copy and store this SAS key to place in your import specification file in the next step.
- Treat this SAS key as a secret. It provides access to your files in the storage container.

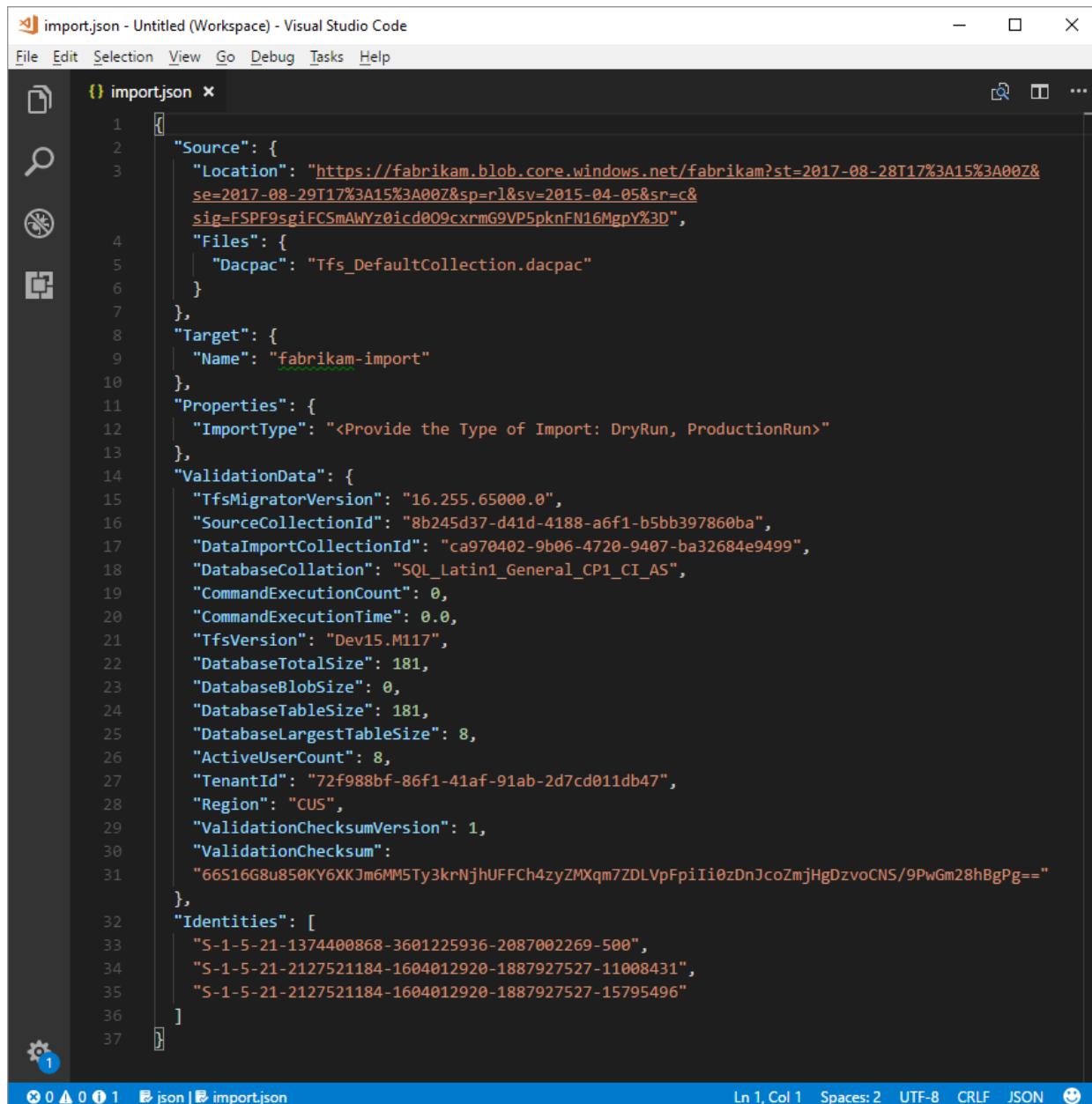
## Step 5: Complete the import specification

Earlier in the process you partially filled out the import specification file generally known as *import.json*. At this point, you have enough information to complete all the remaining fields except for the import type. The import type will be covered later, in the import section.

In the *import.json* specification file, under **Source**, complete the following fields:

- **Location:** Paste the SAS key you generated from the script and then copied in the preceding step.
- **Dacpac:** Ensure that the file, including the *.dacpac* file extension, has the same name as the DACPAC file you uploaded to the storage account.

Using the Fabrikam example, the final import specification file should look like the following:



The screenshot shows a Visual Studio Code window with the title "import.json - Untitled (Workspace) - Visual Studio Code". The code editor displays the following JSON content:

```
1  "Source": {  
2      "Location": "https://fabrikam.blob.core.windows.net/fabrikam?st=2017-08-28T17%3A15%3A00Z&se=2017-08-29T17%3A15%3A00Z&sp=r&sv=2015-04-05&sr=c&sig=FSPF9sg1FC5mAlYz0icd009cxrmG9VP5pknFN16MgpY%3D",  
3      "Files": {  
4          "Dacpac": "Tfs_DefaultCollection.dacpac"  
5      }  
6  },  
7  "Target": {  
8      "Name": "fabrikam-import"  
9  },  
10 "Properties": {  
11     "ImportType": "<Provide the Type of Import: DryRun, ProductionRun>"  
12 },  
13 "ValidationData": {  
14     "TfsMigratorVersion": "16.255.65000.0",  
15     "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",  
16     "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",  
17     "DatabaseCollation": "SQL_Latin1_General_CI_AS",  
18     "CommandExecutionCount": 0,  
19     "CommandExecutionTime": 0.0,  
20     "TfsVersion": "Dev15.M117",  
21     "DatabaseTotalSize": 181,  
22     "DatabaseBlobSize": 0,  
23     "DatabaseTableSize": 181,  
24     "DatabaseLargestTableSize": 8,  
25     "ActiveUserCount": 8,  
26     "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",  
27     "Region": "CUS",  
28     "ValidationChecksumVersion": 1,  
29     "ValidationChecksum":  
30         "66516G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="  
31     },  
32     "Identities": [  
33         "S-1-5-21-1374400868-3601225936-2087002269-500",  
34         "S-1-5-21-2127521184-1604012920-1887927527-11008431",  
35         "S-1-5-21-2127521184-1604012920-1887927527-15795496"  
36     ]  
37 }
```

The status bar at the bottom shows: Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JSON 😊

## Restrict access to Azure DevOps Services IPs only

We highly recommend that you restrict access to your Azure Storage account to only IPs from Azure DevOps

Services. You do this by allowing connections only from the set of Azure DevOps Services IPs that are involved in the collection database import process. The IPs that need to be granted access to your storage account depend on the region you're importing into. Use the `IpList` option to get the list of IPs that need to be granted access.

Included in the help documentation are instructions and examples for running Migrator from the Azure DevOps Server instance itself and a remote machine. If you're running the command from one of the Azure DevOps Server instance's application tiers, your command should have the following structure:

```
Migrator IpList /collection:{CollectionURI} /tenantDomainName:{name} /region:{region}
```

#### NOTE

Alternatively, you can also use [Service Tags](#) in place of explicit IP ranges. Azure Service Tags are a convenient way for customers to manage their networking configuration to allow traffic from specific Azure services. Customers can easily allow access by adding the tag name `azuredevops` to their network security groups or firewalls either through the portal or programmatically.

### Determine the import type

Imports can be queued as either a dry run or a production run. The `ImportType` parameter determines the import type:

- **DryRun**: Use a dry run for test purposes. The system deletes dry runs after 21 days.
- **ProductionRun**: Use a production run when you want to keep the resulting import and use the organization full time in Azure DevOps Services after the import finishes.

#### TIP

We always recommend that you complete a dry-run import first.

```
import.json - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
{} importjson x
1 [
2   "Source": {
3     "Location": "https://fabrikam.blob.core.windows.net/fabrikam?st=2017-08-28T17%3A15%3A00Z&se=2017-08-29T17%3A15%3A00Z&sp=r&sv=2015-04-05&sr=c&sig=FSPF9sg1C5mAwYz0icd009cxrmG9VP5pknFN16MgpY%3D",
4     "Files": {
5       "Dacpac": "Tfs_DefaultCollection.dacpac"
6     }
7   },
8   "Target": {
9     "Name": "fabrikam-import"
10  },
11  "Properties": {
12    "ImportType": "DryRun"
13  },
14  "ValidationData": {
15    "TfsMigratorVersion": "16.255.65000.0",
16    "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17    "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18    "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19    "CommandExecutionCount": 0,
20    "CommandExecutionTime": 0.0,
21    "TfsVersion": "Dev15.M117",
22    "DatabaseTotalSize": 181,
23    "DatabaseBlobSize": 0,
24    "DatabaseTableSize": 181,
25    "DatabaseLargestTableSize": 8,
26    "ActiveUserCount": 8,
27    "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28    "Region": "CUS",
29    "ValidationChecksumVersion": 1,
30    "ValidationChecksum": "66S16G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
31  },
32  "Identities": [
33    "S-1-5-21-1374400868-3601225936-2087002269-500",
34    "S-1-5-21-2127521184-1604012920-1887927527-11008431",
35    "S-1-5-21-2127521184-1604012920-1887927527-15795496"
36  ]
37 ]
```

0 0 1 json | import.json Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JSON

## Dry-run organizations

Dry-run imports help teams test the migration of their collections. Organizations are expected not to remain around forever but to exist for a short time. In fact, before a production migration can be run, any completed dry-run organizations will need to be deleted. All dry-run organizations have a *limited existence and are automatically deleted after a set period of time*. Information about when the organization will be deleted is included in the success email you should receive after the import finishes. Be sure to take note of this date and plan accordingly.

Most dry-run organizations have 15 days before they're deleted. Dry-run organizations can also have a 21-day expiration if more than 100 users have a basic or greater license at *import time*. After the specified time period, the dry-run organization is deleted. You can repeat dry-run imports as many times as you need before you do a production migration. You need to delete any previous dry runs before you attempt a new one. When your team is ready to perform a production migration, you'll need to manually delete the dry-run organization.

For more information about post-import activities, see the [post import](#) article.

If you encounter any import problems, see [Troubleshoot import and migration errors](#).

## Run an import

Your team is now ready to begin the process of running an import. We recommend that you start with a

successful dry-run import before you attempt a production-run import. With dry-run imports, you can see in advance how an import will look, identify potential issues, and gain experience before you head into your production run.

**NOTE**

If you need to repeat a completed production-run import for a collection, as in the event of a rollback, contact Azure DevOps Services [Customer Support](#) before you queue up another import.

**NOTE**

Azure administrators can prevent users from creating new Azure DevOps organizations. If the Azure AD tenant policy is turned on, your import will fail to finish. Before you begin, verify that the policy isn't set or that there is an exception for the user that is performing the migration. For more information, see [Restrict organization creation via Azure AD tenant policy](#).

**NOTE**

Azure DevOps Services does not support per-pipeline retention policies, and they will not be carried over to the hosted version.

## Considerations for rollback plans

A common concern for teams that are doing a final production run is what their rollback plan will be if anything goes wrong with import. This is why we highly recommend doing a dry run to make sure that you're able to test the import settings you provide to the data migration tool for Azure DevOps.

Rollback for the final production run is fairly simple. Before you queue the import, you detach the team project collection from Azure DevOps Server or Team Foundation Server, which will make it unavailable to your team members. If for any reason you need to roll back the production run and bring the on-premises server back online for your team members, you can do so. You simply attach the team project collection on-premises again and inform your team that they'll continue to work normally while your team regroups to understand any potential failures.

## Queue an import

**IMPORTANT**

Before you proceed, ensure that your collection was [detached](#) prior to generating a DACPAC file or uploading the collection database to a SQL Azure VM. If you don't complete this step, the import will fail. In the event that your import fails, see [Troubleshoot import and migration errors](#).

You start an import by using the data migration tool's `import` command. The import command takes an import specification file as input. It parses the file to ensure that the provided values are valid and, if successful, it queues an import to Azure DevOps Services. The import command requires an internet connection, but does *not* require a connection to your Azure DevOps Server instance.

To get started, open a Command Prompt window, and change directories to the path to the data migration tool. We recommended that you take a moment to review the help text provided with the tool. Run the following command to see the guidance and help for the import command:

```
Migrator import /help
```

The command to queue an import will have the following structure:

```
Migrator import /importFile:{location of import specification file}
```

Here is an example of a completed import command:

```
Migrator import /importFile:C:\DataMigrationToolFiles\import.json
```

After the validation passes, you'll be asked to sign in to Azure AD. It's important to sign in with an identity that's a member of the same Azure AD tenant as the identity map log file was built against. The user that signs in becomes the owner of the imported organization.

**NOTE**

Each Azure AD tenant is limited to five imports per 24-hour period. Only imports that are queued count against this cap.

When your team initiates an import, an email notification is sent to the user that queued the import. About 5 to 10 minutes after it queues the import, your team can go to the organization to check on the status. After the import finishes, your team is directed to sign in, and an email notification is sent to the organization owner.

## Related articles

- [Migrate options](#)
- [Post-import](#)

# Import large collections

12/13/2022 • 12 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

For databases that the data migration tool warns are too big, a different data packaging approach is required to migrate to Azure DevOps Services. If you're unsure whether your collection exceeds the size threshold, you should run a data migration tool validation on the collection. The validation lets you know whether you need to use the SQL Azure VM method for import.

## Determine if you can reduce the collection size

Before you proceed, we recommend checking to see whether your [old data can be cleaned up](#). Over time, collections can build up very large volumes of data. This is a natural part of the DevOps process, but you might find that you don't need to retain all of the data. Some common examples of no longer relevant data are older workspaces and build results.

Cleaning older, no-longer-relevant artifacts could remove a lot more space than you might expect, and it could determine whether you use the DACPAC import method or a SQL Azure VM.

### IMPORTANT

After you've deleted older data, it *can't* be recovered unless you restore an older backup of the collection.

If you're under the DACPAC threshold, follow the instructions to [generate a DACPAC](#) for import. If you still can't get the database under the DACPAC threshold, you need to set up a SQL Azure VM to import to Azure DevOps Services.

## Set up a SQL Azure VM to import to Azure DevOps Services

Let's walk through how to accomplish this. At a high level, you'll:

- Set up a SQL Azure VM.
- (Optional) Restrict access to Azure DevOps Services IPs only.
- Configure IP firewall exceptions.
- Restore your database on the VM.
- Configure your collection for import.
- Configure the import specification file to target the VM

## Set up a SQL Azure VM

You can set up a SQL Azure VM from the Azure portal with just a few clicks. To learn how, see [Use the Azure portal to provision a Windows virtual machine with SQL Server](#).

Azure DevOps Services is available in several [Azure regions](#) across the globe.

## IMPORTANT

To ensure that the import starts successfully, it's critical to place your data in the correct region. If you set up your SQL Azure VM in a location other than the regions listed in the following table, the import will fail to start.

If you're using this import method, determine where to create your SQL Azure VM by referring to the following table. Creating your VM in a region other than those in this list is not supported for running an import.

DESIRED IMPORT REGION	SQL AZURE VM REGION
Central United States	Central US
Western Europe	West Europe
Australia East	Australia East
Brazil South	Brazil South
South India	South India
Central Canada	Canada Central
Asia Pacific	Southeast Asia (Singapore)
UK South	UK South

Although Azure DevOps Services is available in multiple regions in the US, only the Central United States region accepts new organizations. Companies can't import their data into other US Azure regions at this time.

## NOTE

DACPAC customers should consult the region table in the ["Step 3: Upload the DACPAC file" section](#). The preceding guidelines are for SQL Azure VMs only.

Here are a few more SQL Azure VM configurations that we recommend:

- Use D Series VMs, because they're optimized for database operations.
- Ensure that the D Series VMs have at least 28 gigabytes (GB) of RAM. For imports, we recommend Azure D12 V2 VM sizes.
- [Configure the SQL temporary database](#) to use a drive other than drive C. Ideally the drive should have ample free space; at least equivalent to your database's [largest table](#).
- If your source database is still over 1 terabyte (TB) after you've [reduced its size](#), you need to [attach additional 1-TB disks](#) and combine them into a single partition to restore your database on the VM.
- If your collection databases are over 1 TB in size, consider using an SSD for both the temporary database and collection database. Also, consider using larger VMs with 16 virtual CPUs (vCPUs) and 128 GB of RAM.
- You need to have a public facing IP address for the service to reach this machine.

## (Optional) Restrict access to Azure DevOps Services IPs only

We highly recommend that you restrict access to your VM to only IPs from Azure DevOps Services. You do this

by allowing connections only from the set of Azure DevOps Services IPs that are involved in the collection database import process. The IPs that need to be granted access to your collection database depend on the region you're importing into. The following tables can help you identify the correct IPs. The only port that's required to be opened to connections is the standard SQL connection port 1433.

First, no matter what Azure DevOps Services region you import into, you must grant the following IP addresses access to your collection database.

**NOTE**

In the following table, the two IP addresses listed with x.x.x.0/23 indicate a range. Allow the entire /23 range. For example, if you're importing into the Central United States region, allow the /23 range for 20.37.158.0. For IP addresses with x.x.x.0/24, allow the /24 range.

SERVICE	IP ADDRESS
Azure DevOps Services Identity Service	168.62.105.45, 40.81.42.115

Next, grant access to the Regional Identity Service. You need to grant an exception for the data migration tool instance only in the region that you're importing into.

SERVICE	IP ADDRESS
Regional Identity Service - Central United States	13.89.236.72, 52.165.41.252, 52.173.25.16, 13.86.38.60, 20.45.1.175, 13.86.36.181, 52.154.53.1, 52.158.209.56, 20.37.138.122, 20.37.158.0/23, 20.37.139.247, 20.37.158.5
Regional Identity Service - West Europe	20.67.123.240, 52.166.54.85, 13.95.233.212, 52.236.145.119, 52.142.235.223, 52.236.147.103, 23.97.221.25, 52.233.181.148, 52.149.110.153, 51.144.61.32, 52.236.147.236, 40.74.28.0/23
Regional Identity Service - Australia East	13.75.145.145, 40.82.217.103, 20.188.213.113, 104.210.88.194, 40.81.62.114, 20.37.194.0/24
Regional Identity Service - Brazil South	20.40.114.3, 191.235.90.183, 191.232.38.181, 191.233.25.175, 191.235.226.0/24
Regional Identity Service - India South	104.211.227.29, 40.81.75.130, 52.172.54.122, 52.172.49.252, 20.41.194.0/24
Regional Identity Service - Canada Central	52.237.19.6, 40.82.190.38, 52.228.82.0/243
Regional Identity Service - Asia Pacific (Singapore)	20.195.68.0/24
Regional Identity Service - UK South	40.81.159.67, 51.104.26.0/24

Next, grant access to the data migration tool for Azure DevOps itself. You need to grant an exception for the data migration tool instance only in the region that you're importing into.

SERVICE	IP ADDRESS
Data migration tool - Central United States	52.173.74.9, 52.165.184.188, 20.45.1.234, 13.86.39.123
Data migration tool - West Europe	40.115.43.138, 13.95.15.128, 52.236.146.105, 40.67.219.89, 40.119.145.63, 52.142.236.228, 52.142.238.75
Data migration tool - Australia East	13.75.134.204, 40.82.219.41, 20.40.124.19
Data migration tool - Brazil South	104.41.24.164, 20.40.115.123
Data migration tool - India South	13.71.120.31, 40.81.76.137
Data migration tool - Canada Central	52.237.18.100, 52.237.24.61, 40.82.191.163
Data migration tool - Asia Pacific (Singapore)	20.195.68.0/24
Data migration tool - UK South	40.81.153.223, 51.105.8.98, 51.104.26.2

Next, grant Azure DevOps Services access. Again, you need to grant an exception for the Azure DevOps Services instance only in the region that you're importing into.

SERVICE	IP ADDRESS
Azure DevOps Services - Central United States	13.89.236.72, 52.165.41.252, 52.173.25.16, 13.86.38.60, 20.45.1.175, 13.86.36.181, 52.158.209.56
Azure DevOps Services - West Europe	52.166.54.85, 13.95.233.212, 52.236.145.119, 52.142.235.223, 52.236.147.103, 23.97.221.25, 52.233.181.148, 52.149.110.153, 51.144.61.32, 52.236.147.236
Azure DevOps Services - Australia East	13.75.145.145, 40.82.217.103, 20.188.213.113, 104.210.88.194, 40.81.62.114
Azure DevOps Services - Brazil South	20.40.114.3, 191.235.90.183, 191.232.38.181, 191.233.25.175
Azure DevOps Services - India South	104.211.227.29, 40.81.75.130, 52.172.54.122, 52.172.49.252
Azure DevOps Services - Canada Central	52.237.19.6, 40.82.190.38
Azure DevOps Services - Asia Pacific (Singapore)	20.195.68.0/24
Azure DevOps Services - UK South	40.81.159.67, 51.105.8.98, 51.104.26.2, 51.104.26.5

Next, grant Azure Pipelines Releases service access. You need to grant an exception for the Azure DevOps Services instance only in the region that you're importing into.

## Release Management IPs

SERVICE	IP ADDRESS
Releases service - United States	23.102.153.83, 23.101.127.247, 23.100.85.250, 13.86.39.233, 40.80.217.53, 52.232.229.122
Releases service - West Europe	13.95.223.69, 104.45.64.13
Releases service - Australia East	13.73.204.151, 20.40.176.135
Releases service - Brazil South	191.235.94.154, 20.40.116.69
Releases service - India South	52.172.15.233, 40.81.79.60
Releases service - Canada Central	52.237.28.171, 40.82.189.127
Releases service - Asia Pacific (Singapore)	20.195.68.0/24
Releases service - UK South	40.81.156.207

Next, grant Azure Artifacts access. Again, you need to grant an exception for the Azure DevOps Services instance only in the region that you're importing into.

### Azure Artifacts IPs

Add exceptions for all three services that make up Azure Artifacts.

SERVICE	IP ADDRESS
Azure Artifacts - United States	52.173.148.93, 104.43.253.181, 23.99.179.148, 40.80.222.154, 40.119.0.130, 40.119.0.139, 13.86.125.169, 20.41.44.47, 40.90.219.165
Azure Artifacts - West Europe	104.46.45.12, 52.236.148.212
Azure Artifacts - Australia East	13.73.100.166, 20.40.176.15, 40.81.59.69
Azure Artifacts - Brazil South	191.234.179.224, 20.40.115.214
Azure Artifacts - India South	52.172.11.191, 40.81.74.79
Azure Artifacts - Canada Central	52.237.24.224, 40.85.224.121, 13.71.189.199, 40.82.188.122
Azure Artifacts - Asia Pacific (Singapore)	20.195.68.0/24
Azure Artifacts - UK South	51.145.120.132

SERVICE	IP ADDRESS
Azure Artifacts Feed - United States	52.173.251.89, 20.45.1.3, 40.67.190.224, 20.41.58.125, 40.119.1.14, 20.45.1.249

SERVICE	IP ADDRESS
Azure Artifacts Feed - West Europe	40.118.19.43, 52.236.146.118
Azure Artifacts Feed - Australia East	13.70.143.138, 20.40.176.80
Azure Artifacts Feed - Brazil South	191.235.93.87, 20.40.116.17
Azure Artifacts Feed - India South	52.172.8.41, 40.81.79.49
Azure Artifacts Feed - Canada Central	52.237.19.70, 40.82.188.254
Azure Artifacts Feed - Asia Pacific (Singapore)	20.195.68.0/24
Azure Artifacts Feed - UK South	51.145.120.49

SERVICE	IP ADDRESS
Azure Artifacts Blob - United States	70.37.94.103, 40.78.129.25, 40.67.155.236, 52.230.216.163, 20.45.3.51
Azure Artifacts Blob - West Europe	23.97.221.25
Azure Artifacts Blob - Australia East	40.127.86.30, 20.188.213.113, 40.82.221.14
Azure Artifacts Blob - Brazil South	191.235.90.183
Azure Artifacts Blob - India South	52.172.54.122
Azure Artifacts Blob - Canada Central	52.237.16.145, 52.237.16.145, 52.233.38.115, 40.82.187.186
Azure Artifacts Blob - Asia Pacific (Singapore)	20.195.68.0/24
Azure Artifacts Blob - UK South	51.143.174.59, 40.81.152.41

## Test Plans IPs

Add exceptions for Test Plans IP addresses only in the region you're migrating into.

SERVICE	IP ADDRESS
Test Plans - United States	52.253.227.131, 40.91.89.233, 20.41.47.199, 40.91.117.40, 40.91.126.113, 20.37.141.154
Test Plans - West Europe	40.119.145.57
Test Plans - Australia East	20.40.177.101
Test Plans - Brazil South	20.40.118.62

SERVICE	IP ADDRESS
Test Plans - India South	40.81.72.10
Test Plans - Canada Central	40.82.184.28
Test Plans - Asia Pacific (Singapore)	20.195.68.0/24
Test Plans - UK South	40.81.159.9

### Analytics IPs (Azure DevOps Server 2019 or later only)

If you included preview features with your import, add an exception for the analytics IPs only in your target import region.

SERVICE	IP ADDRESS
Analytics service - United States	20.41.43.22, 20.36.236.83, 20.41.40.50, 52.143.251.221, 52.242.212.199, 13.86.33.148, 13.86.39.80
Analytics service - West Europe	52.236.146.143, 52.236.146.9, 52.149.108.23
Analytics service - Australia East	20.40.179.159
Analytics service - Brazil South	20.40.113.248
Analytics service - India South	40.81.73.58
Analytics service - Canada Central	40.82.185.214
Analytics service - Asia Pacific (Singapore)	20.195.68.0/24
Analytics service - UK South	40.81.159.247

#### NOTE

Alternatively, you can also use [Service Tags](#) in place of explicit IP ranges. Azure Service Tags are a convenient way for customers to manage their networking configuration to allow traffic from specific Azure services. Customers can easily allow access by adding the tag name `azuredevops` to their network security groups or firewalls either through the portal or programmatically.

## Configure IP firewall exceptions

Granting exceptions for the necessary IPs is handled at the Azure networking layer for your SQL Azure VM. To get started, go to your SQL Azure VM in the [Azure portal](#). In **Settings**, select **Networking**. This will take you to the network interface page for your SQL Azure VM. The data migration tool requires the Azure DevOps Services IPs to be configured for inbound connections only on port 1433. You can grant exceptions for the IPs by selecting **Add inbound port rule** in the networking settings.

PROTOCOL	SOURCE	DESTINATION	ACTION	
TCP	Any	Any	Allow	...
TCP	Any	Any	Allow	...
Any	VirtualNetwork	VirtualNetwork	Allow	...
Any	AzureLoadBalancer	Any	Allow	...
Any	Any	Any	Deny	...

On the Add inbound security rule pane, select **Advanced** to configure an inbound port rule for a specific IP.

Add inbound security rule X

**Advanced**  

**Service** i

Custom ▼

**\* Port range** i

8080 ✓

**\* Priority** i

1510

**\* Name**

Port\_8080 ✓

**Description**

In the **Source** drop-down list, select **IP Addresses**, enter an IP address that needs to be granted an exception, set the **Destination port range** to **1433** and, in the **Name** box, enter a name that best describes the exception you're configuring.

Depending on other inbound port rules that have been configured, you might need to change the default priority for the Azure DevOps Services exceptions so they don't get ignored. For example, if you have a "deny on all inbound connections to 1433" rule with a higher priority than your Azure DevOps Services exceptions, the data migration tool might be unable to make a successful connection to your database.

\* Source ⓘ

IP Addresses ▾

\* Source IP address range ⓘ

168.62.105.45 ✓

\* Source port range ⓘ

\*

\* Destination ⓘ

Any ▾

\* Destination port range ⓘ

1433 ✓

\* Protocol

Any TCP UDP

\* Action

Allow Deny

\* Priority ⓘ

1010 ✓

\* Name

VSTS\_Identity\_Service ✓

Description

[Empty text area]

Repeat adding inbound port rules until all necessary Azure DevOps Services IPs have been granted an exception. Missing one IP could result in your import failing to start.

## Restore your database on the VM

After you set up and configure an Azure VM, you need to take your detached backup from your Azure DevOps Server instance to your Azure VM. Azure has [several documented methods](#) for how to accomplish this task. The collection database needs to be restored on your SQL instance and doesn't require Azure DevOps Server to be installed on the VM.

## Configure your collection for import

After your collection database has been restored on your Azure VM, configure a SQL login to allow Azure DevOps Services to connect to the database to import the data. This login allows only *read* access to a single

database.

To start, open SQL Server Management Studio on the VM, and then open a new query window against the database to be imported.

Set the database's recovery to simple:

```
ALTER DATABASE [<Database name>] SET RECOVERY SIMPLE;
```

Create a SQL login for the database, and assign that login the 'TFSEXECROLE':

```
USE [<database name>]
CREATE LOGIN <pick a username> WITH PASSWORD = '<pick a password>'
CREATE USER <username> FOR LOGIN <username> WITH DEFAULT_SCHEMA=[dbo]
EXEC sp_addrolemember @rolename='TFSEXECROLE', @membername='<username>'
```

Following our Fabrikam example, the two SQL commands would look like the following:

```
ALTER DATABASE [Foo] SET RECOVERY SIMPLE;

USE [Foo]
CREATE LOGIN fabrikam WITH PASSWORD = 'fabrikamimport1!'
CREATE USER fabrikam FOR LOGIN fabrikam WITH DEFAULT_SCHEMA=[dbo]
EXEC sp_addrolemember @rolename='TFSEXECROLE', @membername='fabrikam'
```

#### NOTE

Be sure to enable [SQL Server and Windows authentication mode](#) in SQL Server Management Studio on the VM. If you don't enable authentication mode, the import will fail.

## Configure the import specification file to target the VM

Update the import specification file to include information about how to connect to the SQL Server instance.

Open your import specification file and make the following updates:

1. Remove the DACPAC parameter from the source files object.

The import specification before the change is shown in the following code:

```
"Source": {
  "Location": "<Provide the SASKey to the Azure storage container with the collection and import files.>",
  "Files": {
    "Dacpac": "Tfs_DefaultCollection.dacpac"
  }
},
```

The import specification after the change is shown in the following code:

```
"Source": {
  "Properties": {
    "ConnectionString": "Data Source=8.8.8.8;Initial Catalog=Tfs_Foo;Integrated Security=False;
User ID=fabrikam;Password=fabrikam1!;Encrypt=True;TrustServerCertificate=True"
  }
},
```

2. Fill out the required parameters and add the following properties object within your source object in the specification file.

```

"Properties":
{
    "ConnectionString": "Data Source={SQL Azure VM Public IP};Initial Catalog={Database
Name};Integrated Security=False;User ID={SQL Login Username};Password={SQL Login
Password};Encrypt=True;TrustServerCertificate=True"
}

```

Following the Fabrikam example, after you apply the changes, the import specification would look like the following:

```

import.json - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
import.json x
1 [
2     "Source": {
3         "Properties": {
4             "ConnectionString": "Data Source=8.8.8.8;Initial Catalog=Tfs_Foo;Integrated Security=False;
5             User ID=fabrikam;Password=fabrikam1!;Encrypt=True;TrustServerCertificate=True"
6         }
7     },
8     "Target": {
9         "Name": "fabrikam-import"
10    },
11    "Properties": {
12        "ImportType": "DryRun"
13    },
14    "ValidationData": {
15        "TfsMigratorVersion": "16.255.65000.0",
16        "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17        "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18        "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19        "CommandExecutionCount": 0,
20        "CommandExecutionTime": 0.0,
21        "TfsVersion": "Dev15.M117",
22        "DatabaseTotalSize": 181,
23        "DatabaseBlobSize": 0,
24        "DatabaseTableSize": 181,
25        "DatabaseLargestTableSize": 8,
26        "ActiveUserCount": 8,
27        "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28        "Region": "CUS",
29        "ValidationChecksumVersion": 1,
30        "ValidationChecksum":
31            "66S16G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
32    },
33    "Identities": [
34        "S-1-5-21-1374400868-3601225936-2087002269-500",
35        "S-1-5-21-2127521184-1604012920-1887927527-11008431",
36        "S-1-5-21-2127521184-1604012920-1887927527-15795496"
37    ]
38]

```

0 1 json import.json Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JSON 😊

Your import specification is now configured to use a SQL Azure VM for import. Proceed with the [rest of preparation steps](#) to import to Azure DevOps Services. After the import has finished, be sure to delete the SQL login or rotate the password. Microsoft does not retain the login information after the import has finished.

## Related articles

- [Validation and import processes](#)

# Validate and resolve errors related to process templates

12/13/2022 • 9 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

As part of the migration import process, the data migration tool checks the process used by the projects in the collection. Fix any errors that get flagged.

After resolving the errors, rerun the data migration tool's `validate` command to verify that all errors have been fixed.

## NOTE

It's recommended that you use the [Migration Guide](#) to progress through your import. The guide links to the technical documentation as needed.

With the release of Azure DevOps Server 2019 the TFS Database Import Service was rebranded to Migrate to Azure DevOps. This includes Tfsmigrator becoming the data migration tool or migrator for short. This service still works exactly the same as the old Import Service. If you're on an older version of on-premises with TFS as the branding you can still use this feature to migrate to Azure DevOps as long as you upgrade to one of the supported versions.

## Process validation types

During validation, the data migration tool determines the target process model for each project. It automatically assigns one of the following two process models to each project in the collection:

- **Inherited process model:** If the project was created with the Agile, Scrum, or CMMI process template, and was never customized.
- **Hosted XML process model:** If the project process appears to have been customized. A customized process contains custom fields, work item types, or other types of customizations.

When the Hosted XML process is the targeted process model, the data migration tool validates if the customizations can be migrated. The data migration tool generates two files during the validation:

- **DataMigrationTool.log:** Contains the set of process validation errors found in the collection. Fix all process errors found to proceed with your migration.
- **TryMatchOobProcesses.log:** Lists for each project the target process model - Inheritance or Hosted XML. For projects that are set to target the Hosted XML process model, it explains why they are considered to be customized. You don't have to fix these errors, but they give you guidance what to do in case you want to migrate to the Inheritance process model. Note that once a collection is imported, you can migrate a project to an Inheritance process model.

Most customers have a mix of projects within a collection. Some projects use a default process template and others use custom process templates. The data migration tool checks and validates each project accordingly. It is very possible that you'll have a mix of projects, some mapped to an Inherited process and others to a Hosted XML process.

We recommend that for any project that has not been customized, that you review the `TryMatchOobProcesses.log` to determine if there are any errors. If so, make the adjustments accordingly so

that the project can be mapped to an Inherited process upon data import.

## Update to a system process

If you started with an older version of Azure DevOps Server, odds are your projects are still using an older process template. If those projects have not been updated using the [Configure Features Wizard](#) then the data migration tool will find process errors. In some rare cases, if your process is very old, even the Configure Features Wizard won't be able to resolve the errors.

Here are some examples of error messages you may receive:

```
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element PortfolioBacklog is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element BugWorkItems is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element FeedbackRequestWorkItems is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element FeedbackResponseWorkItems is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField Team.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField RemainingWork.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField Order.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField Effort.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField Activity.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField ApplicationStartInformation.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField ApplicationLaunchInstructions.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574:  
ProcessConfiguration doesn't specify required TypeField ApplicationType.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF400572: The Project Process  
Settings must be configured for this feature to be used.
```

If you have never customized your project (added fields, work item types, etc.), then fixing these errors is actually pretty simple. If you have customized your process, then this approach won't work. You'll need to manually change the process templates so that your customizations don't get overwritten.

First, make sure you know what process your project started as. Is it Scrum, Agile or CMMI? In this example, let us assume Agile. Next, go to the [Process Customization Scripts](#) provided on GitHub and download the repo. In this instance, we are going to focus on contents in the `Import` folder.

Use the `ConformProject.ps1` script to conform a project of your choosing to the Agile system process. This will update the entire project to be Agile.

```
.\ConformProject.ps1 "<collection url>" "<project name>" "c:\process-customization-scripts\import\agile"
```

Make sure you do this for each and every project.

## Resolve process errors

Are your process templates customized? Are you using an older outdated process template? If so, you'll most likely have process validation errors. The data migration tool does an exhaustive check against your process templates. It checks to make sure that it is valid for Azure DevOps Services. Odds are that you'll need to make

some adjustments and apply them to your collection.

#### NOTE

If you are using an OOB Agile, Scrum, or CMMI process, you probably won't see any errors in the **DataMigrationTool.log**. Instead, check the **TryMatchOobProcesses.log** for errors. If you are error free, then your project will map to an OOB process.

There are several customizations that won't work in Azure DevOps Services. Make sure you review the [list of customizations](#) that are supported.

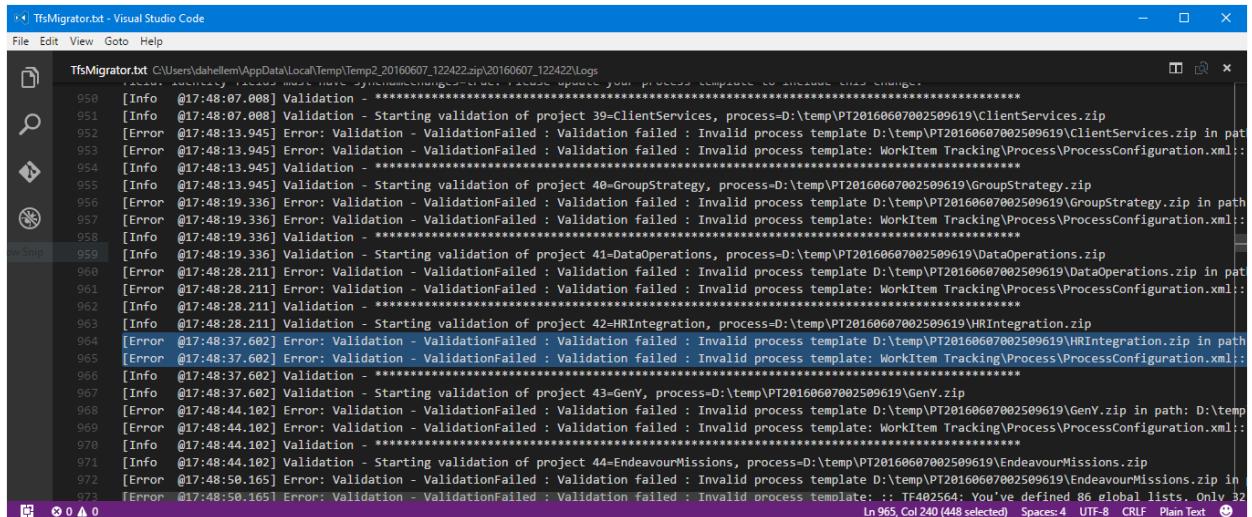
If you have projects that are using an older process template, the data migration tool will find several errors. This is because your process templates hasn't been updated to match the most recent process templates. To start, try running the [Configure Features Wizard](#) for each project. This will attempt to update your process templates with the most recent features. Doing so should drastically reduce the error count.

Finally, make sure you have `witadmin` on the machine that you intend to use to fix the process errors. This can be your local desktop. The `witadmin` command line tool is used in the automated scripts and is required whenever making changes to the process templates.

## Step 1 - Review errors

**DataMigrationTool.log** file will be generated and contains the list of errors that the validation process found. To view the logs, open **DataMigrationTool.log** file. Search for the string "Validation - Starting validation of project 1". Each project is validated. Scan through all the projects and search for any lines that contain a prefix of [Error

....



```
TfsMigrator.txt C:\Users\dahelleml\AppData\Local\Temp\Temp2_20160607_122422.zip\20160607_122422\Logs
File Edit View Goto Help

950 [Info @17:48:07.008] Validation - *****
951 [Info @17:48:07.008] Validation - Starting validation of project 39=ClientServices, process=D:\temp\PT20160607002509619\ClientServices.zip
952 [Error @17:48:13.945] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\ClientServices.zip in path
953 [Error @17:48:13.945] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::
954 [Info @17:48:13.945] Validation - *****
955 [Info @17:48:13.945] Validation - Starting validation of project 40=GroupStrategy, process=D:\temp\PT20160607002509619\GroupStrategy.zip
956 [Error @17:48:19.336] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GroupStrategy.zip in path
957 [Error @17:48:19.336] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::
958 [Info @17:48:19.336] Validation - *****
959 [Info @17:48:19.336] Validation - Starting validation of project 41=DataOperations, process=D:\temp\PT20160607002509619\DataOperations.zip
960 [Error @17:48:28.211] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\DataOperations.zip in path
961 [Error @17:48:28.211] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::
962 [Info @17:48:28.211] Validation - *****
963 [Info @17:48:28.211] Validation - Starting validation of project 42=HRIntegration, process=D:\temp\PT20160607002509619\HRIntegration.zip
964 [Error @17:48:37.602] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\HRIntegration.zip in path
965 [Error @17:48:37.602] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::
966 [Info @17:48:37.602] Validation - *****
967 [Info @17:48:37.602] Validation - Starting validation of project 43=GenY, process=D:\temp\PT20160607002509619\GenY.zip
968 [Error @17:48:44.102] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GenY.zip in path: D:\temp
969 [Error @17:48:44.102] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::
970 [Info @17:48:44.102] Validation - *****
971 [Info @17:48:44.102] Validation - Starting validation of project 44=EndeavourMissions, process=D:\temp\PT20160607002509619\EndeavourMissions.zip
972 [Error @17:48:58.165] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\EndeavourMissions.zip in path: :: TF402564: You've defined 86 global lists. Only 32
973 [Error @17:48:58.165] Error: Validation - ValidationFailed : Validation failed : Invalid process template :: TF402564: You've defined 86 global lists. Only 32
Ln 965, Col 240 (448 selected) Spaces:4 UTF-8 CRLF Plain Text
```

For a list of validation errors, see [Resolve validation errors for process import](#). For each validation error, we have provided the error number, description, and the method to resolve.

## Step 2 - Fix errors

Once you've determined which projects have errors and the error details, fix the errors. Fixing the errors requires that you change the XML syntax and apply the changes back to the project.

#### NOTE

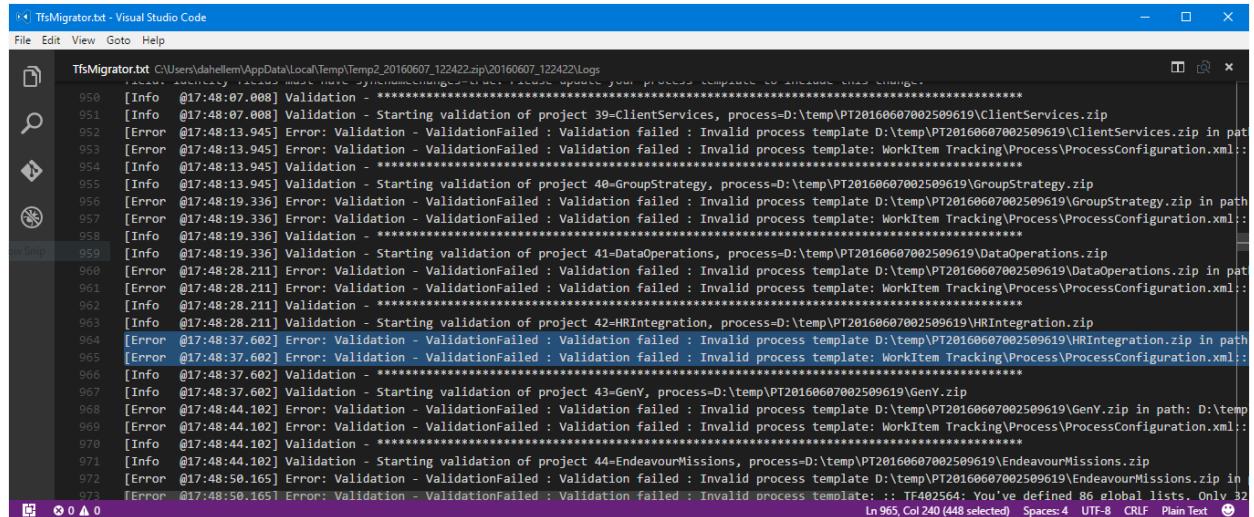
We recommend you don't use TFS Power Tools to do this work. Instead, we highly recommended that you modify the XML manually.

To get the process template from the project add the `/SaveProcesses` parameter when running the data migration tool command.

```
Migrator validate /collection:{collection URL} /SaveProcesses
```

This command will extract the XML from the project and place it into the same folder as the logs. Extract the zip files to your local machine so that you can edit the files.

Now, fix the XML. Use the logs from the `DataMigrationTool.log` file to determine the errors for each project.



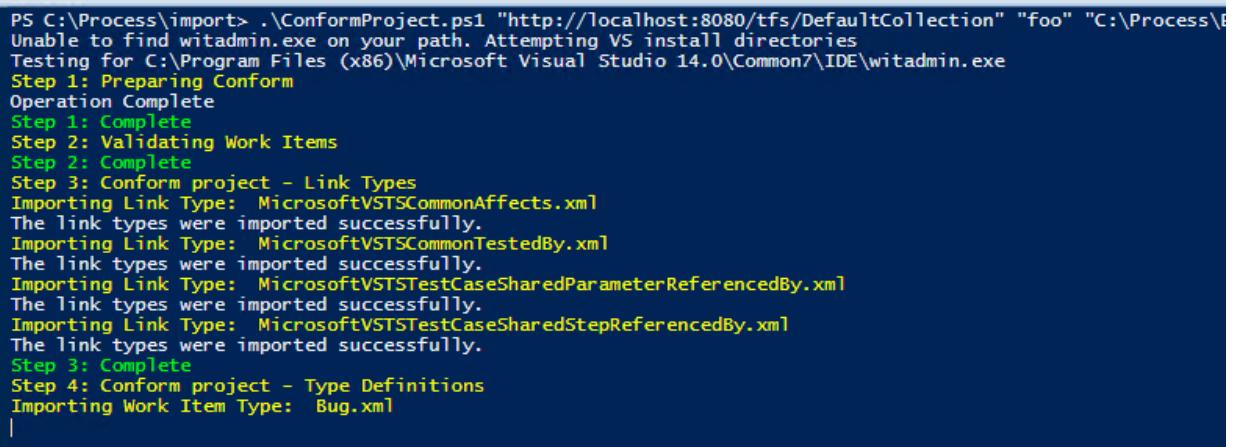
```
TfsMigrator.txt C:\Users\dahellem\AppData\Local\Temp\Temp2_20160607_122422.zip\b20160607_122422\Logs
950 [Info @17:48:07.008] Validation - *****
951 [Info @17:48:07.008] Validation - Starting validation of project 39-ClientServices, process=D:\temp\PT20160607002509619\ClientServices.zip
952 [Error @17:48:13.945] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\ClientServices.zip in path: D:\temp\PT20160607002509619\ClientServices.zip
953 [Error @17:48:13.945] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::10
954 [Info @17:48:13.945] Validation - *****
955 [Info @17:48:13.945] Validation - Starting validation of project 40-GroupStrategy, process=D:\temp\PT20160607002509619\GroupStrategy.zip
956 [Error @17:48:19.336] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GroupStrategy.zip in path: D:\temp\PT20160607002509619\GroupStrategy.zip
957 [Error @17:48:19.336] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::10
958 [Info @17:48:19.336] Validation - *****
959 [Info @17:48:19.336] Validation - Starting validation of project 41-DataOperations, process=D:\temp\PT20160607002509619\DataOperations.zip
960 [Error @17:48:28.211] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\DataOperations.zip in path: D:\temp\PT20160607002509619\DataOperations.zip
961 [Error @17:48:28.211] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::10
962 [Info @17:48:28.211] Validation - *****
963 [Info @17:48:28.211] Validation - Starting validation of project 42-HRIntegration, process=D:\temp\PT20160607002509619\HRIntegration.zip
964 [Error @17:48:37.602] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\HRIntegration.zip in path: D:\temp\PT20160607002509619\HRIntegration.zip
965 [Error @17:48:37.602] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::10
966 [Info @17:48:37.602] Validation - *****
967 [Info @17:48:37.602] Validation - Starting validation of project 43-GenY, process=D:\temp\PT20160607002509619\GenY.zip
968 [Error @17:48:44.102] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GenY.zip in path: D:\temp\PT20160607002509619\GenY.zip
969 [Error @17:48:44.102] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml::10
970 [Info @17:48:44.102] Validation - *****
971 [Info @17:48:44.102] Validation - Starting validation of project 44-EndeavourMissions, process=D:\temp\PT20160607002509619\EndeavourMissions.zip
972 [Error @17:48:50.165] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\EndeavourMissions.zip in path: D:\temp\PT20160607002509619\EndeavourMissions.zip
973 [Error @17:48:50.165] Error: Validation - ValidationFailed : Validation failed : Invalid process template: :: TF402564: You've defined 86 global lists. Only 32
Ln 965, Col 240 (448 selected) Spaces:4 UTF-8 CRLF Plain Text ☺
```

Some errors will require you to do use a `witadmin changefield` command. Changing a field name is the most common example. To save yourself some time, we recommend you run the `witadmin changefield` command and then re-run the data migration tool. Doing this will re-export the XML with the corrected names. Otherwise, you'll need to manually fix the fields in the XML syntax as well.

Once you make a fix, apply the changes back to the Azure DevOps Server. To do this, depending on the changes you made, you'll need to run one or more `witadmin` commands. To make this easier for you, we created a PowerShell script to automate the process. The script contains all of the `witadmin` commands needed to conform the entire process.

You can get the scripts at [Process Customization Scripts](#). Use the `import/ConformProject.ps1` script.

```
.\conformproject.ps1 "<collection url>" "<project name>" "<process template folder>"
```



```
PS C:\Process\import> .\ConformProject.ps1 "http://localhost:8080/tfs/DefaultCollection" "foo" "C:\Process\Import\"
Unable to find witadmin.exe on your path. Attempting VS install directories
Testing for C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\witadmin.exe
Step 1: Preparing Conform
Operation Complete
Step 1: Complete
Step 2: Validating Work Items
Step 2: Complete
Step 3: Conform project - Link Types
Importing Link Type: MicrosoftVSTSCommonAffects.xml
The link types were imported successfully.
Importing Link Type: MicrosoftVSTSCommonTestedBy.xml
The link types were imported successfully.
Importing Link Type: MicrosoftVSTSTestCaseSharedParameterReferencedBy.xml
The link types were imported successfully.
Importing Link Type: MicrosoftVSTSTestCaseSharedStepReferencedBy.xml
The link types were imported successfully.
Step 3: Complete
Step 4: Conform project - Type Definitions
Importing Work Item Type: Bug.xml
|
```

When the script has completed, re-run the data migration tool to validate the collection. Follow steps 1 through 3 until the data migration tool generates no more validation errors.

**TIP**

If you are new to XML and `witadmin`, we suggest you make one fix at a time and then conform. Continue this loop until all errors are resolved.

## Common validation errors

**VS402841: Field X in work item type Bug has syncnamechanges=false but has rules making it an identity field. Identity fields must have syncnamechanges=true. Please update your process template to include this change.**

In Azure DevOps Services we added a rule so that every identity field must have the `syncnamechanges=true` attribute. In Azure DevOps Server that rule does not apply. Therefore, the data migration tool will identify this as an issue. Don't worry, making this change on Azure DevOps Server on-prem will not cause any harm.

Run the `witadmin changefield` command. Syntax for the command looks similar to the following:

```
witadmin changefield /collection:http://AdventureWorksServer:8080/tfs/DefaultCollection /n:fieldname  
/syncnamechanges:true
```

For more information on the `witadmin changefield` command see [Manage work item fields](#).

**TF402556: For field System.IterationId to be well defined, you must name it Iteration ID and set its type to Integer.**

This error is typical for old process templates. Try running the [Configure Features Wizard](#) on each project.

Alternatively you can run the follow `witadmin` command:

```
witadmin changefield /collection:http://AdventureWorksServer:8080/tfs/DefaultCollection /n:fieldname  
/name:newname
```

**TF402571: Required element BugWorkItems is missing from Process Configuration.**

This error typically occurs when a process hasn't been updated in a while. Try running the [configure features wizard](#) on each project to resolve.

**TF402564: You've defined XX global lists. Only 64 are allowed.**

By default, Azure DevOps Services will support 64 global lists. You'll typically run across this error if you have a large amount of build pipelines. The global list named Builds - `TeamProjectName` gets created for each new build pipeline. You'll need to remove the outdated global lists.

## Related articles

- [Migration and process model FAQs](#)
- [witadmin : Customize and manage objects for tracking work](#)
- [Differences between Azure DevOps Services and Azure DevOps Server process template customizations](#)
- [Configure features after Azure DevOps Server upgrade](#)
- [Resolve validation errors](#)
- [Define global lists in Azure DevOps Server](#)
- [Process customization PowerShell scripts](#)

# Post import

12/13/2022 • 5 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

An organization is ready for use once an import has completed successfully. However, there are common tasks that you should perform before opening the organization up to all of your users. Below is a list of the most common after import tasks that should be completed. Tasks are listed in recommended order of completion.

## NOTE

It's recommended that you use the [Migration Guide](#) to progress through your import. The guide links to the technical documentation as needed.

With the release of Azure DevOps Server 2019 the TFS Database Import Service has been rebranded to become data migration tool for Azure DevOps. This includes TfsMigrator becoming the data migration tool or migrator for short. This service still works exactly the same as the old Import Service. If you're on an older version of on-premises with TFS as the branding you can still use this feature to migrate to Azure DevOps as long as you upgrade to one of the supported versions.

## Immediately after import

Immediately after the organization becomes available you will want to take a small team and perform spot checks on the organization. It's recommended that this team consists of the project collection administrators. This shouldn't be an in-depth check, but rather making sure that major pieces from your collection were brought over. Did your source code get imported? Are you seeing your build history? Are all of our area paths still present? It's best to confirm these artifacts are present before opening the organization to the entirety of your user base.

After spot checking the organization you will want to consider if you want to rename it. [Renaming an organization](#) is a simple operation, but it has [large impacts](#) on users currently using the organization. Some examples being Team Explore connections breaking or bookmarks no longer working. Getting a rename out of the way while it's just a small group of users using the organization allows the rest of the users to come in and configure their connections once.

## Set up billing

To pay for users or services in Azure DevOps Services, like hosted build and deployment agents, you need to [set up billing](#) for your organization. If you import more than one collection, you should ensure all your organizations are set up for billing with the same Azure subscription, and that your subscription is enabled for [multi-organization billing](#). You can then assign as many Basic users as you need free of charge during the calendar month in which you run the import.

## Manage users and access

Your organization includes 5 free users with [Basic](#) access. Basic includes features like Git and Team Foundation version control, tools for Agile planning and Java teams, and more. Also, you can add [Visual Studio subscribers](#) for free—they get basic features plus additional features—based on their subscription level. Also, you can add [Stakeholder](#) for free, which allows them to have partial access to Agile tools, create work items, and view backlogs and boards.

As Visual Studio subscribers log in to the organization, they are automatically detected. For all other users, you need to [assign paid access](#). Keep in mind, if you automate access using [group rules](#), the rules only apply to existing users if you [remove direct assignments](#), which were applied to users during import.

**Behavior change**—Starting between Monday, November 11th and Wednesday, November 13th, the default access behavior for imports will change. Previously, all imports tried to give users an equivalent access level post import. This means that users that had *Basic* received Basic access, and other users started with *Stakeholder* access. Once this change happens, all users will start out with free *Stakeholder* access. **You will continue to be able to assign Basic access to any users who need it at no cost, until the end of the calendar month during which your import is run.** If you have any questions or concerns about this change, feel free to [contact us](#).

## Builds

Next, you will want to configure your build agents. As part of the migration, all of your build pipelines have been brought over, but agents and pools need to be reconfigured against the new organization. Azure DevOps Services offers the ability to use a Microsoft-hosted pool of build agents that you can use, or you can connect your self-hosted build agent(s). It's important to note that only one self-hosted build agent is included for free. After that there is a [fee](#) for having additional self-hosted build agents. To pay for Microsoft-hosted and self-hosted build agents you will need to link a subscription to your organization. See the following resources for details on performing this task:

- [Build Agents](#)

If you plan on using your existing on-premises private build agents, there is one more recommended step that needs to be taken after registering them to your new organization. Clearing their cache will ensure that you don't encounter any build issues related to older TFVC or Git pointers to your on-premises collection. See [refreshing caches on client computers](#) for details on how to accomplish this task.

## Release management

If you used Release Management in Azure DevOps Server then your release pipelines and history data will be included with your import. However, like builds, you'll need to reconfigure your [agents](#) and pools against the new organization.

## Azure Artifacts

Azure Artifacts is included with Azure DevOps Services for all users granted a **Basic** license. There is no need to install an extension. Your Azure Artifacts data should be available post import.

## Azure Boards

If you have an existing GitHub Enterprise Server connection associated with your Azure DevOps Server, it will not work as expected. Work item mentions within GitHub may be delayed or never show up in Azure DevOps Services. This problem occurs because the callback URL associated with GitHub is no longer valid.

To resolve the problem, consider the following:

- **Remove and re-create the connection:** Remove and re-create the connection to the GitHub Enterprise Server repository. Follow the sequence of steps provided in [Connect from Azure Boards](#) documentation.
- **Fix the webhook url:** Go to GitHub's repository settings page and edit the webhook url to point out to the migrated Azure DevOps Services organization url:  
`https://dev.azure.com/{OrganizationName}/_apis/work/events?api-version=5.2-preview`

## Notify your teams

After getting your builds running and license subscription configured, it's recommended that the organization be opened up to all users for validation. This is when individual users can ensure that all of the content is in place, they have the right access level, and that they can pull code. Be sure to point users to our [documentation](#) on connecting to Azure DevOps Services from all of our supported IDEs and Team Explorer.

Users of TFVC with local workspaces will need to remap their workspaces against the new organization and Git users will have to reconfigure their remotes to be able to pull code.

If anything is reported as missing from the migrated organization, please reach out to [AzureDevOpsImport@microsoft.com](mailto:AzureDevOpsImport@microsoft.com). For other functional issues, please reach out to [customer support](#).

# Troubleshoot import and migration errors

12/13/2022 • 19 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

The data migration tool flags errors that you need to correct prior to performing a migration to Azure DevOps Services. This article describes the most common warnings and errors that you may receive when preparing to migrate. After correcting each error, run the **migrator validate** command again to verify resolution of all errors.

## NOTE

We recommended that you use the [Migration guide](#) to progress through your import. The guide links to the technical documentation as needed.

With the release of Azure DevOps Server 2019, the Team Foundation Server (TFS) Database Import Service was re-branded to become the data migration tool for Azure DevOps. The data migration tool, **TfsMigrator** has been renamed **migrator** for short. The service still works exactly the same as the previous import service. If you're on an older version of on-premises with TFS as the branding, you can still use **migrator** to migrate to Azure DevOps as long as you upgrade to one of the supported versions. For details, see [Migrate data from Azure DevOps Server to Azure DevOps Services](#).

## Resolve size warnings

Extra-large collections may generate one of the following messages after running the data migration tool. If you receive any of these warnings or errors, we recommend that you try to [reduce your database's size](#).

### Database size above recommended size

The following warning means you need to use the SQL Azure VM method to complete your import. Once a database reaches a certain size, it becomes faster to setup a SQL Azure VM to complete the import to Azure DevOps Services. To setup the VM and complete your import, follow the instructions linked from the warning message.

The database is currently {Database Size}GBs. This is above the recommended size of {DACPAC Size Limit}GBs to use the DACPAC import method. Please see the following page to learn how to import using a SQL Azure VM: <https://aka.ms/AzureDevOpsImportLargeCollection>

This warning DOES NOT mean that your collection is too large for import.

### Table size above recommended size

Similar to the previous warning, the following warning means you must use the SQL Azure Virtual Machine (VM) method to complete the import. Follow the instructions linked from the warning message to setup the VM and complete your import.

The largest table size is currently {Table size}GBs. This is above the recommended size of {Size limit}GBs to use the DACPAC import method. Please see the following page to learn how to import using a SQL Azure VM: <https://aka.ms/AzureDevOpsImportLargeCollection>

This warning DOES NOT mean that your collection is too large for import.

### Database metadata size above recommended size

The following warning means that your database is approaching the limit for total metadata size. Metadata size refers to the size of your database without including files, code, and other binary data. We recommend that you [reduce the size](#) of your database before import. Reducing the size provides the additional benefit of speeding up your import.

The database metadata size is currently {Metadata Size}GBs. This is above the recommended size of {Warning Size}GBs. It's recommended that you consider cleaning up older data as described in [Cleaning up old data] (/azure/devops/server/upgrade/clean-up-data).

The warning **DOES NOT** mean that your collection is too large for import, rather its metadata size is larger than the vast majority of other databases.

#### Database metadata size above maximum supported size

Unlike the previous warnings, the following error **WILL** block you from moving forward with your migration.

It indicates that the volume of metadata in your collection is too large. To proceed with the import, you need to [reduce](#) the size below the indicated limit.

The database metadata size is currently {Metadata Size}GBs. This is above the maximum supported size of {Metadata Limit}GBs.

## Resolve collation warnings

Collation warnings refer to your collection database's collation. Collations control the way string values are sorted and compared. Collections that aren't using either `SQL_Latin1_General_CI_AS` or `Latin1_General_CI_AS` will generally receive one of the **warning** messages.

#### No native support

Receiving the following warning means that you need to consider collation implications before performing the import.

The collection database's collation '{collation}' is not natively supported in Azure DevOps Services. Importing your collection will result in your collation being converted to one of the supported Azure DevOps Services collations. See more details at <https://aka.ms/AzureDevOpsImportCollations>

This warning **DOES NOT** mean that you can't import your collection.

This warning requires you to acknowledge acceptance of the warning. Accepting the warning allows the data migration tool to continue import preparations.

When you import a non-supported collation into Azure DevOps Services, the collation is transformed to a supported collation. While this transform generally works without issue, unexpected results post import or import failures could occur.

For instance, customers may notice different ordering for strings containing non-English characters. Non-English characters like 'é' may become equivalent to the English 'e' after import. It's important that you complete and verify a dry run import when importing a collection with a non-supported collation.

#### No native support, no internet connection

If the data migration tool can't connect to the internet, it can't validate conversion of your collation. It's only a warning, so you can continue with your migration process. However, when you run the `prepare` command, an internet connection is required and collation conversion is validated at that time.

The collections database's collation '{collation}' is not natively supported in Azure DevOps Services. It could not be validated that the collation can be converted during import to a supported Azure DevOps Services collation, as there was no internet connection. Please run the command again from a machine with an internet connection. See more details at <https://aka.ms/AzureDevOpsImportCollations>

## Unsupported database collation

Generally you can convert a non-supported collation to a supported collation at import time. However, some collations can't be converted. If your collection uses one of these collations, you'll receive the following **error** message.

The collection database's collation '{collation}' is not supported for import to Azure DevOps Services. It will need to be changed to a supported collation before it can be imported. See more details at <https://aka.ms/AzureDevOpsImportCollations>

In order to continue, you need to [change your collection's collation](#) to one of the supported collations on Azure DevOps Services.

## Resolve identity errors

Identity errors aren't common when validating a collection, but when they do occur you need to fix them prior to migration to avoid undesired results. Generally, identity problems stem from valid operations on previous versions of TFS that are no longer valid on your current Azure DevOps Server version. For example, while it was once allowed for some users to be members of a built-in valid users group, it isn't in the more recent versions.

The following sections provide guidance for resolving the most common identity errors.

### ISVError: 100014

This error indicates that a permission is missing from a system security group. For example, every collection that you create has Project Collection Valid Users and Project Collection Administrators groups. The system creates them by default. These groups don't support editing of their permissions.

This error indicates that one or more groups is missing a permission that it's expected to have. To resolve this error, use the **TFSSecurity.exe** command to apply the expected permissions onto the flagged system groups. Your first step is to identify which [TFSSecurity](#) command(s) you need to run.

#### Project Collection Valid Users error message

Examine the error message(s) the data migration tool highlighted. If the flagged group ends with "0-0-0-0-3", such as in the following example, you need to fix a missing permission for the **Project Collection Valid Users** group.

Run the following command, replace the scope with the one from the error message and specify your collection URL.

```
TFSSecurity.exe /a+ Identity "{scope}\\" Read sid:{Group SID} ALLOW /collection:{collectionUrl}
```

You determine the scope and group security ID (SID) from the error message.

```
ISVError:100014 Missing permission for group:Microsoft.TeamFoundation.Identity;S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-3 for scope:397c326b-b97c-4510-8271-75aac13de7a9. Expected:1 and Actual:0
```

The final command appears similar to the following entry:

```
TFSSecurity.exe /a+ Identity "397c326b-b97c-4510-8271-75aac13de7a9\\\" Read sid:S-1-9-XXXXXXXXXX-XXXXXXXXXX-  
XXXXXXXXXX-XXXXXXXXXX-0-0-0-0-3 ALLOW /collection:https://localhost:8080/defaultcollection
```

#### Project Collection Administrators error message

Carefully examine the error message(s) the data migration tool highlighted. If the flagged group that ends with "0-0-0-0-1", such as in the following example, then you will need to fix a missing permission for the **Project Collection Administrators** group. Run the following commands against **TFSSecurity.exe**, replace the scope with the one from the error message and specify your collection.

```
TFSSecurity.exe /a+ Identity "{scope}\\\\\" Read sid:{Group SID} ALLOW /collection:{collectionUrl}  
TFSSecurity.exe /a+ Identity "{scope}\\\\\" Write sid:{Group SID} ALLOW /collection:{collectionUrl}  
TFSSecurity.exe /a+ Identity "{scope}\\\\\" Delete sid:{Group SID} ALLOW /collection:{collectionUrl}  
TFSSecurity.exe /a+ Identity "{scope}\\\\\" ManageMembership sid:{Group SID} ALLOW /collection:{collectionUrl}
```

In the following example, take the scope and group **SID** from the error message and add them to the preceding command.

```
ISVError:100014 Missing permission for group:Microsoft.TeamFoundation.Identity;S-1-9-XXXXXXXXXX-XXXXXXXXXX-  
XXXXXXXXXX-XXXXXXXXXX-0-0-0-0-1 for scope:0c7c2216-fa4b-4107-a203-82b324a147ef. Expected:15 and Actual:0
```

The final command appears similar to the following entry:

```
TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\\\" Read sid:S-1-9-XXXXXXXXXX-XXXXXXXXXX-  
XXXXXXXXXX-XXXXXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection  
TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\\\" Write sid:S-1-9-XXXXXXXXXX-XXXXXXXXXX-  
XXXXXXXXXX-XXXXXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection  
TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\\\" Delete sid:S-1-9-XXXXXXXXXX-  
XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection  
TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\\\" ManageMembership sid:S-1-9-XXXXXXXXXX-  
XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection
```

When you need to correct multiple errors, we recommend that you create a batch file to automate execution of the commands. Once you've executed the commands, you need to rerun the data migration **validate** tool to verify resolution. If some errors still persist, contact [Azure DevOps Services customer support](#).

#### ISVError: 300005

ISVError: 300005 indicates that a non-group identity is a member of an everyone group, more commonly known as the Valid Users groups. Valid Users groups are default groups defined for all projects and collections. These groups are not editable. They are designed to only contain other Azure DevOps permission or security groups as members. This error indicates that an Active Directory (AD) group or user identity has a direct membership in a Valid Users group.

#### IMPORTANT

Ensure that you have a backup of your collection and configuration databases before running the following commands to resolve the error.

Since you can't directly edit Valid Users groups, you need to run a SQL statement against the configuration

database to remove the offending identity and correct the invalid membership. Carefully examine the error messages highlighted by the data migration tool. Copy the `GroupSid`, `MemberId`, and `ScopeId` as you'll need to place these values into the following command.

```
DECLARE @p6 dbo.typ_GroupMembershipTable

INSERT into @p6 values('{GroupSid}', 'Microsoft.TeamFoundation.Identity', '{MemberId}', 0)

EXEC prc_UpdateGroupMembership
@partitionId=1,@scopeId='{ScopeId}',@idempotent=1,@incremental=1,@insertInactiveUpdates=0,@updates=@p6,@eventAuthor='9EE20697-5343-43FC-8FC5-3D5D455D21C5',@updateGroupAudit=0
```

The following example lists an example of an ISVError: 300005 message from the data migration tool.

```
ISVError:300005 Unexpected non group identity was found to have direct membership to everyone group.
GroupSid:S-1-9-1551374245-3746625149-2333054533-2458719197-2313548623-0-0-0-3, MemberId:76050ddf-4fd8-48c4-a1ff-859e44364519, ScopeId:7df650df-0f8b-4596-928d-13dd89e5f34f
```

If the error message lists a `MemberSid`, you need to get the `MemberID` from the `dbo.tbl_Identity` table in the configuration database. With the `MemberID`, you can then look up the GUID for the `MemberSid`.

```
ISVError:300005 Unexpected non group identity was found to have direct membership to everyone group.
GroupSid:S-1-9-1551374245-3746625149-2333054533-2458719197-2313548623-0-0-0-3,
MemberSid:System.Security.Principal.WindowsIdentity;S-1-5-21-124525095-708259637-1543119021-1737349,
ScopeId:7df650df-0f8b-4596-928d-13dd89e5f34f
```

```
DECLARE @MemberId uniqueidentifier

SET @MemberId = (Select Id from dbo.tbl_Identity where Sid ='S-1-5-21-124525095-708259637-1543119021-1737349');

SELECT @MemberId
```

Copy the `GroupSid`, `MemberId`, and `ScopeId` into the SQL command.

```
DECLARE @p6 dbo.typ_GroupMembershipTable

INSERT into @p6 values('S-1-9-1551374245-3746625149-2333054533-2458719197-2313548623-0-0-0-3', 'Microsoft.TeamFoundation.Identity', '76050ddf-4fd8-48c4-a1ff-859e44364519', 0)

EXEC prc_UpdateGroupMembership @partitionId=1,@scopeId='7df650df-0f8b-4596-928d-13dd89e5f34f',@idempotent=1,@incremental=1,@insertInactiveUpdates=0,@updates=@p6,@eventAuthor='9EE20697-5343-43FC-8FC5-3D5D455D21C5'
```

Run the completed command against the Azure DevOps Server configuration database. You'll need to repeat this command for each ISVError: 300005 instance reported. You can batch errors with the same scope ID into a single command. Once you've executed the commands, rerun the data migration tool validate again to ensure that the errors have been corrected. If the errors still persist, contact [Azure DevOps Services customer support](#).

## IMPORTANT

To address these errors, the collection must be attached.

If you receive a -1 result when you run the command, ensure that your collection database that produced the error is attached to your Azure DevOps Server instance and that you're running the command on the configuration database.

## Azure Active Directory timeout exception

On rare occasions, you may receive an Azure Active Directory (Azure AD) timeout error when running the data migration tool **prepare** command.

```
Exception Message: Request failed (type AadGraphTimeoutException)
```

This error means that the requests to Azure AD to find the matching Azure AD identities for users in your collection timed out. Generally, you can resolve this error by waiting to run the **prepare** command at a less busy time of the day, such as after regular business hours.

In the event that the error continues, you should undertake a few troubleshooting steps. First, you will want to test your connection to Azure AD from the machine running the **prepare** command. Execute the following steps to retrieve information on a user in your Azure AD.

Open PowerShell in elevated mode and replace 'someone@somecompany.com' in the following command with your Azure AD user identity.

```
//Install the AzureAD PowerShell module - ensuring to select Yes to All
Install-Module AzureAD

// Install the MSOnline PowerShell module - ensuring to select Yes to All
Install-Module MSOnline

// Connect to Azure AD and use your Azure AD credentials (someone@somecompany.com) to login when the pop-up appears
Connect-MsolService

// Try to retrieve information on a user from your Azure AD
Get-MsolUser -UserPrincipalName someone@somecompany.com
```

If any of the above steps fail or you're unable to look up a user's identity, a connection issue may exist between the machine running the **prepare** command and Azure AD. Run a network trace while executing the **prepare** command to ensure that nothing within your network is interfering with calls reaching Azure AD. If you've confirmed that the problem isn't with your network, contact Azure support for assistance with troubleshooting. If you're able to retrieve user information, open your log file from the **prepare** attempt and look for a line similar to the following entry.

```
Number of active users is {Number of Users}.
```

If the number of active users is over 50,000, the volume of identities being mapped may require more time than provided by the timeout limit. Inspect your collection for inclusions of large AD groups such as an 'everyone' group. If possible, remove these groups and try again. If you still can't resolve this error, contact [Azure DevOps Services customer support](#).

## Resolve process errors

See the separate [Process Templates](#) page for details on resolving common process errors.

# Resolve field validation errors

## VS403310

The following error message can occur when an inconsistency in collection files is detected. Contact customer support if you encounter this error.

```
VS403310: An inconsistency was detected in some of the files in the collection.
```

## VS403442

Field name conflicts sometimes occur between your local collection and an Azure DevOps Services system field.

```
In order to migrate successfully, you must rename field *{TFSfieldReferenceName}*.
Given name *{TFSfieldName}* is reserved for field *{VSTSfieldReferenceName}*. 
```

To resolve this error, change the name of your collection field. Use the **witadmin changefield** command from [witadmin](#).

```
witadmin changefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldReferenceName
/name:newFieldName
```

## VS403443

The following error indicates a field name conflict exists between your local collection and a specific Azure DevOps Services field.

```
In order to migrate successfully, you must rename field *{TFSfieldReferenceName}* to *{VSTSfieldName}*. Given
name for *{TFSfieldReferenceName}* is *{TFSfieldName}* 
```

To resolve this error, use the **witadmin changefield** command. For details, see [witadmin](#).

```
witadmin changefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldReferenceName
/name:VSTSfieldName
```

## VS403444

The following error indicates a field type conflict exists between your local collection and Azure DevOps Services.

Using [witadmin](#), you can change the data type only for HTML or PlainText fields.

```
In order to migrate successfully, you must set type of field *{TFSfieldReferenceName}* to *{Type}*. Given
type for *{TFSfieldReferenceName}* is *{collectionType}*. 
```

If your field type is HTML or PlainText, then you can change its type to the required type.

```
witadmin changefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldReferenceName
/type:PlainText | HTML
```

### NOTE

If your field type is something different than HTML or PlainText and field data isn't important or the field isn't used in any project, then we recommend you delete the field.

```
witadmin deletefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldReferenceName
```

## IMPORTANT

Deleting a field results in a loss of field data across the collection.

# Resolve import errors

Failures that occur during import fall into one of two categories, [verification failure](#) and [import failure](#).

## Verification failures

Verification failures occur when the import fails to start. The data migration tool attempted to queue an import, but returned an error instead. Verification failure issues indicate that your import request isn't valid. Address the error messages you receive according to the following guidance and then try to import again.

### VS403254

The region that you entered for your Azure DevOps Services import isn't supported.

`VS403254: Region {0} may not be used for the Import, it is not a supported region.`

Open your import specification file and update the region that you've provided with the correct short name for the [region](#).

### VS403249

The organization name your team has selected is already in use by an existing organization. All Azure DevOps Services imports go into a new organization that is created at import time.

`VS403249: The organization {0} already exists. Please select a different name and try the import again.`

Select a different organization name and update the import specification file before retrying the import.

### VS403250 & VS403286

The DACPAC isn't built off a detached collection.

`VS403250: The dacpac is not a detached Azure DevOps Server Collection database.`

`VS403286: The dacpac is from a Azure DevOps Server Configuration database. You must use a detached Azure DevOps Server Collection database.`

[Detach](#) your collection database and generate the DACPAC again.

### VS403243

Unable to make a connection to the database using the provided SQL Connection String.

`VS403243: Unable to connect to the database using the provided SQL Connection String {0}.`

Review the parameters that were provided to ensure they're correct and try again.

### VS403260 & VS403351

The collection database isn't detached.

`VS403260: The database is not detached.`

`VS403351: The DACPAC or source database is missing an expected table. It's possible that the database was not correctly detached from Azure DevOps Server.`

[Detach](#) your collection database and retry the import queue.

### VS403261

The connection string must be encrypted otherwise the password is sent in the clear.

VS403261: The SQL connection string must use encryption.

Add `Encrypt=true` to your SQL connection string.

## VS403262

The connection string must use SQL Authentication.

VS403262: The SQL connection string must use SQL Authentication, Integrated Authentication is not supported.

Add `Integrated Security=False` to your SQL connection string.

## VS403263

Your SQL sign in user account doesn't have the required database role.

VS403263: The User ID {0} must be member of the database role {1}.

Make sure the user account for sign in is assigned the '[TFSEXECROLE](#)' role.

### NOTE

There is a known issue with using `sp_addrolemember` to add `TFSEXECROLE` to an existing SQL login. The role membership isn't applied until all open connections using that identity are closed. If you receive the VS403263 error and have confirmed your identity has the role, we recommend that you create a new identity for your import. Details on how to create a new SQL login that's ready to be used for import can be found at [Import large collections](#).

## VS403264

The connection string doesn't point to an Azure DevOps Server collection database.

VS403264: The database is not a Azure DevOps Server Collection database, it cannot be used for import.

Verify or correct the connection string points to your collection database.

## VS40325

The Azure DevOps Server Update has queued the file migration job. Imports can't be performed until this job has completed. The completion time for this job is dependent on the size of the collection.

VS403255: The collection cannot be imported due to an ongoing post upgrade job. Please wait and try again later

You can track job progress by running the following query on the collection database:

```
SELECT COUNT (*) as remaining_files_to_migrate
FROM   tbl_FileReference
WHERE  PartitionId > 0
      AND Migrate fileId IS NOT NULL
```

Once the number of files remaining to migrate is zero, you can run the data migration tool.

## VS403282

A new line character exists in the source location value. This character could have remained after copying the SAS key from your windows console.

VS403282: The source location parameter contains a new line character. Please ensure the SAS key is defined on a single line in the import specification file.

Remove the line break and try again.

## VS403271

Your import files and DACPAC aren't located in the **required** Azure region to complete the import to your target Azure DevOps Services region.

VS403271: It appears that your DACPAC was uploaded to East US. It's required that customers targeting Central US for import put their DACPACs in Central US. Please move your DACPAC to Central US and requeue the import.

Create a new Windows Azure storage account in the required region and copy your files. The following example shows how to copy your data using AzCopy.

```
AzCopy.exe /Source:https://accountSCUS.blob.core.windows.net/mycontainer /SourceKey:"primary access key"  
/Dest:https://accountCUS.blob.core.windows.net/mycontainer /DestKey:"primary access key" /S
```

## VS403316

Inconsistencies were detected in some Team Foundation version control (TFVC) files within your collection.

VS403316: An inconsistency was detected in some TFVC files for this collection. The inconsistency needs to be corrected prior to running an import to Azure DevOps Services. Please reach out to <https://aka.ms/AzureDevOpsImportSupport> for assistance with addressing this issue.

Work with Azure DevOps Services [customer support](#). Open a support ticket and they'll work with you to resolve the error.

## VS403366

The data migration tool was unable to connect to the SQL Azure VM.

VS403366: A problem occurred while attempting to connect to your database. Please verify that your connection string is correct and that all required IP addresses for Azure DevOps Services have been provided exceptions for your machines firewall.

List of Azure DevOps Services IPs:

Verify that you've entered the information correctly in your connection string and that you can connect to the VM.

The IPs that the error message lists are for Azure DevOps Services. Azure DevOps Services IPs can change temporarily during deployments. Add them to your firewall exceptions and try queuing the import again. For a list of IP addresses, see [Import large collections, Restrict access to Azure DevOps Services IPs only](#)

## VS403373

The data migration tool doesn't support importing multiple copies of the **SAME** collection. However, it **DOES** support importing **split** copies of a collection. Change the GUID for the *DataImportCollectionID*.

From SQL Server Management Studio (SSMS), open the extended properties for the split copies that you haven't imported yet. Add a newly generated GUID to the "TFS\_DATAIMPORT\_COLLECTIONID" property. Then rerun the **prepare** command and use the new **import.json** file to queue the import.

## VS403379

Data import will fail as one or more projects found in this collection are in the soft-deleted stage. Please restore the soft-deleted project(s) or delete them permanently before running the data import. For details, see [Delete a project](#).

VS403379: Data import will fail as one or more projects found in this collection are in the soft-deleted stage. Please restore the soft-deleted project(s) or delete them permanently before running the data import.

Verify the collection against which you are running the data migration tool has projects in the soft-deleted stage. Once a project is deleted, it remains in a soft-delete state for 28 days during which the deleted project can be restored. You can read about how to restore a deleted project in [Restore a project](#). If you have projects in the soft-deleted stage, remove them completely or restore them back before running data import.

### Import failures

Import failures happen when the data migration tool successfully queues an import, but fails to complete the import. The individual that queued the import receives a failure email notification. Most of the time this email includes a reason for the failure. If it does, use the troubleshooting steps provided in the email and this page to resolve the errors and retry your import.

If the error is more complex, then the email you receive provides instructions on how to file a [customer support case](#). After submitting a customer support case, your team will need to roll back by bringing your Azure DevOps Server instance back online and reattach your collection. Your team members can then continue working. We recommended you not attempt the import again until the failure causing issue is resolved.

## Related articles

- [Validate and import](#)
- [Post-import](#)
- [Delete a project](#)
- [Restore a project](#)



# Default permissions quick reference for Azure DevOps

12/13/2022 • 15 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

To use Azure DevOps features, users must be added to a security group with the appropriate permissions and granted access to the web portal. Limitations to select features are based on the *access level* and *security group* to which a user is assigned. The **Basic** access level and higher supports full access to most Azure DevOps Services, except for Azure Test Plans. **Stakeholder** access level provides partial support to Azure Boards and Azure Pipelines. To learn more about access levels, see [About access levels](#) and [Stakeholder access quick reference](#).

## Assign users to a security group

The most common built-in security groups—**Readers**, **Contributors**, and **Project Administrators**—and team administrator role grant permissions to specific features.

In general, use the following guidance when assigning users to a security group:

- Add to the **Contributors** security group full-time workers who contribute to the code base or manage projects.
- Add to the **Project Administrators** security group users tasked with managing project resources.
- Add to the **Project Collection Administrators** security group users tasked with managing organization or collection resources.

To learn more about administrative tasks see [About user, team, project, and organization-level settings](#). For a complete reference of all built-in groups and permissions, see [Permissions and groups](#). For information about access levels, see [About access levels](#).

In the tables provided in this article, a (checkmark) indicates that the corresponding access level or security group has access to a feature by default.

To assign or change an access level, see [Add users and assign licenses](#). If you need to [grant specific users select permissions](#), you can do so.

## Azure Boards

You can plan and track work from the web portal **Boards** hub, and using Visual Studio, Excel, and other clients. For an overview of work tracking features, see [About Agile tools](#). To change permissions, see [Set permissions and access for work tracking](#). In addition to the permissions set at the [project level via the built-in groups](#), you can set permissions for the following objects: [area and iteration paths](#) and individual [queries and query folders](#).

## Work tracking

You can plan and track work from the web portal **Work** hub, and using Eclipse, Visual Studio, Excel, Project, and other clients.

**NOTE**

Team administrators can configure settings for their team's tools. Organization owners and members of the Project Administrators group can configure settings for all teams. To be added as an administrator, see [Add team administrators](#) or [Change project-level permissions](#).

Access to the following tasks are controlled by each user's access level or by permission assignments. Members of the Readers, Contributors, or Project Administrators group are assumed to have Basic access or greater.

**General work item permissions**

You can use work items to track anything you need to track. To learn more, see [Understand how work items are used to track issues, tasks, and epics](#).

**NOTE**

You can change the work item type or move work items to another project within a project collection. These features require that the data warehouse is disabled. With the data warehouse disabled, you can use the [Analytics Service](#) to support your reporting needs. To learn more about disabling the data warehouse, see [Disable the data warehouse and cube](#).

**Task or permission**

**Readers**

**Contributors**

**Project admins**

**View work items in this node** (Area Path permission)

- ✓
- ✓
- ✓

**Edit work items in this node** (Area Path permission)

- ✓
- ✓

**Create tag definition**

- ✓
- ✓

**Change work item type** (Project-level permission)

- ✓
- ✓

**Move work items out of this project** (Project-level permission)

- ✓
- ✓

**Email work items**

✓

✓

✓

Apply a work item template

✓

✓

**Delete and restore work items** (Project-level permission) (able to restore from the Recycle bin)

✓

✓

**Permanently delete work items** (Project-level permission)

✓

[Provide feedback](#) (through the Microsoft Feedback client)

✓

✓

[Request feedback](#)

✓

✓

#### NOTE

Work items are subject to rules applied to them. Conditional rules based on user or group membership are cached for your web browser. If you find yourself restricted to update a work item, you may have encountered one of these rules. If you believe you've encountered an issue that doesn't apply to you, see [Work item form IndexDB caching issues](#). To learn more about conditional rules, see [Rules and rule evaluation](#).

## Boards

You use **Boards** to implement Kanban methods. Boards present work items as cards and support quick status updates through drag-and-drop.

### Task

### Readers

### Contributors

### Team admins

### Project admins

View boards and open work items

✓

✓

✓

Add work items to a board; update status through drag-and-drop



Reorder work items or reparent child items through drag-and-drop; update a field on a card



Add work items to a board; update status, reorder, or reparent child items through drag-and-drop; update a field on a card



Add child items to a checklist



Assign to a sprint (from card field)



Configure board settings



### **Backlogs features access**

[Backlogs](#) display work items as lists. A product backlog represents your project plan and a repository of all the information you need to track and share with your team. Portfolio backlogs allow you to group and organize your backlog into a hierarchy.

**Task**

**Readers**

**Contributors**

**Team admins**

**Project admins**

View backlogs and open work items



Add work items to a backlog



Use bulk edit features





Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane; Assign items to a sprint using the Planning pane



Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane; Assign items to a sprint using drag-and-drop



Configure team settings, backlog levels, show bugs, work days off



## Sprints

You use sprint tools to implement Scrum methods. The **Sprints** set of tools provide filtered views of work items that a team has assigned to specific iteration paths or sprints.

**Task**

**Readers**

**Contributors**

**Team admins Project admins**

View sprint backlogs, taskboards, and open work items



Add work items to a sprint backlog or taskboard



Prioritize/reorder a sprint backlog or taskboard; add child items to a backlog item; reassign items to a sprint using the Planning pane



View team capacity and work details



Set team capacity



Use bulk edit features



Define team sprints



## Queries

[Queries](#) are filtered lists of work items based on criteria that you define by using a query editor. [Adhoc searches](#) are powered by a semantic search engine.

Task

Readers

Contributors

Project admins

View and run managed queries, view query charts



Create and save managed My queries, query charts



Create, delete, and save Shared queries, charts, folders



## Delivery plans

[Delivery plans](#) display work items as cards against a calendar view. This format can be an effective communication tool with managers, partners, and stakeholders for a team.

Task

Readers

Contributors

Team admins

Project admins

View delivery plans



Create, edit, or delete a delivery plan, Contributors can only edit or delete plans that they create





Manage permissions for a delivery plan, Contributors can only manage permissions for plans that they create



## Azure Repos

You can manage your source code from the web portal **Repos** hub, or using Xcode, Eclipse, IntelliJ, Android Studio, Visual Studio, or Visual Studio Code.

Stakeholders for private projects have no access to **Repos**. Stakeholders for public projects have the same access to **Repos** as **Contributors**.

## Code: Source control

You can connect to your code from the web portal **Code** hub, or using Xcode, Eclipse, IntelliJ, Android Studio, Visual Studio, or Visual Studio Code. Stakeholders for private projects have no access to **Code**.

### Git

You can use [Git repositories](#) to host and collaborate on your source code. For an overview of code features and functions.

#### Permission

##### Readers

##### Contributors

##### Build Admins

##### Project Admins

---

**Read** (clone, fetch, and explore the contents of a repository); also, can create, comment on, vote, and **Contribute to pull requests**



**Contribute, Create branches, Create tags, and Manage notes**



**Create repository, Delete repository, and Rename repository**



**Edit policies, Manage permissions, Remove others' locks**



**Force push** (rewrite history, delete branches and tags)



**Bypass policies when completing pull requests**

(not set for any security group)

**Bypass policies when completing pull requests, Bypass policies when pushing, Force push** (rewrite history, delete branches and tags)

(not set for any security group)

---

## TFVC

Team Foundation Version Control (TFVC) provides a centralized version control system to manage your source control.

### NOTE

Tasks such as create, delete, or rename a TFVC repository are not supported. Once a TFVC repository is created you can't delete it. Also, you can only have one TFVC repository per project. This is different from Git repositories which allow for adding, renaming, and deleting multiple repositories.

## Permission

Readers

Contributors

Build Admins

Project Admins

**Check in, Label, Lock, Merge, Pend a change in a server workspace, Read**

Read only



**Administer labels, Manage branches, Manage permissions, Revise other users' changes, Undo other users' changes, Unlock other users' changes**



## Azure Pipelines

You can define and manage your builds and releases from the web portal **Pipelines** hub. For an overview of pipelines features and functions, see [Continuous integration on any platform](#).

TASK	READERS	CONTRIBUTORS	BUILD ADMINS	PROJECT ADMINS	RELEASE ADMINS
View release pipelines	✓	✓	✓	✓	✓

Task	Readers	Contributors	Build Admins	Project Admins	Release Admins
Define builds with continuous integration		✓	✓	✓	
Define releases and manage deployments		✓		✓	✓
Approve releases		✓	✓	✓	✓
Azure Artifacts (5 users free)		✓		✓	✓
Queue builds, edit build quality		✓	✓	✓	
Manage build queues and build qualities			✓	✓	
Manage build retention policies, delete and destroy builds		✓	✓	✓	
Administer build permissions			✓	✓	
Manage release permissions				✓	✓
Create and edit task groups		✓	✓	✓	✓
Manage task group permissions			✓	✓	✓
Can view library items such as variable groups	✓	✓	✓	✓	✓
Use and manage library items such as variable groups			✓	✓	✓

## Build

### Task

#### Readers

#### Contributors

## Build admins

### Project admins

View builds

- ✓
- ✓
- ✓
- ✓

View build pipeline

- ✓
- ✓
- ✓
- ✓
- ✓

Administer build permissions

- ✓
- ✓

Delete or Edit build pipeline

- ✓
- ✓
- ✓

Delete or Destroy builds

- ✓
- ✓

Edit build quality

- ✓
- ✓
- ✓

Manage build qualities

- ✓
- ✓

Manage build queue

- ✓
- ✓

Override check-in validation by build

✓

Queue builds

✓

✓

✓

Retain indefinitely

✓

✓

✓

✓

Stop builds

✓

✓

Update build information

✓

## **Release**

Task

Stakeholders

Readers

Contributors

Project Admins

Release

Admins

Approve releases

✓

✓

✓

✓

View releases

✓

✓

✓

✓

✓

View release pipeline

- ✓
- ✓
- ✓
- ✓

Administer release permissions

- ✓
- ✓

Delete release pipeline or release stage

- ✓
- ✓
- ✓

Delete releases

- ✓
- ✓
- ✓

Edit release pipeline

- ✓
- ✓

Edit release stage

- ✓
- ✓
- ✓

Manage deployments

- ✓
- ✓

Manage release approvers

- ✓
- ✓
- ✓

Manage releases

- ✓
- ✓

## Task groups

You use task groups to encapsulate a sequence of tasks already defined in a build or a release pipeline into a single reusable task. Task group permissions follow a hierarchical model. You can set defaults for all permissions at the project-level and over-write on an individual task group pipeline. You [define and manage task groups](#) in the **Task groups** tab in Azure Pipelines.

TASK	READERS	CONTRIBUTORS	BUILD ADMINS	PROJECT ADMINS	RELEASE ADMINS
Administer task group permissions			✓	✓	✓
Delete task group			✓	✓	✓
Edit task group			✓	✓	✓

## Build and Release

You can define and manage your builds and releases from the web portal, **Build and Release**. For an overview of pipelines features and functions, see [Continuous integration on any platform](#). From the web portal, you can set permissions for all or individual builds and releases. See [Set build and release permissions](#).

### Build

Task

Readers

Contributors

Build

Admins

Project Admins

View builds

- ✓
- ✓
- ✓
- ✓
- ✓

View build definition

- ✓
- ✓
- ✓
- ✓
- ✓

Administer build permissions

- ✓

✓

Delete or Edit build definitions

✓

✓

✓

Delete or Destroy builds

✓

✓

Edit build quality

✓

✓

✓

Manage build qualities

✓

✓

Manage build queue

✓

✓

Override check-in validation by build

✓

Queue builds

✓

✓

✓

Retain indefinitely

✓

✓

Stop builds

✓

✓

Update build information

✓

## **Release**

Task

Readers

Contributors

Project Admins

Release

Admins

Approve releases

- ✓
- ✓
- ✓

View releases

- ✓
- ✓
- ✓
- ✓
- ✓

View release definition

- ✓
- ✓
- ✓
- ✓
- ✓

Administer release permissions

- ✓
- ✓

Delete release definition or release stage

- ✓
- ✓
- ✓

Delete releases

- ✓
- ✓
- ✓

Edit release definition

- ✓
- ✓

Edit release stage

✓

✓

✓

Manage deployments

✓

✓

Manage release approvers

✓

✓

✓

Manage releases

✓

✓

## Azure Test Plans

Users granted **Basic + Test Plans** or **Visual Studio Enterprise** access level can define and manage manual tests from the web portal. For an overview of manual test features and functions, see [Testing overview](#). You set several [test permissions at the project level](#) from [Project Settings>Permissions](#).

## Test

Users granted **Visual Studio Enterprise** or **Advanced** access level can define and manage manual tests from the web portal. For an overview of manual test features and functions, see [Testing overview](#). You set several [test permissions at the project level](#) from [Project Settings>Permissions](#).

Permission

Level

Readers

Contributors

Project Admins

**View test runs**

Project-level

✓

✓

✓

**Create test runs**

**Delete test runs**

Project-level



**Manage test configurations**

**Manage test environments**

Project-level



**Create tag definition**

**Delete and restore work items**

Project-level



**Permanently delete work items**

Project-level



**View work items in this node**

Area Path



**Edit work items in this node**

**Manage test plans**

**Manage test suites**

Area Path



**NOTE**

The **Change work item type** permission doesn't apply to test-specific work items. Even if you choose this feature from the work item form, changing the work item type is disallowed.

## Azure Artifacts

You can manage feeds from the web portal, [Artifacts](#). Users granted Stakeholder or Basic access, or higher can access Azure Artifacts features. To set permissions, see [Secure feeds using permissions](#).

You can manage feeds from the web portal, [Artifacts](#). Users granted Basic access or higher can access Azure Artifacts features. Users granted Stakeholder access have no access to Azure Artifacts. To set permissions, see [Secure feeds using permissions](#).

# Package management

You can manage feeds from the web portal, **Build and release > Packages**. Users granted Basic access or higher can access Package management features. Users granted Stakeholder access have no access. To set permissions, see [Secure feeds using permissions](#).

Feeds have four permission roles: Readers, Collaborators, Contributors, and Owners. Owners can add user accounts or security groups to any role.

PERMISSION	READER	COLLABORATOR	CONTRIBUTOR	OWNER
List, install, and restore packages	✓	✓	✓	✓
Push packages			✓	✓
Unlist/deprecate packages			✓	✓
Delete/unpublish package				✓
Promote a package to a view			✓	✓
Add/remove upstream sources				✓
Save packages from upstream sources		✓	✓	✓
Edit feed permissions				✓

By default, the Project Collection Build Service is a Contributor and your project team is a Reader.

## NOTE

To access a feed in a different organization, a user must be given access to the project hosting that feed.

Feeds have three permission roles: Readers, Contributors, and Owners. Owners can add user accounts or security groups to any role.

PERMISSION	READER	CONTRIBUTOR	OWNER
List and restore/install packages	✓	✓	✓
Push packages		✓	✓
Unlist/deprecate packages		✓	✓
Delete/unpublish package			✓
Edit feed permissions			✓

PERMISSION	READER	CONTRIBUTOR	OWNER
Rename and delete feed			✓

By default, the Project Collection Build Service is a Contributor and your project team is a Reader.

**NOTE**

To access a feed in a different organization, a user must be given access to the project hosting that feed.

## Notifications, alerts, and team collaboration tools

To manage notifications, see [Manage personal notifications](#) and [Manage team notifications](#).

**NOTE**

There are no UI permissions associated with managing notifications. Instead, you can manage them using the [TFS Security command line tool](#).

Task

Readers

Contributors

Team admins

Project admins **Project Collection admins**

---

View the project page, navigate using the project page

- ✓
- ✓
- ✓
- ✓

Edit the project page

- ✓

Set personal notifications or alerts

- ✓
- ✓
- ✓

Set team notifications or alerts

- ✓
- ✓

Set project-level notifications or alerts

- ✓

View Project READMEs

- ✓
- ✓
- ✓
- ✓

View Project wikis or code wikis

- ✓
- ✓
- ✓
- ✓
- ✓

Provision or create a project wiki

- ✓
- ✓
- ✓
- ✓

Publish code as a wiki

- ✓
- ✓
- ✓
- ✓

Request feedback

- ✓
- ✓
- ✓
- ✓

Provide feedback

- ✓
- ✓
- ✓
- ✓

Search across projects, organizations, collections

- ✓
- ✓
- ✓
- ✓

Dashboards, charts, reports, and widgets

You can define and manage team and project dashboards from the web portal, **Dashboards**. For an overview of dashboard and chart features, see [Dashboards](#). You can set [individual dashboard permissions](#) to grant or restrict the ability to edit or delete dashboards.

Users granted Stakeholder access to private projects can't view or create query charts. Stakeholder access to public projects can view and create query charts.

You can define and manage team dashboards from the web portal, **Dashboards**. For an overview of dashboard and chart features, see [Dashboards](#). You set [dashboard permissions at the team level](#) from the team dashboard page.

---

## Task

### Readers

### Contributors

### Team admins

### Project admins

---

View team and project dashboards

- ✓
  - ✓
  - ✓
  - ✓
- 

View team dashboards

- ✓
  - ✓
  - ✓
- 

Add and configure project dashboards

- ✓
  - ✓
- 

Add and configure team dashboards

- ✓
- ✓
- ✓

## Power BI Integration and Analytics views

From the web portal **Analytics views**, you can create and manage Analytics views. An Analytics view provides a simplified way to specify the filter criteria for a Power BI report based on the Analytics Service data store. The Analytics Service is the reporting platform for Azure DevOps. To learn more, see [What is the Analytics Service?](#).

You set [permissions](#) for the service at the project level, and for shared Analytics views at the object level. Users with Stakeholder access have no access to view or edit Analytics views.

## Task

Readers

Contributors

Project admins

View Analytics



View a shared Analytics view



Add a private or shared Analytics view



Edit and delete shared Analytics views



## Related articles

- [Add users to a project or team](#)
- [Security and permission management tools](#)
- [Permissions and groups reference](#)
- [About access levels](#)
- [Web portal navigation](#)
- [Troubleshoot permissions](#)

# About access levels

12/13/2022 • 9 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

Access levels grant or restrict access to select web portal features. This is in addition to permissions granted through security groups, which provide or restrict specific tasks. Access levels enable administrators to provide their user base access to the features they need and only pay for those features.

## IMPORTANT

### Version

Azure DevOps Services

Azure DevOps Services

Azure DevOps Server 2022

Azure DevOps Server 2020

Azure DevOps Server 2019

TFS 2018

Previous versions

To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents. Feature support differs depending on whether you are working from Azure DevOps Services or an on-premises version of Azure DevOps Server.

To learn which on-premises version you are using, see [Look up your Azure DevOps platform and version](#)

When you add a user or group to a team or project, they're automatically granted access to those features supported by the default access level and those supported by the security group to which they are added. Most users can access most features by being assigned to the **Basic** access level and **Contributors** security group. For a simplified overview of the permissions assigned to the most common groups **Readers**, **Contributors**, and **Project Administrators**, see [Default permissions](#).

To add user accounts or groups to specific access levels, see [Manage users and access](#). Make sure to set each user's access level based on what you've purchased for that user.

To add user accounts or groups to specific access levels, see [Change access levels](#). Make sure to set each user's access level based on what you've purchased for that user.

## Supported access levels

Assign users or groups of users to one of the following access levels:

- **Basic:** Provides access to most features. Assign to users with a Visual Studio Professional subscription, an Azure DevOps Server CAL, and to users for whom you're paying for Basic access in an organization.
- **Basic + Test Plans:** Provides access to all features included in **Basic**, as well as Azure Test Plans. Assign to users with a Visual Studio Test Professional or MSDN Platforms subscription, and to users for whom you're paying for Basic + Test Plans access in an organization.

- **Stakeholder:** Can be assigned to unlimited users for free. Provides partial access to private projects and mostly full access to public projects. Assign to users with no license or subscriptions who need access to a limited set of features. For feature access details, see [Stakeholder access quick reference](#).
- **Visual Studio Subscriber:** Assign to users who already have a Visual Studio subscription. The system automatically recognizes the user's subscription—Visual Studio Enterprise, Visual Studio Professional, Visual Studio Test Professional, or MSDN Platform—and enables any other features that are included in their subscription level. If you assign **Basic** or **Stakeholder**, they also receive their Visual Studio subscription benefits upon sign-in.

**TIP**

As a best practice when adding new users, we recommend assigning the **Visual Studio Subscriber** level when appropriate (as opposed to **Basic**) to prevent being charged the **Basic** rate before the user signs in for the first time.

- **Stakeholder:** Provides partial access, can be assigned to unlimited users for free. Assign to users with no license or subscriptions who need access to a limited set of features.
- **Basic:** Provides access to most features. Assign to users with an Azure DevOps Server CAL, with a Visual Studio Professional subscription, and to users for whom you're paying for Basic access in an organization.
- **Basic + Test Plans:** Provides access for users who have a monthly Test Manager subscription, Visual Studio Test Professional, or MSDN Platforms subscription.
- **VS Enterprise:** Provides access to premium features. Assign to users with a subscription to Visual Studio Enterprise.
- **Stakeholder:** Provides partial access, can be assigned to unlimited users for free. Assign to users with no license or subscriptions who need access to a limited set of features.
- **Basic:** Provides access to most features. Assign to users with a CAL or with a Visual Studio Professional subscription.
- **Advanced** (legacy access level, deprecated in Azure DevOps Server 2019): Provides access to premium features. Only assign to users with a subscription to MSDN Platforms or Visual Studio Test Professional.
- **VS Enterprise:** Provides access to premium features. Assign to users with a subscription to Visual Studio Enterprise.

The following table indicates those features available for each supported access level. Visual Studio Test Professional and MSDN Platform subscriptions grant access to the same features as Visual Studio Enterprise.

---

**Feature**

**Stakeholder**

**Basic &  
Visual Studio Professional**

**Basic + Test Plans &  
Visual Studio Enterprise**

---

**Feature**

**Stakeholder**

**Basic &  
Visual Studio Professional**

**Basic + Test Plans &  
Visual Studio Enterprise**

---

## Feature

### Stakeholder

Basic &

Visual Studio Professional

Advanced &

Visual Studio Enterprise

---

### Administer organization

Can configure resources when also added to a security group or role: team administrator, Project Administrator, or Project Collection Administrator.



### Advanced backlog and sprint planning tools

Includes full access to all [backlog](#) and [sprint planning](#) tools.



### Advanced home page

Includes [access to projects, work items, and pull requests defined across projects you work in](#).



### Advanced portfolio management

Includes full access to define features and epics from a [portfolio backlog](#) or [Kanban board](#).



### Agile boards

Stakeholders have limited access to [Kanban boards](#) and [Taskboards](#). Stakeholders can add work items and update status through drag-and-drop, but can't update fields displayed on cards (except for the work item State) and can't [view or set capacity](#).



### Agile boards

Stakeholders have limited access to [Kanban boards](#) and [Taskboards](#). Stakeholders can't add work items, drag-and-drop cards to update status, update fields displayed on cards, nor [view or set capacity](#).





## Agile Portfolio Management

Includes limited access to [portfolio backlogs](#) and [Kanban boards](#). Stakeholders can't change the backlog priority order, can't assign items to an iteration, use the mapping pane, or exercise forecasting.



## Artifacts

Includes full access to all Azure Artifacts features, up to 2 GiB free storage.



---

## Author Release Pipelines and Manage Releases

Includes defining [release pipelines](#), [multi-stage continuous deployment \(CD\) pipelines](#), and [using approvals and gates to control deployments](#); when the [Free access to Pipelines Preview feature is enabled](#), Stakeholders gain access to all Azure Pipelines features.



## Basic backlog and sprint planning tools

Includes limited access to add and modify items on [backlogs](#) and [sprint backlogs and taskboards](#). Stakeholders can't assign items to an iteration, use the mapping pane, or forecasting.



## Build

Includes full access to all features to [manage continuous integration and continuous delivery of software](#).



## Chart Authoring

Can create work tracking [query charts](#).



---

## Chart Viewing

Can only view work tracking query charts. Stakeholders can't view query charts from the Queries page, however can view them when added to a dashboard.



---

## Code

Includes full access to all features to manage code using [Git repositories](#) or using [Team Foundation Version Control \(TFVC\)](#) Team Foundation Version Control (TFVC).



### [Delivery Plans](#)

Includes full access to add and view Delivery plans.



### [Delivery Plans](#)

Includes full access to add and view Delivery plans.



**Request and Manage Feedback** Includes full access to request and manage feedback on working software.



### **Standard Features**

Includes [working across projects](#), [View dashboards](#), [View wikis](#), and [Manage personal notifications](#). Stakeholders can't view Markdown README files defined for repositories and can only read wiki pages.



### **Test services in build and release**

Includes [running unit tests with your builds](#), [reviewing](#), and [analyzing](#) test results.



### **Test Case Management**

Includes [adding test plans and test suites](#), [creating manual test cases](#), [deleting test artifacts](#), and [testing different configurations](#).



### **Test Execution and Test Analysis**

Includes running [manual](#), [tracking test status](#), and [automated tests](#).



### **Test summary access to Stakeholder license**

Includes [requesting Stakeholder feedback using the Test & Feedback extension](#).



---

### View My Work Items

Access to [add and modify work items](#), [follow work items](#), [view and create queries](#), and [submit, view, and change feedback responses](#). Stakeholders can only assign existing tags to work items (can't add new tags) and can only save queries under My Queries (can't save under Shared Queries).



---

### View Releases and Manage Approvals

Includes [viewing releases](#) and [approving releases](#); when the [Free access to Pipelines Preview feature is enabled](#) feature is enabled, Stakeholders gain access to all Azure Pipelines features.



---

## Visual Studio subscription access

Visual Studio subscribers are entitled to **Visual Studio subscription** features as a subscriber benefit. When you add those users, be sure to assign them the **Visual Studio subscription** access level.

The system automatically recognizes their subscription and enables any other features that are included, based on their subscription level.

## VS Enterprise access

Visual Studio Enterprise subscribers are entitled to **VS Enterprise** access as a subscriber benefit. When you add those users, be sure to assign them the **VS Enterprise** access level.

With VS Enterprise access, users have access to any fee-based, Marketplace extension published by Microsoft Marketplace extension published by Microsoft that is included for active Visual Studio Enterprise subscribers.

## Advanced access

Users assigned Advanced access can manage test cases when you have [purchased the Test Manager extension](#) for Azure Test Plans and assigned to the user accounts to gain full access to [Web-based test case management tools](#).

Users assigned Advanced access have all the Basic features, plus [web-based test case management tools](#). You can [buy monthly access](#) or add users who already have a Visual Studio Test Professional with MSDN or MSDN Platforms subscription.

## Programmatic mapping of access levels

You can manage access levels programmatically using the `az devops user add` (Azure DevOps Services only) or

the [User Entitlement - Add REST API](#). The following table provides a mapping of the access level selected through the user interface and the `AccountLicenseType`, `licensingSource`, and `msdnLicenseType` parameters.

ACCESS LEVEL (USER INTERFACE) LICENSEDISPLAYNAME	ACCOUNTLICENSETYPE	LICENSINGSOURCE	MSDNLICENSETYPE
Basic	express	account	none
Basic + Test Plans	advanced	account	none
Visual Studio Subscriber	none	msdn	eligible
Stakeholder	stakeholder	account	none
Visual Studio Enterprise subscription	none	msdn	enterprise

#### NOTE

The `earlyAdopter` accountLicenseType is an internal value used solely by Microsoft.

You can manage access levels programmatically using the [User Entitlement - Add REST API](#). The following table provides a mapping of the access level selected through the user interface and the `AccountLicenseType`, `licensingSource`, and `msdnLicenseType` parameters.

ACCESS LEVEL (USER INTERFACE) LICENSEDISPLAYNAME	ACCOUNTLICENSETYPE	LICENSINGSOURCE	MSDNLICENSETYPE
Basic	express	account	none
Basic + Test Plans	advanced	account	none
Visual Studio Subscriber	none	msdn	eligible
Stakeholder	stakeholder	account	none
VS Enterprise	none	msdn	enterprise

You can manage access levels programmatically using the [User Entitlement - Add REST API](#). The following table provides a mapping of the access level selected through the user interface and the `AccountLicenseType`, `licensingSource`, and `msdnLicenseType` parameters.

ACCESS LEVEL (USER INTERFACE) LICENSEDISPLAYNAME	ACCOUNTLICENSETYPE	LICENSINGSOURCE	MSDNLICENSETYPE
Basic	express	account	none
Advanced	advanced	account	none
Stakeholder	stakeholder	account	none

ACCESS LEVEL (USER INTERFACE) LICENSEDISPLAYNAME	ACCOUNTLICENSETYPE	LICENSINGSOURCE	MSDNLICENSETYPE
VS Enterprise	none	msdn	enterprise

## What features are available to users who are added to two different access levels?

If a user belongs to a group that has **Basic** access and another group that has **VS Enterprise** access, the user has access to all features available through **VS Enterprise**, which is a superset of **Basic**.

## Service account access

Azure DevOps Server [service accounts](#) are added to the default access level. If you make Stakeholder the default access level, you must add the service accounts to Basic or Advanced/VS Enterprise access.

Service accounts don't require a CAL or other purchase.

## Related articles

- [Stakeholder access quick reference](#)
- [Free access to Pipelines Preview](#)
- [Manage users and access](#)
- [Get started as a Stakeholder](#)
- [Export a list of users and their access levels](#)
- [Default permissions and access](#)
  
- [Stakeholder access quick reference](#)
- [Change access levels](#)
- [Get started as a Stakeholder](#)
- [Export a list of users and their access levels](#)
- [Default permissions and access](#)
- [Compare features between plans](#)

# Azure DevOps Services status

12/13/2022 • 3 minutes to read • [Edit Online](#)

## Azure DevOps Services

We have a team of engineers around the world who look after the health of Azure DevOps 24 hours a day. Their primary goal is to ensure that our customers are always productive and successful with our service. From time to time, like any online service, our service experiences performance slowdowns and stability issues. In these cases, we aim to respond quickly to restore the service. It's our top priority to communicate the incident status and our next steps to mitigate the issue. We do so through the [Azure DevOps Services status portal](#).

If you're experiencing a problem with any of our Azure DevOps Services, you can check the service health to determine if we're already working on the issue. Many of the events we post are based on our Customer Impact Assessment (CIA). The CIA is modeled after our availability model that measures real customer experiences representing both reliability and performance.

## Services within the product suite

Azure DevOps is a product suite of service offerings. The [geographic region](#) indicates where an organization is hosted in the cloud. The data residency, sovereignty, compliance, and resilience requirements are honored within the geographical boundaries.

In addition to the specific Azure DevOps Services, the matrix also displays two other categories: **Core** and **Other**. The **Core** category encompasses the set of features that are fundamental to all five services, such as authentication or the web portal. The **Other** category corresponds to features that complement the suite, such as extensions.

For more information about pricing and acquisition, see the [pricing and acquisition page](#).

## Service health matrix

The service status portal provides a two-dimensional matrix view of active events mapped to a given service and geography. To help clarify which specific aspects of the service are affected, we communicate impact of each of these services by geographic region in the service matrix.

## Service health indicators

The Azure DevOps Services status portal indicates the status of Azure DevOps Services according to the following indicators. These indicators reflect the severity of a service health event based on the number of customers affected by the issue. Typically, the highest severity events impact a large percentage of our customers and render some parts of the product unusable.

 **Healthy**    **Degraded**    **Unhealthy**    **Advisory**

- **Healthy:** Indicates the service is broadly available.
- **Degraded:** Indicates a lower-severity event that affects the performance of a service feature, but doesn't impact broad service availability.
- **Unhealthy:** Indicates a high-severity event that affects the performance of a service and its broad availability.
- **Advisory:** Indicates that a service is under investigation to determine the performance and availability

impact.

## Service status and event logs

You can access more information on active events from the [Status history page](#). This page provides a view into current active events and past events. Each event under investigation or previously investigated is logged in the form of an event log. Each log has other associated information such as the impacted service, geography, and event duration. Choose the provided hyperlink to view the event log, which provides detailed information on the event under investigation.

You can also filter the logs to adjust the scope of your search into past events. In addition, you can use the REST API build automated alerting solutions to help you stay on top of events.

## When and how to report availability issues

If you're experiencing an issue with Azure DevOps and see a corresponding event that's communicated on the service health portal, we're already working to restore normal operations of the service. You don't need to do anything else to notify us.

However, if you don't see your issue reported on the Azure DevOps Services health page, you can ask a question through the [Azure DevOps Services virtual support agent](#).

For issues not related to availability, refer to our [Developer Community portal](#).

## RSS feed

You can use [the RSS feed](#) to subscribe and receive information in your feed reader.

## Use REST APIs to build automated solutions

The [Azure Resource health REST API](#) can retrieve the current health status of each of the Azure DevOps Services. You can use it to build an automated solution to [monitor the infrastructure incidents](#).

### NOTE

Looking for Azure DevOps REST APIs? See the latest [Azure DevOps REST API reference](#).

For information about .NET client libraries, see [.NET client libraries for Azure DevOps](#).

## Related articles

- [Azure Service Health overview](#)
- [Blog post: How do you measure quality of a service?](#)

# Data protection overview

12/13/2022 • 21 minutes to read • [Edit Online](#)

## Azure DevOps Services

Azure DevOps Services is a cloud-hosted application for your development projects, from planning through deployment. Based on the on-premises capabilities, with additional cloud services, we manage your source code, work items, builds, and tests. Azure DevOps uses platform as a service (PaaS) infrastructure and many Azure services, including Azure SQL, to deliver a reliable, globally available service for your development projects.

This article discusses the steps that Microsoft takes to help keep your projects safe, available, secure, and private. Also, it describes the role you play in keeping your projects safe and secure.

This article is intended for organization administrators and IT professionals who manage their project assets daily. It will be most useful to individuals who are already familiar with Azure DevOps and want to know more about how Microsoft protects stored assets in Azure DevOps.

## Our commitment

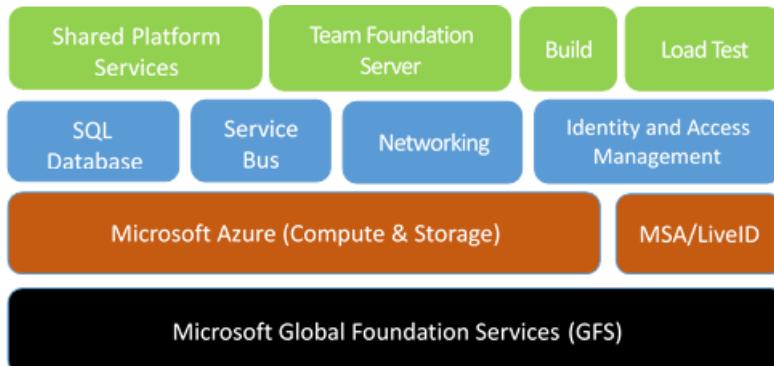
Microsoft helps to ensure that your projects remain safe and secure, without exception. When stored in Azure DevOps, your projects benefit from multiple layers of security and governance technologies, operational practices, and compliance policies. We enforce data privacy and integrity both at rest and in transit.

The threats you face boil down to four basic categories: data availability, service availability, service security, and data privacy. This article explores specific threats within each category, and explains what Azure DevOps does to address them. First, the article describes how data is stored and how Azure DevOps manages access to your data.

Data protection requires active engagement of administrators and users, who must know what steps to take to protect your assets from unauthorized disclosure and tampering. Be explicit when you grant permissions to user access points, so only the right people are accessing data within Azure DevOps.

Whatever your approach, you should consider all data potentially "at risk," no matter where it is or how it's being used. This is true for both data in the cloud and data stored in a private data center. It's important to classify your data, its sensitivity and risk, and the damage it might do if it becomes compromised. Also, categorize your data relative to an overall information security management policy.

## Built on Azure



We host Azure DevOps entirely in Azure data centers. Azure DevOps uses many core Azure services, including

compute, storage, networking, Azure SQL, identity and access management, and Azure Service Bus.

Azure DevOps uses Azure Storage as the primary repository for service metadata and customer data. Depending on the type of data and the storage and retrieval needs, Azure DevOps uses Azure Blob Storage (for binary large objects) and Azure SQL data storage. To understand the Azure DevOps Services approach to data protection, some background on these elements is important.

- **Azure Blob Storage** stores large chunks of unstructured data. All projects use the Azure Blob Storage service. Data includes potentially sensitive or private information, like the contents of source files and attachments for work items. For most projects, the majority of storage in use is this type of unstructured blob storage. For more information, see [Azure Blob Storage](#).
- **Azure SQL Database storage** stores the structured and transactional aspects of your organization, including project metadata, the versioned source control history, and work item details. Database storage gives you fast access to the important elements of your project, and provides indexes into the blob storage to look up files and attachments. For more information, see [Azure SQL Database](#).

Administrators can manage access to resources by [granting or restricting permissions](#) on user identities or groups. Azure DevOps uses federated authentication of user identities via [Azure Active Directory](#) (Azure AD) and Microsoft accounts.

During authentication, the user is routed to the authentication provider, where they provide their credentials. After the authentication provider has verified the user's credentials, Azure DevOps issues an authentication cookie to the user, which allows the user to remain authenticated against Azure DevOps.

In this way, the user's credential information is never shared directly with Azure DevOps. For each Azure DevOps resource that the user attempts to access, permissions are validated based on the user's explicit permissions, as well as permissions inherited through group membership. Administrators can use access controls to protect [access to the organization](#), project collections, team projects, and team-scoped data and functionality. Administrators can also protect more specific assets like version control folders and work item area paths.

## Data availability

Azure DevOps uses many Azure Storage features to ensure data availability if there's a hardware failure, service disruption, or region disaster. Also, the Azure DevOps team follows procedures to protect data from accidental or malicious deletion.

### Data redundancy

To protect data during hardware or service failures, Azure Storage geo-replicates customer data between two regions in the same geography. For example, Azure can geo-replicate data between North and West Europe or between North and South United States.

For Azure Blob Storage, customer data gets replicated three times within a single region, and is replicated asynchronously to a second region in the same geography. As such, Azure always maintains the equivalent of six copies of your data. This enables you to fail over to a separate region if there's a major outage or disaster, while also having local redundancy for hardware failures within a region. For Azure SQL Database storage, daily backups are maintained offsite if there's a regional disaster.

## NOTE

Regarding data redundancy and failover:

- There's an inherent delta, measured in minutes, when Microsoft replicates your data between the primary and secondary region.
- Failover to the secondary region is a decision that Microsoft must make centrally, as it affects all customers on the affected scale unit. Except in extreme circumstances, Microsoft opts to not fail over so that customer data isn't lost.
- Azure DevOps offers a 99.9 percent uptime SLA guarantee, and refunds a portion of the monthly charges if that SLA is missed in a specific month.
- Because there is only one region in Brazil, customer data in Brazil is replicated to the South Central US region for disaster recovery purposes.

## Mistakes happen

To protect against accidental deletion of data, Microsoft also takes point-in-time backups of both the blobs in Azure Blob Storage, and the databases in Azure SQL Database. There's a separate copy of all blobs, and changes are appended to each storage account. Because this data is immutable, you don't need to rewrite any existing storage as part of the backup procedures.

Backups are a standard part of Azure SQL Database, and Azure DevOps makes use of this. We maintain 28 days' worth of your data. In both cases, these backups are also replicated in a paired region, helping to ensure that we recover from a regional outage.

A further protection is that customers can recover their deleted organizations or projects for up to 28 days after deletion. Deleted organizations and projects are in a "soft deleted" stage for 28 days allowing customers to recover as needed. After 28 days they are permanently deleted and cannot be restored.

## IMPORTANT

Accidental deletion here refers to scenarios that arise as a result of an incident on our services and doesn't include accidental deletion of assets (e.g., repositories, work items, attachments, artifacts) by customers. We do not support restoring assets that are accidentally deleted by customers as these backups are meant for business continuity and aid recovery from outage or disaster scenarios only.

## Practice is critical

Having multiple, redundant backups of your data is good but without practice, restoring can be unpredictable. It's been said that "backups never fail, the restores do." While technically incorrect, the sentiment is right.

Microsoft regularly practices restoring various datasets from backup. Geo-redundant storage from Azure is tested regularly. Also, from time to time, we restore from backups to recover from human error, such as when a customer has inadvertently deleted a project in Azure DevOps. There are many combinations of disaster and data corruption scenarios, which we continue to plan and run new tests for regularly.

# Service availability

Azure DevOps offers distributed denial-of-service (DDoS) protections and live site response to help ensure that you have access to your organization and associated assets.

## DDoS protections

In some cases, a malicious DDoS attack can affect service availability. Azure has a DDoS defense system that helps prevent attacks against our service. It uses standard detection and mitigation techniques such as SYN cookies, rate limiting, and connection limits. The system is designed to withstand attacks not only from the outside but also from within Azure.

For application-specific attacks that can penetrate the Azure defense systems, Azure DevOps establishes application and organization level quotas and throttling. This helps prevent any overuse of key service resources during an attack or accidental misuse of resources.

### Live site response

In rare circumstances, you might require a live site response to a problem with service availability. We have an operations team available 24x7, to rapidly identify the issue and to engage the necessary development team resources. Those resources then address the problem. They also aim to update the service status page within minutes of detecting an issue that affects the service. After the team has addressed an issue, they identify the root cause of the issue and track the necessary changes to prevent similar issues in the future.

Azure DevOps live site management processes focus on your experience and the health of your service. These processes minimize the time to detect, respond to, and mitigate problems. All engineering disciplines are involved and responsible, so there are continual improvements evolving out of direct experience. This means that monitoring, diagnostics, resiliency, and quality assurance processes improve over time.

Live site management in Azure DevOps has three distinct tracks: telemetry, incident management, and live site review. Here's what these tracks entail:

Telemetry	Incident management	Live-site review
<ul style="list-style-type: none"><li>• <b>Alerts</b> – define health alerts for failure modes</li><li>• <b>Diagnostics</b> – deliver instrumentation data and operational reports</li><li>• <b>Troubleshooting guides</b> – guidance for investigating an alert is defined by the feature, and then refined by the Service Engineer</li><li>• <b>Failure mode testing</b> – the Service Delivery (SD) team performs failure testing to ensure alerts fire as expected</li><li>• <b>Onboarding</b> – the feature team works with their Service Engineer (SE) to onboard new alerts to the 24 x 7 team</li></ul>	<ul style="list-style-type: none"><li>• <b>Detection</b> – product alerts detect health issues and start the Live Site Incident (LSI) process</li><li>• <b>Triage</b> – The 24 x 7 team receives all critical alerts and confirms impact using Azure DevOps guidance</li><li>• <b>Escalation</b> – both Dev and Ops have individuals in an on-call rotation. SE is the initial escalation path. The SE calls Dev, as needed</li><li>• <b>Incident management</b> – a bridge is managed by the SE who engages Dev. and Partners to troubleshoot</li><li>• <b>Resolution</b> – communication and service restoration are actively driven until customer impact is eliminated</li></ul>	<ul style="list-style-type: none"><li>• <b>Goal</b> – weekly review of LSI ensure that leadership has visibility into live site health and repeat issues</li><li>• <b>Cadence</b> – Incidents from prior week have root cause documented, then reviewed on weekly basis</li><li>• <b>Audience</b> – VS Leadership. Partner team when they drive impact. Developer attends to provide details on Service incident</li><li>• <b>Ownership</b> – Dev. Owns reviews for App and Deploy issues. SD owns for Platform issues.</li><li>• <b>Driving improvements</b> – Bugs and problem work items are logged for gaps (e.g. – missing alerts) and repeat root cause</li></ul>

The operations team also monitors the availability metrics for individual organizations. These metrics provide insights into specific conditions that might affect only some of our customers. Investigations into this data can often result in targeted improvements to address customer-specific issues. In some cases, Microsoft might even contact you directly to understand your experience and work with you to improve the service.

Microsoft publishes a service-level agreement (SLA) and provides a financial guarantee to ensure that we meet this agreement each month. For more information, see [SLA for Azure DevOps](#).

Sometimes partner teams or dependencies have incidents that affect Azure DevOps. All partner teams follow similar approaches to identifying, resolving, and learning from these service outages.

## Service security

Service security requires constant vigilance, from proper design and coding techniques to operational factors. Microsoft actively invests in the prevention of security holes and in breach detection. If there's a breach,

Microsoft uses security response plans to minimize data leakage, loss, or corruption. For more information, see [About security, authentication, and authorization](#).

## Secure by design

Azure DevOps is designed to be secure. Azure DevOps uses the Microsoft Security Development Lifecycle at the core of its development process. The Microsoft Operational Security Assurance program guides its cloud operation procedures. The following methodologies specify the following requirements:

- Threat modeling during service design.
- Following design and code best practices.
- Verifying security with standard tooling and testing.
- Limiting access to operational and customer data.
- Gating rollout of new features through a rigid approval process.

The Azure DevOps team has annual training requirements for all engineers and operations personnel, and sponsors informal "brown bag" meetings hosted by Microsoft engineers. After they've solved an issue raised in a brown bag meeting, they share what they learned with the rest of the team.

A cloud service is only as secure as the host platform. Azure DevOps uses PaaS for much of its infrastructure. PaaS automatically provides regular updates for known security vulnerabilities. VMs hosted in Azure use infrastructure as a service (IaaS), such as for a [hosted build service](#). Such images receive regular updates to include the latest security patches available from Windows Update. The same update rigor applies for on-premises machines, including those used for deployment, monitoring, and reporting.

The Azure DevOps team conducts regular, security-focused penetration testing of Azure DevOps. Using the same techniques and mechanisms as malicious attackers, penetration testing tries to exploit the live production services and infrastructure of Azure DevOps. The goal is to identify real-world vulnerabilities, configurations, errors, or other security gaps in a controlled process. The team reviews the results to identify other areas of improvement and to increase the quality of the preventative systems and training. You can review the results of recent Azure DevOps penetration tests on the [Service Trust Portal](#).

## Credential security

Your credentials in Azure DevOps are stored using industry best practices. Learn more about [credential storage](#).

## Reporting security issues

If during your penetration testing you believe you've discovered a potential security flaw related to the Azure DevOps service, report it to Microsoft within 24 hours. For more information, see [Report a computer security vulnerability](#).

### IMPORTANT

Although notifying Microsoft of penetration testing activities is no longer required, you must still comply with the [Microsoft Cloud Unified Penetration Testing Rules of Engagement](#).

## Bounty program

Azure DevOps participates in the [Microsoft Online Services Bounty Program](#). This program rewards security researchers who report issues to us, and encourages more people to help keep Azure DevOps secure. For more information, see the [Azure DevOps Bounty Program](#).

## Restricting access

Microsoft maintains strict control over who gets access to our production environment and customer data. Access gets granted at the level of least privilege that's required and only after proper justifications get provided and verified. If a team member needs access to resolve an urgent issue or deploy a configuration change, they must apply for "just-in-time" access to the production service. Access is revoked as soon as the situation is

resolved.

Access requests and approvals get tracked and monitored in a separate system. All access to the system correlates against these approvals and if we detect unapproved access, the operations team gets alerted to investigate.

We use two-factor authentication for all remote system access. So, if the username and password for one of our developers or operation staff got stolen, the data remains protected. This means additional authentication checks via smart card or a phone call to a pre-approved number must occur before any remote access to the service is permitted.

In addition, Microsoft uses secrets to manage and maintain the service, such as RDP passwords, SSL certificates, and encryption keys. These are all managed, stored, and transmitted securely through the Azure portal. Any access to these secrets requires specific permission, which is logged and recorded in a secure manner. All secrets are rotated on a regular cadence, and can be rotated on-demand if there's a security event.

The Azure DevOps operations team uses hardened administrator workstations to manage the service. These machines run a minimal number of applications and operate in a logically segmented environment. Operations team members must provide specific credentials with two-factor authentication to access the workstations. All access is monitored and securely logged. To isolate the service from outside tampering, we don't permit applications such as Outlook and Office, as they're often targets of spear-phishing and other types of attacks.

### **Intrusion protection and response**

We encrypt data via HTTPS and SSL to ensure it isn't intercepted or modified while in transit between you and Azure DevOps.

Also, data we store on your behalf in Azure DevOps gets encrypted as follows:

- Data stored in Azure SQL databases gets encrypted using [Transparent Data Encryption \(TDE\)](#). TDE protects against the threat of malicious activity by doing real-time encryption of the database, associated backups, and transaction log files at rest.
- Azure Blob Storage connections get encrypted to protect your data in transit. To protect data at rest stored in Azure Blob Storage, Azure DevOps uses [Azure Storage Service Encryption \(SSE\)](#).

The Azure infrastructure helps the Azure DevOps team to log and monitor key aspects of the service. This helps ensure that activities within the service are legitimate, and detects breaches or attempted breaches. In addition, all deployment and administrator activities are securely logged, as is operator access to production storage. Real-time alerts are raised because the log information is automatically analyzed to uncover potentially malicious or unauthorized behavior.

Where a possible intrusion or high priority security vulnerability gets identified, the team has a clear response plan. This plan outlines responsible parties, steps required to secure customer data, and how to engage with security experts at Microsoft. The team also notifies any organization owners if data was disclosed or corrupted, so that they can take appropriate steps to remedy the situation.

Finally, to help combat emerging threats, Azure DevOps employs an "Assume Breach" strategy. A highly specialized group of security experts within Microsoft, known as the Red Team, assumes the role of sophisticated adversaries. This team tests breach detection and response, to accurately measure readiness and the impacts of real-world attacks. This strategy strengthens threat detection, response, and defense of the service. It also allows the team to validate and improve the effectiveness of the entire security program.

## **Data privacy**

You should have confidence that your data is being handled appropriately and for legitimate uses. Part of that assurance involves appropriately restricting usage so that your data is used only for legitimate reasons.

## General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is the biggest change in data protection laws in Europe since the 1995 introduction of the European Union (EU) Data Protection Directive 95/46/EC. For more information about the GDPR regulation, see the [overview page in the Microsoft Trust Center](#).

### Data residency and sovereignty

Azure DevOps is available in the following eight geographies across the world: United States, Canada, Europe, United Kingdom, India, Australia, Asia Pacific, and Brazil. By default, your organization is assigned to your closest geography, but you do have the option to choose a different geography. If you change your mind later, it's possible to migrate your organization to a different geography, with the assistance of Microsoft support.

Azure DevOps doesn't move or replicate customer data outside of the chosen geography. Instead, your data is geo-replicated to a second region within the same geography. The only exception is Brazil, which replicates data to the South Central US geography for disaster recovery purposes.

#### NOTE

For builds and releases running on Microsoft-provided macOS agents, your data will be transferred to a GitHub data center in the US.

To learn more, see [Azure DevOps data location](#).

### Law enforcement access

In some cases, third parties such as law enforcement entities might approach Microsoft to obtain access to customer data stored within Azure DevOps. We attempt to redirect the requests to the organization owner for resolution. When compelled by court order to disclose customer data to a third party, Microsoft makes a reasonable effort to notify the organization owner in advance, unless we're legally prohibited from doing so.

Some customers require their data storage in a particular geographic location to ensure a specific legal jurisdiction for any law enforcement activities. All customer data, such as source code, work items, test results, and geo-redundant mirrors and offsite backups, get maintained within one of the previously mentioned geographies.

### Microsoft access

From time to time, Microsoft employees need to obtain access to customer data stored within Azure DevOps. As a precaution, all employees who have or might ever have access to customer data must pass a background check, which verifies previous employment and criminal convictions. We permit access to the production systems only when there's a live site incident or other approved maintenance activity, which gets logged and monitored.

Because not all data within our system gets treated the same, we classify data to distinguish between the following data types:

- Customer data - what you upload to Azure DevOps.
- Organization data - information used when you sign up for or administer your organization.
- Microsoft data - information required for or collected through the operation of the service.

Based on the classification, we control usage scenarios, geo-location requirements, access restrictions, and retention requirements.

### Microsoft promotional use

Microsoft occasionally wants to contact customers to let them know about additional features and services that might be useful. Because not all customers want to be contacted about these offers, you can opt in and opt out of marketing email communications.

Microsoft never uses customer data to target specific offers for specific users or organizations. Instead, we use organization data and aggregate usage statistics at the organization level to determine groups that should receive specific offers.

## Building confidence

You can be confident in other efforts Microsoft makes on behalf of Azure DevOps. These efforts include internal adoption policies at Microsoft, the level of transparency provided into the state of our service, and progress towards receiving certification of our information security management systems.

### Internal adoption

Teams across Microsoft are adopting Azure DevOps internally. The Azure DevOps team moved into an organization in 2014 and uses it extensively. More broadly, we have established guidelines to enable the adoption plans for other teams.

Obviously, large teams move more gradually than smaller ones, given their investments in existing DevOps systems. For teams able to move quickly, we have established a project classification approach. It assesses risk tolerance, based on project characteristics, to determine if the project is appropriate for Azure DevOps. For larger teams, the adoption typically occurs in phases, with more planning.

Additional requirements for internal projects include associating the organization with Azure AD to ensure proper user identity life cycle and password complexity. For more sensitive projects, two-factor authentication is also required.

### Compliance certifications

Some of you want to understand third-party evaluation of our data security procedures. Azure DevOps has achieved the following certifications:

- ISO 27001:2013
- ISO 27018:2019
- HIPAA (Health Insurance Portability and Accountability Act)
- BAA (Business Associate Agreement)
- EU Model Clauses
- SOC 1 Type 2
- SOC 2 Type 2
- Germany C5

The SOC audit for Azure DevOps covers controls for data security, availability, processing integrity, and confidentiality. The SOC reports for Azure DevOps are available through the [Microsoft Service Trust Portal](#).

## Steps you can take

Proper data protection requires your active engagement, as well as that of your administrators and users. Your project data stored within Azure DevOps is only as secure as the end-user access points. Match the level of permission strictness and granularity for those organizations with your project's sensitivity level.

### Classify your data

The first step is to classify your data. Classify data based on sensitivity and risk horizon, and the damage that might occur if it gets compromised. Many enterprises have existing classification methods that can be reused when projects move to Azure DevOps. For more information, you can download the "Data classification for cloud readiness" document from Microsoft Trustworthy Computing.

### Adopt Azure Active Directory

Use Azure Active Directory (Azure AD) to manage your organization's access to Azure DevOps. Azure AD

provides another way to improve the security of your users' credentials. Azure AD allows your IT department to manage its end-user access policy, password complexity, password refreshes, and expiration if the user leaves your organization. Through Active Directory federation, you can directly link Azure AD to your organization's central directory, so you have only one location to manage these details for your enterprise.

The following table compares Microsoft account and Azure AD characteristics relative to Azure DevOps access:

Properties	Microsoft Account	Azure AD
Identity creator	User	Organization
Single username / password for all work assets	No	Yes
Password lifetime & complexity control	User	Organization
Azure DevOps membership limits	Any Microsoft account (MSA)	Organization's directory
Traceable identity	No	Yes
Organization & IP ownership	Unclear	Organization
Two-factor authentication enrollment	User	Organization
Device-based conditional access	No	Organization

Learn more about [configuring this support for your organization](#).

### Require two-factor authentication

You might want to restrict access to your organization by requiring more than one factor to sign in. You can require multiple factors with Azure AD. For example, you can require phone authentication, in addition to a username and password, for all authentication requests.

### Use BitLocker

For sensitive projects, you can use BitLocker on your Windows laptop or desktop computer. BitLocker encrypts the entire drive on which Windows and your data reside. When BitLocker is enabled, it automatically encrypts any file you save on that drive. If your laptop or desktop machine falls into the wrong hands, BitLocker prevents unauthorized access of local copies of data from your projects.

### Limit use of alternate authentication credentials

The default authentication mechanism for Git-related tooling is alternate authentication (sometimes referred to as basic authentication). This mechanism allows the end user to set up an alternate username and password for use during Git command-line operations. This username and password combination can also be used to access any other data for which that user has permissions. By its nature, alternate authentication credentials are less secure than the default federated authentication.

You can still make choices for increased security. All communication gets sent over HTTPS, and there are password complexity requirements. Your organization should continue to evaluate whether additional policies are required to meet your project security requirements. You can disable alternate authentication credentials altogether if you decide that it doesn't meet your organization's security requirements. For more information, see [Change application connection & security policies for your organization](#).

### Secure access to your organization

Azure AD provides the ability for administrators to control access to Azure resources and applications such as Azure DevOps. With conditional access control in place, Azure AD checks for the specific conditions you set for a

user to access an application. After access requirements are met, the user is authenticated and can access the application.

Azure DevOps supports enforcing certain types of conditional access policies (for example, IP fencing) for custom Azure DevOps authentication mechanisms. These mechanisms include personal access tokens, alternate authentication, OAuth, and SSH keys. If your users are accessing Azure DevOps through a third-party client, only IP-based policies (IPv4 based only) are honored.

## Additional resources

- [Azure DevOps home page](#)
- [Azure DevOps data location](#)
- [Microsoft privacy statement](#)
- [Azure DevOps support](#)
- [Features and services included with Azure DevOps](#)
- [Azure trust center](#)
- [Microsoft Security Development Lifecycle](#)

# Data locations for Azure DevOps

12/13/2022 • 2 minutes to read • [Edit Online](#)

## Azure DevOps Services

You can choose the location for your data during initial sign-up and creation of your organization. Azure DevOps operates in the following geographical locations, or "geos".

## Data locations

Azure DevOps data is available in the following eight geographies across the world:

- Australia
- Brazil
- Canada
- Asia Pacific
- Europe (EU)
- India
- United Kingdom
- United States (US)

We default your organization to your closest geography. However, you can choose a different geography. If you change your mind afterward, you can [migrate your organization to a different geography](#).

For more information, see [Data residency in Azure](#).

## Customer data

Except [as noted](#), Azure DevOps maintains all customer data within your selected geography. Customer data includes the following data types:

- source code
- work items
- test results
- geo-redundant mirrors and offsite backups

Azure DevOps works with and uses many Microsoft Azure services. For more information and details on customer data retention by location, see [Data residency in Azure](#).

## Profile data

Azure DevOps stores information that's global in nature, such as user identities and profile information as follows:

- EU-based users: profile data is in EU data center
- US-based users: profile data is in US data center
- Users from all other countries and regions: profile data is in US data center

## Allow list data for tenant policies

We recommend using groups with your tenant policy allow list(s). If you use a named user, be aware that a reference to the named user's identity will reside in the United States, Europe (EU), and Southeast Asia

(Singapore).

## Transferring your data

We don't transfer customer data outside of your selected geography. However, we will transfer your data if we need to do any of the following actions:

- provide customer support
- troubleshoot the service
- comply with legal requirements

If needed, you can transfer your data using preview, beta, or other pre-release services. These services typically store your data in the United States, but may store it globally.

### NOTE

For builds and releases running on Microsoft-provided macOS agents, your data will be transferred to a GitHub data center in the US. This data center location is owned and managed by GitHub with compliance certifications, such as SOC 1 & 2 Type II reports available [here](#).

Microsoft doesn't control or limit the geographies from which you or your users may access your data.

### NOTE

Because there's only one region in Brazil, customer data is replicated to south-central United States for disaster recovery and load balancing purposes. For more information, see [Data residency in Azure](#).

## Related articles

- [Get started with Azure DevOps](#)
- [Data protection overview](#)

# How we store your credentials for Azure DevOps Services

12/13/2022 • 2 minutes to read • [Edit Online](#)

## Azure DevOps Services

### IMPORTANT

Azure DevOps no longer supports Alternate Credentials authentication since the beginning of March 2, 2020. If you're still using Alternate Credentials, we strongly encourage you to switch to a more secure authentication method (for example, personal access tokens). [Learn more](#).

## Credential security

Microsoft is committed to ensuring that your projects remain safe and secure, without exception. In Azure DevOps, your projects benefit from multiple layers of security and governance technologies, operational practices, and compliance policies. We enforce data privacy and integrity both at rest and in transit. In addition, we adhere to the following practices with respect to the credentials or secrets that Azure DevOps stores. To learn more about how to choose the right authentication mechanism, see [Guidance for authentication](#).

## Personal access tokens (PATs)

- We store a hash of the PAT
- Raw PAT is generated in-memory on the server side as 32 bytes randomly generated through RNGCryptoServiceProvider then shared with the caller as a base-32-encoded string. This value is NOT stored
- PAT hash is generated in-memory on the server side as an *HMACSHA256Hash* of the raw PAT using a 64-byte symmetric signing key stored in our key vault
- Hash is stored in our database

## Secure shell (SSH) keys

- We store a hash of the enclosing organization ID and the SSH public key
- Raw public key is provided directly by the caller over SSL
- SSH hash is generated in-memory on the server side as an *HMACSHA256Hash* of the organization ID and raw public key using a 64-byte symmetric signing key stored in our key vault
- Hash is stored in our database

## OAuth credentials (JWTs)

- These are issued as fully self-describing JSON web tokens (JWTs) and are NOT stored in our service
- The claims in JWTs issued and presented to our service are validated using a certificate stored in our key vault

# Launch Visual Studio via Azure DevOps Services

12/13/2022 • 2 minutes to read • [Edit Online](#)

## Azure DevOps Services

When you first open [Visual Studio 2015](#), you can sign in and connect to [Azure DevOps Services](#).

If you're already signed in to Visual Studio or using Visual Studio 2017, [connect to Azure DevOps Services](#).

Once you're connected, you can store or share code in free, unlimited, private, cloud-based Git repositories or Team Foundation Version Control (TFVC). Organize and manage your work with Agile tools for DevOps, continuous integration, and continuous delivery. Your team can build often, test early, and ship faster.

To set up Visual Studio without Azure DevOps Services, learn how to [get started](#). To host your own server, learn how to [install and set up Azure DevOps Server](#).

Azure DevOps Services is free for [up to five users with access to Basic features](#) and for unlimited [Visual Studio subscribers](#) and [Stakeholders who can access limited features](#). Learn [what else you get with Azure DevOps Services](#). If you want, you can also use Azure DevOps Services with any IDE or code editor, like the following examples:

- [Eclipse, Android Studio, or IntelliJ](#)
- Xcode (see [Git](#) or [TFVC](#))
- [Visual Studio Code](#)

## How do I set up Visual Studio 2015 for Azure DevOps Services when I sign in?

1. [Download and install Visual Studio](#), if you don't have the version you want already. [Which versions can I use with Azure DevOps Services?](#)

If you have a Visual Studio subscription that includes the Visual Studio IDE, get the version that's available with your subscription.

2. Start Visual Studio, and then sign in to create your profile.

This profile saves your settings and roams with you when you sign in to Visual Studio on any computer. [Why else should I sign in?](#) If you're a Visual Studio subscriber, use the sign in address for your subscription.



## Welcome. Sign in to Visual Studio.

Visual Studio will automatically keep you signed in, sync your settings between devices, and connect to online developer services.

[Learn more](#)

[Sign in](#)

[Not now, maybe later.](#)

[Can't sign in?](#)

3. Enter your sign in address, and then enter your password.
4. Add your Visual Studio profile details. You only need to add these details once.

The screenshot shows a profile setup screen. It includes fields for 'Full name' (Jamal Hartnett), 'Contact e-mail' (jamalhartnett@outlook.com), 'Phone number' (empty), and 'Country/Region' (United States). The 'Full name' field has a question mark icon next to it.

5. Give your organization a name, and confirm its location.

The screenshot shows a step to create an organization. It asks for a 'Visual Studio Team Services site' URL (https://fabrikam) and a location (.visualstudio.com). It states the account will be hosted in the South Central US region. There is a 'Change options' link and a note about Microsoft's use of contact information. A 'Continue' button is at the bottom.

[How can I create an organization later or change its location?](#)

6. Create your first project to store your code, work items, backlog, builds, tests, and other assets. Name your project, select a process to organize your work, and choose the version control to manage your code.

 **Visual Studio®**

## Create your first team project

Welcome. Your account, <https://fabrikam.visualstudio.com/>, is created and ready to go. Now create your first team project where you'll host your code and backlog. [Learn more](#)

Project name: \*

Process template: \*

Version control: \*

 Git [?](#)     Team Foundation Version Control [?](#)

Create a README.md file to describe this project

[Not now, maybe later](#) [Create project](#)

Not sure which to choose? Learn which [process](#) and version control ([Git](#) or [TFVC](#)) work best for you.

7. If you're a new Visual Studio user, you can change your settings here, or change them later in Visual Studio options.

 **Visual Studio®**

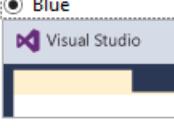
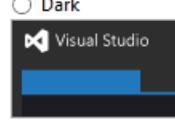
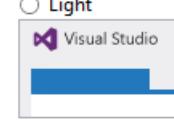
Hello, Jamal Hartnett

 [jamalhartnett@outlook.com](mailto:jamalhartnett@outlook.com)  
[View your Visual Studio profile](#)

Start with a familiar environment

Development Settings:

Choose your color theme

**Blue**     **Dark**     **Light** 

You can always change these settings later.

[Start Visual Studio](#)

These changes are saved with your profile, and your settings roam with you wherever you sign in.

8. To view your new organization, sign in to <https://dev.azure.com/{yourorganization}>.

## Next steps

[Add users to your organization](#)

## Related articles

- Add code to [Git](#) or [TFVC](#).
- [Create your backlog](#) to organize your work, manage your process, or [customize your process](#).



# About projects and scaling your organization

12/13/2022 • 11 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019 | TFS 2018

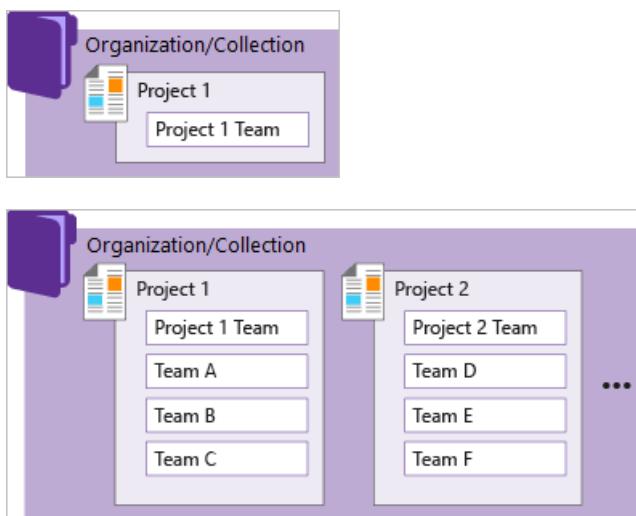
A project in Azure DevOps provides a place for users to plan, track progress, and collaborate on building software solutions. A project represents a fundamental container where you can store data and source code.

When you create your project, Azure DevOps automatically creates a team of the same name, which is sufficient for small organizations. For enterprise-level organizations, it may be necessary to scale up and create more teams and projects. You can have up to 1000 projects within an organization in Azure DevOps.

The following diagram shows one project and team versus multiple projects and teams in an organization or collection. This structure allows teams to configure the tools in ways that work for them and complete administrative tasks at the appropriate levels. As your organization grows, your tools can grow to support a [culture of team autonomy and organizational alignment](#).

## One project + team

## Multiple projects + teams



For more information, see [Work tracking, process, and project limits](#) and [Plan your organizational structure](#).

## Manage work across your organization

When you connect to Azure DevOps, you connect to an organization or project collection. Within that container, one or more projects may be defined. At least one project must be created to use the system.

You can scale your organization in the following ways:

- To support different business units, you can add projects
- Within a project, you can add teams
- Add repositories and branches
- To support continuous integration and deployment, you can add agents, agent pools, and deployment pools
- To manage a large number of users, you can manage access through Azure Active Directory

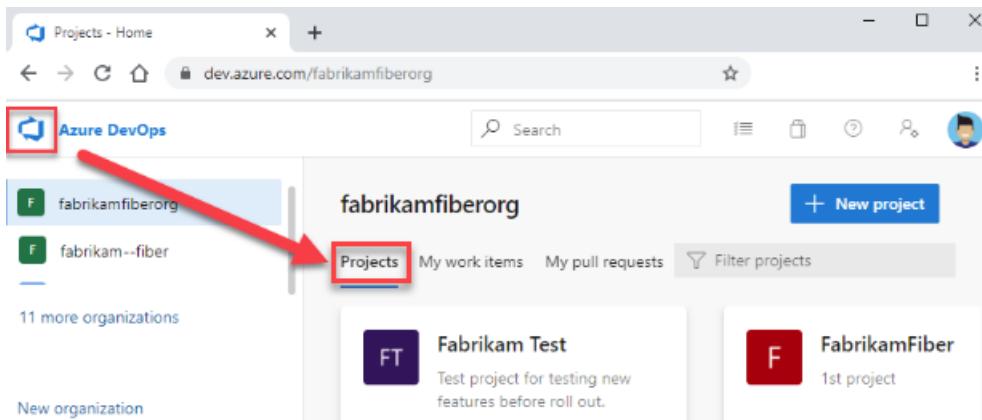
You can scale your on-premises Azure DevOps deployment in the following ways:

- To increase performance, you can add server instances
- To support different business units, you can add project collections and projects
- Within a project, you can add teams
- Add repositories and branches
- To support continuous integration and deployment, you can add agents, agent pools, and deployment pools
- To manage a large number of users, you can manage access through Active Directory

## View projects in your organization

View the projects defined for your organization by opening the **Projects** page.

1. Select  **Azure DevOps** to open **Projects**.

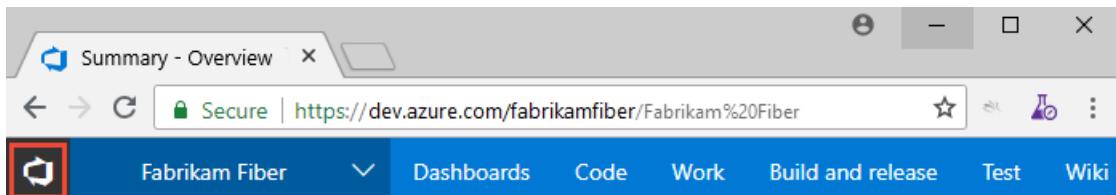


The screenshot shows the 'Projects - Home' page in a web browser. The URL is dev.azure.com/fabrikamfiberorg. The top navigation bar includes the 'Azure DevOps' icon, a search bar, and user profile options. Below the bar, the organization 'fabrikamfiberorg' is selected. A red box highlights the 'Projects' tab in the top navigation bar. Other tabs include 'My work items', 'My pull requests', and 'Filter projects'. The main content area displays two projects: 'Fabrikam Test' (FT) and 'FabrikamFiber' (F). A sidebar on the left shows '11 more organizations' and a 'New organization' button.

2. Choose a project from the list of projects.

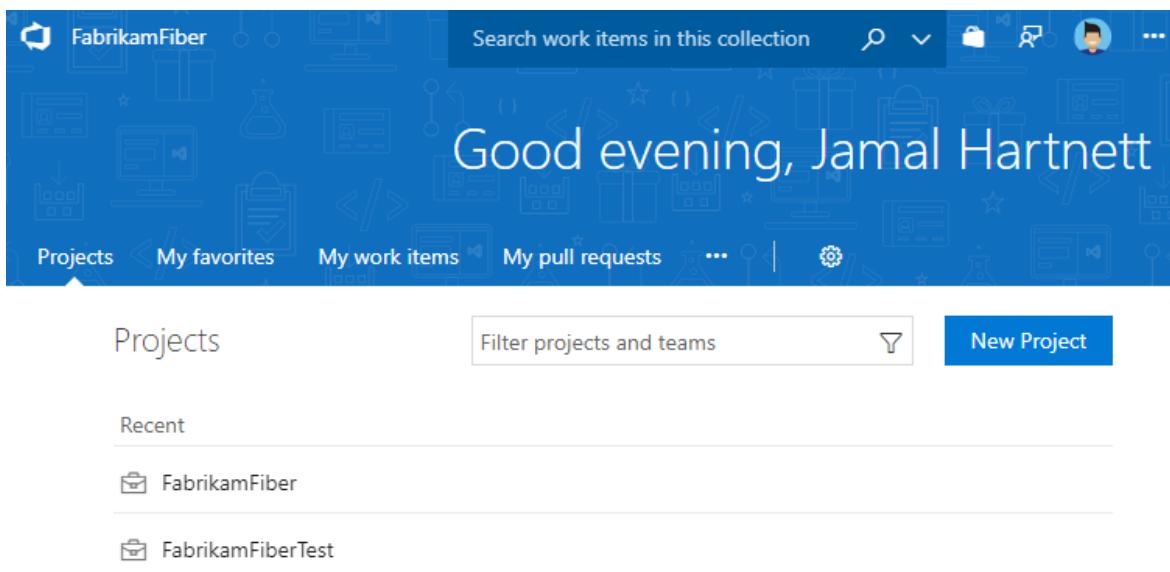
For more information, see [Create a project](#).

1. Select  **Azure DevOps** to open **Projects**.



The screenshot shows the 'Summary - Overview' page for the 'Fabrikam Fiber' organization. The top navigation bar includes the organization icon, a secure connection indicator, and links for 'Dashboards', 'Code', 'Work', 'Build and release', 'Test', and 'Wiki'. A red box highlights the organization icon.

2. Choose a project from the projects list.



The screenshot shows the 'Projects' page for the 'FabrikamFiber' organization. The top navigation bar includes the organization icon, a search bar, and user profile options. A red box highlights the 'Projects' tab in the top navigation bar. The main content area displays a list of recent projects: 'FabrikamFiber' and 'FabrikamFiberTest'. A 'Filter projects and teams' search bar and a 'New Project' button are located at the bottom of the list.

# Limit visibility of projects

By default, users added to an organization can view all organization and project information and settings.

The **Limit user visibility and collaboration to specific projects** preview feature for the organization limits user access in the following ways.

- Restricts views that display a list of users, list of projects, billing details, usage data, and more information accessed through **Organization settings**.
- Limits the set of users or groups that appear through people-picker search selections and the ability to @mention users.

## IMPORTANT

- The limited visibility features described in this section apply only to interactions through the web portal. With the REST APIs or `azure devops` CLI commands, project members can access the restricted data.
- Guest users who are members in the limited group with default access in Azure AD, can't search for users with the people picker. When the preview feature's turned *off* or when guest users aren't members of the limited group, guest users can search all Azure AD users, as expected.

## Limit access to organization settings

To limit access to organization settings, [enable the Limit user visibility and collaboration to specific projects preview feature](#). Users and groups in the "Project-scoped users group" can't access organization settings. They can only see the [Overview](#) and [Projects](#) pages and those projects to which they've been added.

## NOTE

All security groups are organization-level entities, even those groups that only have permissions to a specific project. From the web portal, visibility of some security groups may be limited based on user permissions. However, you can discover the names of all groups in an organization using the `azure devops` CLI tool or our REST APIs. To learn more, see [Add and manage security groups](#).

## NOTE

All security groups are collection-level entities, even those groups that only have permissions to a specific project. From the web portal, visibility of some security groups may be limited based on user permissions. However, you can discover the names of all groups in an organization using the `azure devops` CLI tool or our REST APIs. To learn more, see [Add and manage security groups](#).

## NOTE

All security groups are collection-level entities, even those groups that only have permissions to a specific project. From the web portal, visibility of some security groups may be limited based on user permissions. However, you can discover the names of all groups in an organization using the REST APIs. To learn more, see [Add and manage security groups](#).

## Limit visibility within people pickers

Organizations that are connected to Azure Active Directory (Azure AD) can use people pickers. People pickers support searching all users and groups added to Azure AD, not just those users and groups added to your project. People pickers support the following Azure DevOps functions:

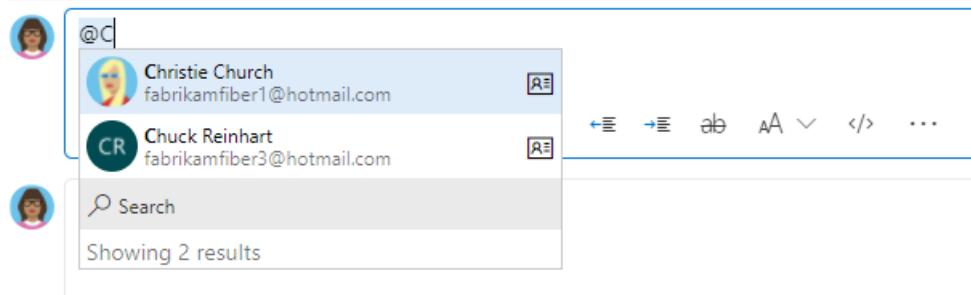
- Select a user identity from a work tracking field, such as "Assigned to"
- Select a user or group with *@mention* in a work item discussion or field, pull request discussion, commit

comments, or changeset or shelveset comments

- Select a user or group using *@mention* from a wiki page

As shown in the following image, start to enter a user in the people picker box until you find a match to the user name or security group.

## Discussion



### WARNING

When you enable the **Limit user visibility and collaboration to specific projects** preview feature, project-scoped users can't search for users who were added to the organization through Azure AD group membership, rather than through an explicit user invitation. We're working on a solution to this behavior. As a work around, you can disable the **Limit user visibility and collaboration to specific projects** preview feature.

Users and groups within the **Project-scoped users** group can only see and select users and groups in the project they're connected to from a people picker. To scope people pickers for all project members, see [Limit identity search and selection](#).

## View historical data

All project members can view identities that were added to a comment, discussion, or assignment. For example, everyone in the project (even users with the new restriction) can still see a user's name assigned to a work item when the user's no longer part of the project. The same is true for @mentions in PRs, comments, discussions, and more.

## Use a single project

We recommend that you use a single project to support your organization or enterprise. A single project minimizes the maintenance of administrative tasks and supports the most optimized and full-flexibility [cross-link object](#) experience.

Even if you have many teams working on hundreds of different applications and software projects, you can easily manage them within a single project. A project serves to isolate data stored within it and you can't easily move data from one project to another. When you move data from one project to another, you typically lose the history associated with that data.

For more information, see [How many projects do you need?](#).

## Add another project

You may want to add another project in the following instances:

- To prohibit or manage access to the information contained within a project to select groups
- To support custom work tracking processes for specific business units within your organization
- To support entirely separate business units that have their own administrative policies and administrators
- To support testing customization activities or adding extensions before rolling out changes to the working project

- To support an open-source software (OSS) project

You may want to add another project in following instances:

- To prohibit or manage access to the information contained within a project
- To support custom work tracking processes for specific business units within your organization
- To support entirely separate business units that have their own administrative policies and administrators
- To support testing customization activities or adding extensions before rolling out changes to the working project

## Use private and public projects

You can have both private and public projects. You can also [change the visibility of a project from private to public](#).

**Private projects** require that you add and manage user access. Users must sign in to gain access to a project, even if it's read-only access. All project members have access to the project and organization information. For more information, see [Resources granted to project members](#).

**Public projects** don't require users to sign in to gain read-only access to many of the following services. Public projects provide support to share code with others and to support continuous integration/continuous deployment (CI/CD) of open-source software.

For more information about features and access levels for public projects, see [Make a private project public](#).

## Version control support

Git repositories can be browsed and cloned, but only via HTTPS. SSH and GVFS endpoints are unavailable. Clients like Visual Studio and IntelliJ work with the HTTPS clone URL but don't offer the connected experience linking to work items and other collateral.

## Dashboard widget support

The following dashboard widgets won't display any useful information for non-members.

- Assigned to me
- Code tile
- New work item
- Pull request
- Query results
- Requirements quality
- Sprint burndown
- Sprint capacity
- Sprint overview
- Team members
- Welcome
- Work links
- Other links

## Structure your project

Use the following elements to structure your project to support your business needs.

- Create a Git repository for each subproject or application, or [create root folders within a TFVC repository](#) for each subproject. If you're using TFVC and heading toward a combined project model, create root folders for different teams and projects, just as you would create separate repos in Git. Secure folders as needed and control which segments of the repo you're actively using with workplace mappings.
- Define area paths to support different subprojects, products, features, or teams.
- Define iteration paths (also known as sprints) that can be shared across teams.
- Add a team for each product team that develops a set of features for a product. Each team you create automatically creates a security group for that team, which you can use to manage permissions for a team. For more information, see [Portfolio management](#).
- Grant or restrict access to select features and functions using custom security groups.
- Create query folders to organize queries for teams or product areas into folders.
- Define or modify notifications set at the project level.

## Customize and configure your project

You can configure and customize most services and applications to support your business needs or the way your teams work. Within each project, you can do the following tasks. For a comprehensive view of which resources can be configured, see [About team, project, and organizational-level settings](#).

- **Dashboards:** Each team can [configure their set of dashboards](#) to share information and monitor progress.
- **Source control:** For each [Git repository](#), you can apply branch policies and define branch permissions. For TFVC repositories, you can [set check-in policies](#).
- **Work tracking:** You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For more information, see [Customize an inheritance process](#).
- **Azure Pipelines:** You can fully customize your build and release pipelines, and define build steps, release environments, and deployment schedule. For more information, see [Build and release](#).
- **Azure Test Plans:** You can define and configure test plans, test suites, test cases, and test environments. You can also add test steps within your build pipelines. For more information, see [Exploratory and manual testing](#) and [continuous testing for your builds](#).
- **Dashboards:** Each team can [configure their set of dashboards](#) to share information and monitor progress.
- **Source control:** For each [Git repository](#), you can apply branch policies and define branch permissions. For TFVC repositories, you can [set check-in policies](#).
- **Work tracking:** You can add fields, change the workflow, add custom rules, and add custom pages to the work item form of most work item types. You can also add custom work item types. For more information, see [Customize the on-premises XML process model](#).
- **Build and release:** You can fully customize your build and release pipelines, and define build steps, release environments, and deployment schedule. For more information, see [Build and release](#).
- **Test:** You can define and configure test plans, test suites, test cases, and test environments. You can also add test steps within your build pipelines. For more information, see [Exploratory and manual testing](#) and [continuous testing for your builds](#).

## Add a team

As your organization grows, you can add teams equipped with configurable Agile tools to meet each team's workflow. For more information, see the following articles.

- [Scale Agile to large teams](#)
- [About teams and Agile tools](#)
- [Manage a portfolio of backlogs](#) and see progress.

- [Use delivery plans](#) to scheduled work items by sprint (iteration path) of selected teams against a calendar view.
- [Incrementally adopt practices that scale](#) to create greater rhythm and flow within your organization, engage customers, improve project visibility, and develop a productive workforce.
- [Structure projects to gain visibility across teams](#) or to support [epics](#), [release trains](#), and [multiple backlogs](#) to support the Scaled Agile Framework.

## Connect to a project with other clients

Aside from connecting via a web browser, you can connect to a project from the following clients:

- [Visual Studio \(Professional, Enterprise, Test Professional\)](#)
  - [Visual Studio Code](#)
  - [Visual Studio Community](#)
  - [Eclipse: Team Explorer Everywhere](#)
  - [Office Excel](#)
  - [Azure Test Plans](#) (formerly Test Manager)
  - [Microsoft Feedback Client](#)
- 
- [Visual Studio \(Professional, Enterprise, Test Professional\)](#)
  - [Visual Studio Code](#)
  - [Visual Studio Community](#)
  - [Eclipse: Team Explorer Everywhere](#)
  - [Office Excel](#)
  - [Azure Test Plans](#) (formerly Test Manager)
  - [Microsoft Feedback Client](#)

For more information, see [Compatibility with Azure DevOps Server versions](#).

## Frequently asked questions (FAQs)

### **Q: Can I move or transfer a project to another organization or collection?**

A: Yes, but not without losing data. You can manually copy resources and leave some behind, or use a third-party tool, such as [OpsHub Visual Studio Migration Utility](#), which copies data using the REST APIs.

### **Q: What programmatic tools support projects?**

A. See [Projects REST API](#).

You can also use the [az devops project CLI](#).

## Related articles

- [Get started as an administrator](#)
- [Web portal navigation](#)
- [What do I get with a project?](#)
- [Understand differences between Azure DevOps](#)

